Annual // Report 2020

Since 1997 NLnet foundation (after its historical contribution to the early internet inside and outside of Europe) has been financially supporting organizations and people that contribute to an open information society. It funds those with ideas to fix the safety, robustness and privacy of the internet.

The articles of association for the NLnet foundation state: *"to promote the exchange of electronic information and all that is related or beneficial to that purpose"*. Stichting NLnet is a recognised philantropic non-profit foundation according to the Netherlands Tax Authority (Belastingdienst)

The internet has no borders, and neither does NLnet. It operates internationally, and is driven by donations from individuals and from private and public organisations. NLnet is independent, and all projects are based on open standards, open source software, hardware and content.

Introduction

Dear reader,

Thank you for your interest in the work of NLnet Foundation. 2020 was quite a year, and we are proud that our foundation and the projects and people we support have made very meaningful contributions to mitigate the situation caused by the global SARS-CoV-2 pandemic. Overnight public life as we knew it came close to a standstill, and the internet became an even more critical part of our social fabric than it already had been up to that point. Billions of hours of online classes and meetings were held with online free and open source videoconferencing software supported by NLnet – keeping schools and offices going, while protecting the privacy of users of all ages that could no longer be together in person due to the lockdowns. Projects from our ongoing and historical portfolio of open source and open hardware projects contributed their knowledge, and empowered people to help themselves and others cope. Our work together with the European Commission – in the context of reviewfacility.eu – helped to mature several critical technologies, amongst others by creating a framework for evaluation and transparently assessing solutions on their security and privacy characteristics. And closer to home: thanks to our grants hundreds of independent researchers and developers were able to continue their important contributions to free and open technologies without too much worries about financial stability. We did miss meeting you all in person at meetups and conferences, but the internet stayed up and delivered.

The annual report you have in front of you is only a superficial reflection of a rich and memorable year. Mere written text cannot do proper justice to the dedication of the people we work with, and the wide array of exciting projects they introduced us to. The projects we support continue to push technology forward and towards a more healthy balance of power. Our portfolio grow both in depth and in width in 2020. Together with our partners within NGI Zero we are now facilitating more than 250 (!) NGI Zero PET and Discovery projects that contribute to an open, accessible and diverse internet. In this annual report you'll find more information about these projects, and others. An additional three year subgranting programme, called NGI Assure, became operational in september of this year, and will start yielding the first concrete results in 2021.

It deserves special mention that this year Radically Open Security – the not-for-profit security company that has dedicated its profits to NLnet – passed the half million euro mark with cumulative donations. This is simply amazing, and we cannot thank Melanie Rieback and her team enough for their trust and generosity. Diversity of sources of income is of vital importance for a foundation like ours, in order to retain an independent course. Additionally, in 2020 we received some grants and donations from other foundations like Vietsch Foundation, from various international research networks, from NCSC and the Netherlands Ministry of Economic and Climate Affairs. We continued our fruitful cooperation with The Commons Conservancy, together with the European association of research networks GÉANT (and its members) and our subsidiary The Commons Caretakers B.V. This work would not be possible without many volunteers, from inside and outside of the Commons Conservancy and its programmes.

Stichting NLnet is growing, thanks to all the support and help we get from people and organisations from around the world. We see ourselves as humbly serving a larger movement seeking to empower the internet user, and do our best to play our part in establishing a collective answer to the many issues

that need to be solved before the internet is the internet we want and deserve – robust, resilient, trustworthy and sustainable. We may be a small player operating in a quite a specific niche, but that niche happens to be pretty relevant to all of society. Of course the real ground work is done by the amazing organisations and people we work with. We all share the mission of making a better internet for tomorrow, for everyone across the globe – and we believe there is no time to waste, if only to be prepared for the next time catastrophy strikes.

Enjoy, and let us know if you have any thoughts – we are always looking for great new ideas, and for ways to improve the way we work!

On behalf of the NLnet team,

Bob Goudriaan
Managing Director,
Chair Governing Board

Michiel Leenaars
Director of Strategy

Table of Contents

1 NLnet organisation

History

NLnet's history started in April 1982 with the announcement of a major initiative to develop and provide network services in Europe. The Netherlands Local Unix User Group (**NLUUG**) played a major role in raising the so-called pan-European "UNIX" Network, EUnet; to support these activities the NLUUG members founded NLnet. NLnet was formally established by the NLUUG as a "stichting" (Dutch for foundation) on February 27, 1989.

Funding origins

In November 1994, NLnet Holding BV was formed by the foundation in order to create a commercial base for its internet activities. NLnet Holding BV was the very first commercial Internet access provider in the Netherlands. The sale of NLnet's Internet Service Provider (ISP) activities to UUnet (now part of Verizon) in 1997 provided Stichting NLnet with the means to actively stimulate the development of network technology and to make this freely available to the community in its broadest sense.

These days, the majority of funding for NLnet activities comes from external sources. Private individuals, public sector organisations, foundations and other not-for-profit organisations, as well as commercial organisations donate or bequeathe to NLnet – we believe because they appreciate the way the technology commons are being fostered by NLnet. Stichting NLnet is a recognized public benefit organisation (**Algemeen Nut Beogende Instelling** or ANBI) according to Netherlands legislation.

Domicile

NLnet Foundation holds offices at Science Park Amsterdam, a technology hotspot with a long history of pioneering in network technology R&D in The Netherlands. It is opposite the road of the location where the first regular connection to the public internet outside of the United States of America was made in 1988 (**CWI**), where the NLnet activities were located at the time.

Supervisory Board

In 2020, the Supervisory Board (Raad van Toezicht) of Stichting NLnet consists of:

- Simon Hania, chair
- Harm Rietmeijer
- Hanneke Slager

These positions are non-remunerated positions in accordance with the NLnet Statutes, except for a financial compensation for time spent ('vacatiegeld').

Governing Board

The Governing Board of Stichting NLnet in 2020 consisted of:

- Bob Goudriaan, chair

This is a non-remunerate position in accordance with the NLnet Statutes.

Operations

For daily operations the NLnet Bureau was staffed in 2020 with the following people, totaling the staff to 5 fte (Full Time Equivalent), all are remunerate positions:

- Bob Goudriaan, general director
- Michiel Leenaars, strategy director
- Joost Agterhoek, policy & technology advisor
- Maja Kraljic, diversity advisor
- Jos van den Oever, senior policy & technology advisor
- Patricia Otter, administrator

Total FTE costs in 2020 were € 476 880,-.

Operations support

For external (financial and legal) advice and consultancy, Stichting NLnet is supported by:

- Koningsbos Accountants (accountancy)
- Bourquin Business Lawyer (legal advice)

The NLnet website https://nlnet.nl is supported by TNX and MARKOV Solutions.

Independent Review Committee Internet Hardening Fund

An independent review committee consisting of three experts from the technical and academic internet community review the outcomes of the selection procedure of the Internet Hardening Fund based on criteria of eligibility and efficacy. The review committee may set additional conditions for granting. Members of the committee, their employers, colleagues and family members are disallowed for submitting projects to the *Internet Hardening Fund*. The members of the committee are not linked to NLnet in a role as employee, member of the board of directors or supervisory board.

In 2020 the independent review committee for the Internet Hardening Fund consisted of:

- Leon P. Kuunders, CISA CISM CISSP
- Niels Sijm
- Bert Wijnen

Independent Review Committee NGI

An independent review committee consisting of nine experts from the technical and academic internet

community review the outcomes of the selection procedure of the NGI Zero programmes, based on criteria of eligibility and efficacy. The review committee may set additional conditions for granting. Members of the committee, their employers, colleagues and family members are disallowed for submitting projects to NGI Zero. The members of the committee are not linked to NLnet in a role as employee, member of the board of directors or supervisory board.

In 2020 the independent review committee for NGI Zero consisted of:

- Guido Aben
- Sudha Bhuvaneswari
- Lucien Castex
- Marcin Cieślak
- Tommi Karttaavi
- Tessel Renzenbrink
- George Sadowsky
- Niels Sijm
- Bert Wijnen

Here are biographies of the members of the review committee.

Guido Aben

Guido Aben is director of eResearch at **AARnet**, the Australian educational and research network. He joined AARNet in 2005, having previously had similar roles in European R&E networking. A generalist more than a specialist, he has been involved across the range of the "buy-or-build" spectrum, in projects ranging from the rolling out of a national dark fibre footprint, running cloud services procurements through to the deployment of complex niche builds such as an Internet voting system during national elections. In his current role at AARNet, Guido is responsible for developing services useful to researchers. He holds an MSc in physics from **Utrecht University**.

Sudha Bhuvaneswari

Dr. N. Sudha Bhuvaneswari is an academician holding a PhD degree in Computer Science and holds a designation as Associate professor with a work experience of more than 20 years. She has authored 2 books on "Integrating SOA and Web Services" and "Combating Cyber Threat through Cyber Security Intelligence". She has also authored many chapters in IGI Global and with other publishers. She is an active member in ISOC and a fellow in APNIC 42, APrIGF 2017, APNIC 44, inSIG 2017 and inSIG 2018. Dr.N.Sudha Bhuvaneswari is also a reviewer for a number National and International Conferences and Peer Reviewed Journals.

Lucien Castex

Lucien Castex is the Secretary-General of Internet Society France and a researcher at **Université Sorbonne Nouvelle**. Policy strategist and Internet law expert, Lucien works at the intersection of law and technology with a focus on trust, cybersecurity and internet governance. He is a member of the French national consultative commission on human rights (**CNCDH**), and one of the Expert Advisors of the IoT Security Policy Platform of Internet Society. He is member of the board of Ile-de-France Region's key research sector on digital humanities and new knowledge. He is co-chair of the French

Internet Governance Forum and member of the Multistakeholder Advisory Group (**IGF MAG**) at the United Nations' Internet Governance Forum.

Marcin Cieślak

Marcin Cieślak is an an information technology consultant working with customers internationally on systems integration, project management, internetworking technologies. In addition to his work in the enterprise, he is researching various applications for a decentralized Web. He was one of the founders of **ISOC Poland**, and currently still its president. He is also a technical volunteer for Wikipedia and a contributor to the MediaWiki software. He organized the world-wide Wikipedia community gathering in Gdańsk, Poland in 2010. He commutes between Warsaw, Poland and Frankfurt am Main, Germany.

Tommi Karttaavi

Tommi Karttaavi is the Director for Information Society issues at the **Association of Finnish Local and Regional Authorities**. He has previously worked for the Finnish Ministry of Finance, Ministry of the Interior, teleoperator Elisa Communications and the Helsinki University of Technology (Aalto University) among others. He is a member of the Internet Society since 1998 and has served as Board Member and the President of the Finnish ISOC Chapter. He has also worked for the **Internet Society** as an European Chapters Develoment Manager. He has a MSc in Computer Science from the **University of Helsinki**.

Tessel Renzenbrink

Tessel Renzenbrink is board member of Internet Society Netherlands, and secretary of the board of **Gr1p**. Gr1p strives for broad civic participation in the shaping of digital society. She is a professional freelance writer and web editor focusing on the impact of technology on society, particularly on the internet and information technology and on renewable energy technologies. Her publications regularly appear in **Elektor magazine** and **Energieoverheid.nl**. She is co-founder Tessel holds an MA in Philosophy from the University of Amsterdam.

George Sadowsky

Dr. **George Sadowsky** (**Wikipedia**) is an American computer scientist who was inducted into the **Internet Hall of Fame** in 2013. On behalf of the United Nations, UNDP, UNFPA, USAID, Sida, and other organizations he has worked in more than 50 developing countries on issues interrelating economics, technology, management and policy. He is the former Executive Director of the GIPI, the **Global Internet Policy Initiative**, with projects in many transition countries to bring multiple sectors of society together to evolve Internet policy, regulation and legislation for the benefit of the country. He founded and directed the **Internet Society**'s series of network technology workshops for students from developing countries, which resulted in thousands of students being trained in Internet fundamentals, network creation, content provision, and national network management. He is the editor of and lead contributor to the World Bank's **Information Technology Security Handbook** that has been distributed worldwide, as well as the editor and lead author of the World Wide Web Foundations recent seminal publication, **Accelerating Development Using the Web: Empowering Poor and Marginalized Populations**.

Niels Sijm

drs. ing. **Niels Sijm** is the system engineer of the **System and Network Engineering** Master's program at the **University of Amsterdam**, and a freelance web technologist under the name of **IT**

Doesn't Matter. He has been building web applications for over ten years, with special interest in and care for web standards and interoperability. Niels has been working with a wide variety of people, both inside and outside IT, ranging from artists and hackers to startups, small businesses, and academia, favouring projects that contribute to society. Apart from engineering, Niels has been teaching (web) technology at various educational institutions.

Bert Wijnen

Bert Wijnen is a highly experienced and active participant in the **Internet Engineering Task Force** (IETF), where he has chaired various Working Groups. He has served as an IETF Area Director (in OPS and SUBIP). He is credited as an author on **30 RFCs**. He is a former member of the **Board of Trustees** at **Internet Society**. He is a former board member of NLnet. His working experience includes Research engineer at the **RIPE-NCC**, Senior Manager Internet Standards at Alcatel-Lucent and Senior Consulting IT Specialist at IBM (where he worked for 28 years).

2 Overview

Statutory goal and Mission

NLnet financially supports open development of information society technologies. NLnet strives to facilitate shock waves of innovation, working towards a technology commons.

The articles of association for the NLnet foundation state: *"to promote the exchange of electronic information and all that is related or beneficial to that purpose"*.

This is done through stimulating strategic technology research and development in the area of computer networking and the internet. NLnet looks at impact, so while projects may revolve around new technologies they can also focus on improving existing technology, encouraging new applications of existing technology or dissemination of relevant knowledge.

The current focus is twofold: on strenghtening the position of the individual user on the internet and on improving the overal security and robustness of the internet.

NLnet actively stimulates the development of open network-related technology and making this technology freely available to the community in the broadest sense of the word. The technology should support and contribute to a better exchange of information.

Free Software, Open Source, Open Content, Open Hardware

Throughout the years, NLnet has supported a wide range of Internet and technology related projects. A precondition for all funding is suitable 'open' licensing conditions - such as GNU GPL, BSD license, Apache License, CERN Open Hardware License, Creative Commons and such. NLnet wants the projects it supports to reach as far and wide as possible, and to have a broad future that is open to continued development well beyond its originators or originating context.

Not-for-profit

NLnet Foundation does not derive any financial benefits from projects or their results. Our focus is on societal return on investment, with a long term perspective of improving the way we live.

Any donations made in gratitude to us will be used to meet the statutory goals of NLnet.

Co-operation

NLnet maintains a warm relationship with other institutes and foundations:

- Accessibility Foundation
- AMS-IX
- Association for Progressive Communications
- Bits of Freedom
- Center for the Cultivation of Technology
- Free Software Foundation
- Free Software Foundation Europe
- GEANT
- ICANN
- iFROSS
- Internet Society & chapters
- ISPConnect
- LOT Network
- OpenForum Europe
- Petites Singularités
- Rathenau Institute
- RIPE/RIPE NCC
- SIDN/SIDN Fonds
- SURFnet
- Software Heritage
- The Commons Conservancy

- CWI
- DDA
- Digital Infrastructure NL
- DHPA
- EDRi
- NixOS Foundation
- NLnet Labs
- NLUUG
- Open Invention Network (OIN)
- OpenDoc Society
- The Hague Security Delta
- Translate House
- USENIX
- Vietsch Foundation
- W3C

Their regular activities, technical conferences, programs and occasional actions are being seen by NLnet as major forums to make its plans public, to encourage cooperation between information technology professionals and to obtain feedback from them. In addition, NLnet regularly interacts with several academic and public institutions such as the European Commission (in particular DG CNECT, DG Sante and DIGIT), Forum Standaardisatie, Netherlands Cyber Security Center and various Netherlands ministries (Ministry of Economic and Climate Affairs, Ministry of Justice and Security, Ministry of the Interior and Kingdom Relations, Ministry of Education, Culture and Research and Ministry of Foreign Affairs) and similar organisations inside and outside of Europe.

If we aren't working with you just yet, and we should – contact us.

3 Strategy and working methods

Strategic Themes

NLnet maintained and expanded focus in 2020 on the following areas of attention through thematic funds:

- NGI Zero Search and Discovery Fund
- NGI Zero Privacy & Trust Fund
- Cryptocurrency Fund
- DNSSEC
- Data Delivery Fund
- Honeypot Technology Fund
- Infrastructure & Hosting Fund
- Internet Hardening
- Internet Measurement and System Stability Fund
- Open Document Format
- Real-time communication
- Research & Education Fund
- Software Quality Fund
- Technology Awareness Fund
- VPN Fund
- NGI Assure

See for more information: **https://nlnet.nl/themes**

Third parties willing to donate to NLnet may choose to dedicate their donations to one of these themes, or to a new theme – or of course to stichting NLnet in general.

Donations and Loans

NLnet offers three types of support:

- Project donations – projects requiring not more than € 50.000 with a duration typically of eighteen months or less. If successful, follow-up projects can be submitted.
- Standalone donations – one-time sponsoring of conferences, workshops, hackathons, seminars, contests and financial compensation of travel costs for participants of these events.
- Loans – for efforts with a significant likelihood that funds spent can be returned to NLnet.

Project donations

NLnet sees a role for itself (and has a strong preference for) supporting strategic projects in the earlier parts of their lifecycle, and strenghtening existing efforts with targetted effort. Project budgets typically range between € 1000 and € 50.000, and often have a duration of eighteen months or less - but that is decided on a case by case basis. This class of project is well-suited for establishing new technologies, but innovation isn't our only game – one may also prove the desirability of sunsetting legacy technologies that no longer meet modern security and privacy requirements. NLnet's funding allows projects to deliver break-throughs in their fields, as well as do technology reconnaissance and critical investigation.

For more details on projects sponsored in 2020 see Annex 1.

Standalone donations

NLnet may choose to provide standalone donations to organisations and individuals, in order to

support and stimulate their activities - assuming these are in line with the NLnet mission and philosophy. Standalone donations also encompass incidental support for community building in the form of workshops, hackathons, conferences, setup of legal entities, and other efforts.

More details on standalone donations sponsored by NLnet in 2020 are provided in Annex 1.

Loans

Projects recieving a grant from NLnet result in free software, open content, free hardware designs and other intangible assets which are given away *pro bono.* Donations are one way traffic: NLnet does not get or expect projects to make money and pay back the donation.

In some cases, however, there is no need fro a grant but merely a temporary cash flow issue. Transient need for financial support typically occur when other sources of income – like a grant from another funding agency or a subsidy from a public institition – operate too slow, putting an organisation at risk.

In other cases there might be a suitable business model but some capital is needed to snowball the effort. When an open effort has a fit with NLnet's mission and applicants are confident that the funds requested are likely to be returned, they can ask NLnet for a loan. Loans have the advantage that the same money can be re-used over and over again for other relevant projects within NLnet's mission.

Distinctive investment

NLnet derives its yearly budgets from the available capital, the interest gained from banking of (a part of) this capital, from donations and subsidies, and some revolving activities. The challenge is or course to make sure that in the long run sufficient funding strength remains to continue its beneficial work.

Therefore NLnet decided to experiment with investing a part of our assets in technologies we understand, in people we trust and in concepts we believe will change the world to the better. And to gain money with this which can be used to accomplish the mission of NLnet.

For this purpose a few investments were made since 2012:

- Appcache Ltd ('5apps') in 2012 (currently 37,5 % equity)
- Rockstart in 2014-2016 (currently convertible loans in GAYR4 BV, GAYR5 BV, and GAYR6 BV)

4 Finances

Fiscal Status

Stichting NLnet finances its projects and activities from donations by individuals and organisations, inheritances and subsidies, as well as the annual return and interest as received on its invested capital and other assets. NLnet actively solicits donations from third parties to finance project activities, and co-sponsors projects with other organisations. A non-negotiable condition is that the independence of NLnet in choosing and financing projects is assured, and that its mission is respected.

Stichting NLnet does not derive any financial benefits from the supported projects or their results.

Since 1999, Stichting NLnet has had a non-profit tax status (so-called Article 24 status, "Algemeen Nut Beogende Instelling").

In accordance with ever changing legislation NLnet in 2007 obtained and in 2009 was confirmed its non-profit tax status (ANBI-regeling) with the Netherlands Tax Authority.

Administration

Salary administration was contracted to Cent Lonen in Haarlem.

Koningsbos Accountants in Amsterdam has been charged with compiling and auditing Stichting NLnet's Annual Accounts for 2020 and have given an unqualified opinion. The accountancy report is a separate document. The main figures are incorporated in this annual report.

Finance

Total income in 2020 (including Share of profit of associates) equalled € 2.899.209. In 2020 NLnet sponsored projects, programs and other activities to the sum of € 2.313.643. The total expenditure was € 2.861.237. The total profit therefore equals € 37.972.

The 2020 profit is calculated before the release of appropriated reserves which were formed for future funding obligations (to the sum of € 99.267). After adding this amount to the result, the total increase in equity in 2020 is € 137.239.

For a breakdown of the costs and an overview of the balance sheet we refer to the **standard public benefit organisation form** (“Standaardformulier publicatieplicht ANBI algemeen”) for the year 2020 on our website.

Annex 1: Programs, projects and activities in 2020

Programs in 2020

NLnet Labs

NLnet Labs is the Research, Development, and Expertise center for those technologies that turn a network of networks into one Internet. Established by the NLnet Foundation in 1999, NLnet Labs contributes innovative ideas to open source software and open standards. NLnet Labs is recognized for its work on DNSSEC and BGP security, as well as being the home of high-quality DNS software and tools, training and engineering efforts. NLnet Labs is led by Dr. Benno Overeinder.

Anno 2020, NLnet Labs Foundation is a fully independent and sustainable organisation that can stand firmly on its own feet. NLnet Labs has been officially recognised as a not-for-profit (ANBI, Algemeen Nut Beogende Instelling), with its own independent governance which has no overlap with that of NLnet Foundation. It collaborates with other organisations such as Verisign Labs, ICANN, SIDN and USC/ISI. NLnet Labs' work is funded by contributions from users who support its mission and want to see the maintenance and development of its software continued for everyone. An addditional source of income are software support contracts through its subsidiary Open Netlabs B.V. which also provides software development, training courses, audits and consultancy.

NLnet Foundation sponsors NLnet Labs by providing free administrative services.

The Commons Conservancy

NLnet actively contributes to **The Commons Conservancy** through a joint Memorandum of Understanding with NLnet and **Géant**. The Commons Conservancy provides a lightweight organisational structure for open projects. Its mission is to strive towards a stable democratic and open global information society in which individuals can collectively scrutinise, reconfigure and improve upon any technology they depend on - unleashing and empowering human innovation at the widest possible scale, with the express intention to empower any individual to participate in all facets of social, cultural, economic and private life under conditions of his or her own choosing and with secure and reliable technology they can have full control over themselves. The Commons Conservancy is an independent foundation.

NLnet supports The Commons Conservancy with logistics, insurance for its board members and recurring costs such as domain name registration for the foundation and its programmes.

Next Generation Internet Initiative

In 2017 and 2018, NLnet conducted a **study** to establish the **vision** of the **Next Generation Internet** initiative of the European Commission, together with Gartner Europe. As of December 1st 2018 NLnet was selected to coordinate two of the four first Research and Innovation Actions to kickstart the Next Generation Internet, an initiative by the European Commission to help shape a trustworthy, resilient and sustainable internet as part of the Horizon 2020 research and innovation program. Between 2018

and 2021 a total of 11.2 million euro is being granted by NLnet to independent researchers and open source developers. Project proposals in line with the NGI vision and the call topics can request between €5.000,- and €50.000,- with the potential to scale up after succesfully finalizing an initial project. The call topics focus on **privacy and trust enhancing technologies** and **search, discovery and discoverability**.

Projects funded through these two calls are supported by a unique coalition of not-for-profit organizations organized in the NGI Zero coalition. Together with NLnet these partners provide researchers and developers with expertise and guidance on security and code quality, accessibility (making technology available to everyone, including people with disabilities), localisation/internationalisation (to increase language diversity on the internet), packaging and reproducible builds, responsible disclosure, diversity, community building and more essential dimensions for any technology that aims to run at internet scale.

In October 2020, a third NGI research and innovation action was started under the name **NGI Assure**. NGI Assure is looking for building blocks that contribute to providing such assurances include (but are not limited to) quantum-proof cryptography, public key infrastructure, (augmented) authenticated key exchange, ratchet mechanisms (such as the Noise protocol) that securely chain key material, distributed hash tables and DAGs to make P2P interaction more secure, conflict-free replicated data types, mixnets and onion routing mechanisms, consensus protocols, distributed ledgers and (post) blockchain technologies that create redundant data sets managed independently by mutually distrustful parties, a priori usage control, symbolic and formal proofs, and tamperproof open hardware implementations of core cryptographic primitives. The work needs to become available under free and open source licenses. Projects receive support by NLnet and its partners towards standardisation efforts, improving security and business models and sustainability.

Reviewfacility.eu

On May 6th 2020 NLnet and Radically Open Security launched the Emergency Tech Review Facility on behalf of the European Commission. Reviewfacility.eu is a collaborative, community-centric effort to quickly and transparently analyze COVID-19-related technological solutions for their applicability, security and privacy characteristics. Together with Radically Open Security and voluntary contributions from privacy, security and accessibility experts the review facility detailed and scrutinized contact tracing mechanisms such as the Google Apple Exposure Notification (GAEN) framework and the European Federation Gateway Service, as well as mobile contact tracing apps used across Europe – including Italy, Poland, Estonia and the Netherlands.

The security quickscans and assessments by Radically Open Security were done together with the app developers, which is standard practice for thus non-profit security consultancy, so any discovered issues or vulnerabilities could be immediately and thoroughly discussed and addressed. NLnet together with Radically Open Security and other expert organizations continues to assess COVID-related technologies, like for example the emerging solutions to obtain and share vaccination certificates.

Received proposals

In 2020 NLnet has received in total 670 project proposals (compared to 743 in 2019), whereof 110 requests were (partially) granted (against 162 in 2019).

Projects supported in 2020

Project summaries

NLnet has provided financial support to an unprecendent amount of projects in 2020. We are proud and humbled to have so many significant and innovative free and open source software and open hardware projects consider our support. Below we give a summary of each of these projects. For each proje, there might be more in-depth project descriptions on the dedicated project pages of our website.

On the **project portfolio overview** on our website you can find all the project historically funded by NLnet, a list of **current ongoing projects**, lists of projects grouped per grant program, a **thematic index** and more.

Discover and move your coins by yourself

Connecting and sharing without a central authority giving permission is what drew users to the internet in its early days. Users could decide for themselves who they wanted to meet and what they did online. This freedom helped break down barriers, although we now know the technology can also be used to put users in a straitjacket and keep tabs on them.

The same dilemma exists today for digital currencies. Unsatisfied with how states and banks handle traditional currencies, they were promoted as a way for people to take back control over their finances. They offer an "internet first" way of make transactions without neither interference nor protection of the traditional system. Digital currencies have been around for over 35 years, but only a decade ago they became a mainstream success when some unknown person(s) put the software for "Bitcoin" online. Bitcoin left many things to be desired, though. Digital currencies are big business, and being at the right place in the right time can and has make a few people very rich. Many attempts have meanwhile been launched, including efforts backed up by large social networks, banks, filesharing technology vendors, etc.

At some point in time this turned into a real hype, resulting in millions of normal people around the world speculating on the future value of these coins. Some people got crazy enough to sell their house, often with very limited understanding of the technology or economy behind it. To make things even more confusing, in order to gain market acceptance, new coins will assign rights to people that already own older coins. So that means one moment you own a coin in one currency, and the next you own some coins in another as well. Of course, this allows for innovation - with the original author(s) of bitcoin still unknown, but also from a philosophical perspective, noone can claim to be the 'ground truth'. So in the end the real value of digital currencies is trust based, although some attempts to financially back them up somehow (so called stablecoins) are being made as well. In the absence of any oversight or proper

governance simple pyramid schemes could lead to gullible consumers losing lots of money.

The abundance of technologies and currencies is sure anyhow to leave users puzzled. Meanwhile there are thousands of offerings, many of which of which have confusingly similar names like Bitcoin Classic, Bitcoin Gold, Bitcoin Diamond, Bitcoin Atom, Bitcoin Private, etc). Many claim technical innovations in the complex blockchain backbone, and of course everyone claims to have the next big thing on their hands. There is a real snag there. If you give some online service the secrets to claim your near worthless new coin, they may be able to use that and steal money from your other coins. Users depend on trustworthy tools to handle technological components. But who will perform this community service? In the absence of such tools, greed and ignorance will drive users to shady service providers or code they happen to find on the internet. This make them vulnerable to theft or abuse.

This project wants to put users back in control and at the same time restore the key role that cryptocurrencies were created for in the first place. Speculation is not the right driver, real world usage is. Within their browser users will be able to check which coins they have and in which amounts. They can create transactions by themselves even for smaller coins and sign off on these transactions, without running the risk of someone running away with all their money. The tool should be usable and valuable for both coin holders with relatively little technological knowledge of cryptocurrencies as well as for the more technically inclined, who are free to contribute to the app and modify it in any way they see fit. Creating transparency and improving discoverability of innovative new solutions contributes to a faster and more fair convergence. And doing so in a public benefit way, leaving control in the hands of the user, is essential to help build sustainable public trust in cryptocurrencies as a valid financial system. In the end, euro's nor dollars or renmibi can be sent via email - so for the next generation internet we will need next generation payment systems.

Technical description

The numerous technologies behind cryptocurrencies are probably the most difficult to understand compared to any other networks, even for technical experts - and especially bitcoin based networks. Most users, even those familiar with the technology for years, have to rely on wallets or run/sync full nodes. Empirically we can see that they usually get lost at a certain point of time, especially when said wallets dictate the use of new "features", like bip39 and alike, multisig, segwit and bech32. Most users don't understand where their coins are and on what addresses, what is the format of these addresses and what are their seeds and what they need to unlock their coins. This situation pushes users to give their private keys to dubious services, resulting to the loss of all of their coins. The alternative is to let exchanges manage their coins, which removes their agency and puts them at risk. The goal of this project is to correct this situation allowing people to simply discover where are their coins and what are their addresses, whatever features are used. It will allow them to discover their addresses from one coin to another, rediscover their seed if they lost a part, sign/verify addresses ownership, discover public keys from private keys and create their hierarchical deterministic addresses. In fact, all the tools needed to discover and check what is related to their coins - and this for any bitcoin based network, in addition it allows them to create their transactions by themselves and send them to the networks, or just check them. The tool is a standalone secure open source webapp inside browsers that must be used offline, this is a browserification of a nodejs module that can be also used or modified for those that have the technical knowledge.

Nais - Informatique et Telecom — **Visit https://NLnet.nl/project/CoinDiscovery**

NGI0 Discovery **Cryptocurrency** **Cryptowallet** **e-Payments**

GNU Name System

Whether you want to look something up online, send an email to a friend or read the morning news, your computer panics and starts asking for help. How does it know where to retrieve or send anything? Luckily, it is connected to the domain name system. This naming system has been translating names users can remember (like ngi.eu or NLnet.nl) into numbers (or with a fancy word: addresses). Your computer has such a unique number itself, but it needs the numbers of the other computers you want to interact with to connect. You probably use domain names every day, whether you type in the address of a website, listen to a podcast or send an email.

It is called a domain name system for a reason, because it comprises more than just a naming convention. Getting a domain name involves talking to a lot of different computers. Your computer or phone basically doesn't know much about the world. One thing it does know, is how to ask that question to other, specialised computers. These computers actually also probably don't know themselves, unless they have recently answered the same question for another user. Names can change really fast for good reasons, so you would need to refresh this data a lot - otherwise users could end up on the wrong computer. The computers you sent your question to, thus pass the question on to other computers - and so forth. After just a few steps, some of the computers that were consulted get parts of the answer we were looking for. And at some point in time, the domain name system will have the entire answer. The magic happens so fast, most people are not even aware how complex this is. For them it "just works". One disadvantage: many other computers have learned something about us, about who we interact with and about our interests - in an neatly labeled way. Someone is connecting to derspiegel.de or globaleaks.com. The more unique your question, the deeper the digging inside the DNS - and the more it stands out.

Domain names are at present an critical component for users, and so also a critical point of failure and a choke point. Without functioning DNS, most people will have a hard time finding basically anything on the network of networks. There have been cases where for instance a Spanish company got their domain name taken away, even thoughwhat they did inside Europe for European citizens was legitimate here. But not in the USA. And since the organisations that handle the .org, .com and .net domain names are based in the USA, these could be forced to remove these names from the DNS.

When DNS was designed, neither security nor resilience was that much of a concern for most users. The internet in its early days was not yet 'open to the public'. This of course has changed dramatically. The massive use of the internet and thereby our dependency on DNS has highlighted very important privacy and security issues with the design of DNS. At present, it is is not always capable of preventing misleading users nor can it prevent some leakage of what users do, who they talk to and where they go.

At a larger schedule, considering resilience with regards to a perfect storm of technical and operational failure: the potential ability to somehow wreck the global DNS is a huge risk for the whole world. Having a redundant backup plan with an entirely different logic is certainly no luxury, but sane disaster

prevention.

Instead of finding workarounds or patches to fix the domain name system, the GNU project offers a complete alternative. It offers privacy and security by design with the development of a new naming system that ensures privacy protection and secure connection between people, networks and services. The goal is a future-proof, private and secure system to work and actually make the internet a safer place for users. Inside the project the developers and researchers will lower the barrier to entry of using this technology, provide a meticulous documentation of the underlying protocol and independently check how it works in the real world.

Technical description

Today, the starting point of any discovery on the Internet is the Domain Name System (DNS). DNS suffers from security and privacy issues. The GNU project has developed the GNU Name System (GNS), a fully decentralized, privacy-preserving and end-to-end authenticated name resolution protocol. In this project, we will document the protocol on a bit-level (RFC-style) and create a second independent implementation against the specification. Furthermore, we will simplify the installation by providing proper packages that, when installed, automatically integrate the GNS logic into the operating system.

GNUnet e.V. — **Visit https://NLnet.nl/project/GNS**

NGI0 Discovery **IETF** **NamingSystem** **Software**

---

f e d i v e r s e . s p a c e

NGI ZERO

A lot of the people we talk to, the media we watch and the services we search for are found in or through using social media. For users these platforms offer easy and usually free services to send public and private messages, stay updated on relevant news and promote your business or product.

But the services these social media offer do actually come at a personal and societal cost. The platforms are not neutral exchange platforms like the rest of the internet. They do not just deal with all messages they receive in the same way. Part of the corporate social network model is to give some messages preferential treatment over others, i.e. there is a noticeable bias towards those that pay. People only have so much attention they can spare every day, and the companies decide what you cannot skip based on what they get paid. This would be equivalent to you always seeing the newsletter from Coca Cola at the top of your email client, but only half of the emails from your father or local charity because they are automatically put in a folder out of sight. This "pay to play" creates a knockout race for attention fueled by commerce, not by arguments, emotions, ethics or societal considerations.

This exposure is worsened by the fact that the platforms monetize your data and behaviour. Social media companies create fine-grained personal profiles, that even include attributed political, relational and other deeply personal matters. By clustering people, profiles becomes more crisp and valuable. But they tend to push people step by step to more extreme options. You liked marijuana. You like drugs. Maybe

you like cocaine? You visited a site with conspiracy theories. Well, here is another one which is even more incredible. When these profiles are made available to advertisers at a premium price, psychometrics such as used by Cambridge Analytica (and others), these allow to influence subsets of the population in both subtle and crude ways.

These selfish business practices continuously raise fundamental societal questions: how do we feel about social media being used by foreign state actors to influence democratic elections through very personalized (and misguided) political campaigns? And how do we contain the algorithmic pressure towards global extremes, rather than brings people together as one would expect from a social network?

Another problematic issue to address is monoculture. Social networks do not allow to cross the boundary of their service in an easy way, leading to social lock in and a "winner takes all" scenario. This limits choice, but also exposes users to legal dangers. Confidential discussions through "private" messages for instance turn out to be not so private, such as the case where a United States got the social network Twitter to hand over the personal communication from European human rights activists and a member of the Icelandic parliament over a severe human rights violation by the USA military. The European Court of Human Rights would certainly not have allowed this, but it happened outside of our jurisdiction - even if all the actors never left Europe.

The federated universe, abbreviated to fediverse, wants to offer social media users a more transparent, ethical and decentralized environment to talk, find and connect. This is done through a plethora of completely independent servers hosted by organisations and individuals around the world. Each has their own policy, each has their own community and reputation. But they can all interoperate. If you don't like any of the existing options, or want to do something different or innovative, you download some open source software and start your own. If you feel some server is toxic, or misbehaves, it just takes one click to stop listening to what is being said. And there is no need to share data with anyone, if you want to. Every node can essentially be a complete social network in itself.

The fediverse is not confined to what a single company wants to do - in every way. That means a broader offering in terms of design, usability and user experience, in terms of technology, ethics and culture. Essentially every server is a full-fledged social network in itself, able to talk to other social networks when it wants. People can use the fediverse for traditional social networking, but they can also integrate it with other services such as online video sharing, all without the fear of having their data being monetized or their activity profiled. Switching from closed social networks to the fediverse contributes to privacy and trust, by enabling users to understand and control who sees their data. The fediverse as a network of social networks, is also more resilient than a single network could ever be.

Fediverse.space will help to discover where discussions and communities that interest users take place. This is an essential feature for a decentralised technology. Searching among the different social networks on the fediverse is still in its infancy, and behaves different from traditional document based search. Fediverse.space visualizes and allows to categorise the hosted servers in different ways, and even analyse trends. For example you can see most discussed topics, primary language used, or any other category that helps users find the network most interesting for them to join. Such a tool improves the discoverability and usability of alternative social environments and can help users switch from the traditional commercial social media to the next generation of open social networks.

Technical description

Fediverse.space is a tool for understanding decentralized social networks, and searching through them. The fediverse, or federated universe, is the set of social media servers, hosted by individuals across the

globe, forming a libre and more democratic alternative to traditional social media. When displaying these servers in an intuitive visualization, clusters quickly emerge. For instance, servers with the same primary language will be close to each other. There are more subtle groupings, too: topics of discussion, types of users (serious vs. ironic), and political leanings all play a role. fediverse.space aims to be the best tool for understanding and discovering communities on this emerging social network.

Visit https://NLnet.nl/project/fediverse_space

NGI0 Discovery **Discovery** **SocialNetworking** **Visualisation**

StreetComplete

Everyone needs to find their way around the world, be it traveling for work, taking a vacation or going to the doctor, dentist, your local municipality and other important (public) services. How we move around and where we go is very personal information: imagine following someone for a week and what this can teach you about their life, their loved ones and what is important to them. Now think about the apps or devices you use for navigation and what they can and probably do log about you. Where does this information go, who has access to it, how does this feed into your data profile that is created and sold by tech platforms to businesses (and sometimes governments)?

Navigation shouldn't be yet another underhanded means for tracking and profiling, it should help you get to where you need to be and inform you about your travel, nothing else. OpenStreetMap is a collective effort to build a tool that brings geographic data and navigation into the public space, as an alternative to commercial services. Users help map areas, roads, buildings and other points of interest and keep this information up to date and enrich it. All data is open and free to use.

Navigation works best when you can efficiently search and find exactly where you need to go, even with limited information. StreetComplete is a project that makes it easier for users of OpenStreetMap to optimize, correct and enrich geographical data without the need for technical knowledge or skills. Simply walk around an area and answer survey style questions like "What is the name of this road, what are the opening hours of this shop, is there a cycleway here?" This way users can help keep the geographic data of OpenStreetMap up to date and enrich it with valuable information, for example on the wheelchair accessibility of a street or building. This makes OpenStreetMap a more valuable and more inclusive source for geographic search and navigation.

Technical description

The project will make collecting data for OpenStreetMap easier and more efficient. OpenStreetMap is the best source of information for general purpose search engines that need a geographic data about

locations and properties of various objects. The objects vary from cities and other settlements to shops, parks, roads, schools, railways, motorways, forests, beaches etc etc etc. The search engine can use the data to answer queries such as "route to nearest wheelchair accessible greengrocer", "list of national parks near motorways" or "London weather". Full OpenStreetMap dataset is publicly available on an open license and already used for many purposes. Improving OSM increases quality of services using open data rather than proprietary datasets kept as a trade secret by established companies.

Visit https://NLnet.nl/project/StreetComplete

NGI0 Discovery **OSM** **OpenData**

Geographic tagging and discovery of Internet Routing and Forwarding

SCION

More and more commercial and public services are digitized and even completely replaced by online platforms and communication channels. For users, businesses and governments, this raises important questions regarding privacy and security. The data created and shared in or between hospitals, banks, companies or municipalities can be sensitive, deeply personal and potentially harmful when it ends up in the wrong hands. All manners of security and privacy protection walls are put in place to prevent this from happening, which nevertheless do not prevent regular and large scale data leaks that sometimes cause real damage to people.

Instead of creating workarounds and improvising defenses for vulnerable points in the route information travels on the internet, SCION instead wipes the slate clean and designs an internet that is secure and private by design. This alternative internet architecture offers users (as well as internet service providers or ISPs) overview and control of their online communication. SCION can ensure path-aware communication where only senders, receivers and ISPs are allowed to set the rules for precisely how internet traffic should be routed across networks and servers, protecting the privacy of everyone involved and ensuring users and hosts have high availability, bandwidth and little to know vulnerabilities to for example DDoS attacks.

This project wants to give users and network administrators even more control over how data and communication travels by adding specific geographic data to the road traveled. For example, where is the datacenter or internet exchange point that this particular internet route uses located? This is especially relevant for data and information that is commercially, politically or technically sensitive and that should not end up in a particular country or be forwarded by a party that is known to snoop or censor. In these cases it may not be enough to construct the most optimal or available route, but also the best protected and trustworthy one.

Technical description

SCION is the first clean-slate Internet architecture designed to provide route control, failure isolation, and explicit trust information for end-to-end communication. As a path-based architecture, SCION end-hosts learn about available network path segments, and combine them into end-to-end paths, which are

carried in packet headers. By design, SCION offers transparency to end hosts with respect to the path a packet travels through the network. This has numerous applications related to trust, compliance, and also privacy. By better understanding of the geographic and legislative context of a path, users can for instance choose trustworthy paths that best protect their privacy. Or avoid the need for privacy intrusive and expensive CDN's by selecting resources closer to them. SCION is the first to have such a decentralised system offer this kind of transparency and control to users of the network.

Anapaya Systems — **Visit https://NLnet.nl/project/SCION-geo**

NGI0 Discovery **Geo-localisation** **Metadata** **Routing**

P l a u d i t

Should the findings of scientific research funded by public means be publicly available? Today we have limited access to the papers and academic articles that researchers write while working at universities supported in part by taxes we all pay. Publishers and publication platforms put most scholarly content behind (very) expensive paywall that usually only the same tax-funded universities can actually afford. The call for open access now grows louder, with scientists, journalists and activists arguing that scientific knowledge should be available for the common good and educate people, inspire innovation and be an important voice in an age of "fake news" and misinformation.

Plaudit supports open academic access by providing a tool scientists can use to independently endorse valuable and important research. This signals readers what articles are reliable and relevant, strengthens the credibility of researchers who become less dependent on major journals and supports the platforms that actually provide open access with an authentic stamp of approval. The tool itself is easy to use and integrate in for publishers and works with identifiers for articles and researchers that are already commonplace in the academic field. This project aims to integrate the Plaudit tool into preprint servers (where work is published before formal peer review and publication in a journal) and academic journals to encourage scientists to endorse relevant work and to channel these endorsements to other researchers, journalists, funders and anyone interested in academic work that is relevant for their interests. Scientists can take back their agency to determine what scientific work is relevant and users, journalists, businesses and governments can learn from these insights and innovate.

Technical description

Plaudit is open source software that collects endorsements of scholarly content from the academic community, and leverages those to aid the discovery and rapid dissemination of scientific knowledge. Endorsements are made available as open data. The NGI Search & Discovery Grant will be used to simplify the re-use of endorsement data by third parties by exposing them through web standards.

Visit https://NLnet.nl/project/Plaudit

Blink RELOAD

One of the things people enjoy the most about the internet, is that it enables them to talk to others remotely almost without limit. Internet allows anyone to keep closely connected with friends and family, and help their kids solve a math problem while they are at work. People collaborate with their colleagues from the couch of their living room, the cafe where they enjoy lunch or on their cell phone on the bus to the gym. Businesses can easily service their customers where this is most convenient to them, without having to travel themselves. This is so convenient, that some businesses have already moved entirely online. Internet communication has become the nerve center of whole neighbourhoods, where people watch over the possessions of their neighbours while these are away for work or leisure.

Connecting and communicating online begins with search and discovery and this is a crucial starting point that users should be able to trust. Ask yourself this: how do you find the person you want to talk with and how can you be sure they are who they say they are? And who logs what names and addresses you are looking for and how long your call was, where you connected from and with whom? Can you still be reached in any other way when your account is removed? Not all service providers give you clear answers to these very relevant questions. And probably the right answers are even more important in less democratic societies where this type of information is critical to bring down opposition and stifle human rights.

With Blink users do not have to rely on proprietary services and can instead use proven internet standards to setup a call, have a video chat, send instant messages or share their screens. Communication is encrypted from user to user to avoid anyone listening in. And to make searching and discovering contacts even more independent from third parties and infrastructure, this project will implement a powerful set of new internet standards. These establish an abstract storage and messaging service between a set of end users which can be discovered locally. These new allow Blink users to directly find, connect and communicate to each other in a peer-to-peer network. The network can be made up entirely of users, without relying on servers that may not be familiar or trustworthy. Search and discovery becomes more straightforward, transparent and reliable.

Technical description

REsource LOcation And Discovery specification (RELOAD) is a standard produced by the IETF standard to (as the name indicates) describe how people can search within a local network to discover other people and devices they can then exchange video and voice calls with, send messages etc. Why make every discovery depend on the availability of a global DNS system, if you are actually near each other...

Blink is a mature open source real-time communication application that can be used on different operating systems, based on the IETF SIP standard. It offers audio, video, instant messaging and desktop

sharing. Blink RELOAD aims to implement RELOAD (RFC 7904) , which describes a peer-to-peer network that allows participants to discover each other and to communicate using the IETF SIP protocol. This offers an alternative discovery mechanism, one that does not rely on server infrastructure, in order to allow participants to connect with each other and communicate. In addition, the RELOAD specification describes means by which participants can store, publish and share information, in a way that is secure and fully under the control of the user, without a third party controlling the sharing process or the information being shared.

Visit https://NLnet.nl/project/BlinkRELOAD

NGI0 Discovery **Contacts** **E2EE** **IETF** **P2P** **Videocalling** **Voicecalling**

Decentralized privacy preserving search by mathematical design

NGI ZERO

Search and discovery are some of the most important and essential use cases of the internet. When you are in school and need to give a presentation or write a paper, when you are looking for a job, trying to promote your business or finding relevant commercial or public services you need, most of the time you will turn to the internet and more importantly the search bar in your browser to find answers. Searching information and making sure your name, company or idea can be discovered is crucial for users, but they actually have little control over this. Search engines decide what results you see, how your website can be discovered and what information is logged about your searches. What filters and algorithms are are used remains opaque for users. They can only follow the rules laid out for them, instead of deciding on their own what, where and how to find the information they are looking for.

This project takes a radically different approach to privacy-friendly and decentralized search and discovery by ensuring that the search engine is decentralized by design. Even the current search engines don't run the work on a single machine - the web is way to big for that. So it is already a distributed task. By harnessing the combined power of many small systems made available by their users, such a collective approach can measure up to the services traditionally delivered by monolithic search companies. The reverse also holds: without enough participants working together and contributing, the use will likely be limited. For now, the first concern is the availability of the essential technical building blocks. Of course, a decentralized search engine should still provide users with relevant results that are at least on par with what proprietary search algorithms can offer, which this project aims to do with machine learning.

The P2P nature has attractive features. It ensures users that there can be no central point of control, not now nor in the future. It is also potentially extremely robust and resilient: the computers contributing capacity are potentially widely spread across the internet. If for some reason like a major disaster, the European internet is cut off from the rest, P2P search may still work.

Technical description

Today, search engines are dominated by centralized services offered by big companies. This leads to the big problem of data centralization. With this project we want to develop a search engine concept which is decentralized by its mathematical design. Since, the biggest problem of alternative search engines is that they have to compete with the well known big players,and their well tuned algorithm. Apart from the privacy preserving aspects needed, it is also important to develop a good working search algorithm. For this, the project will develop its algorithms on the basis of machine learning concepts, which are closely connected to privacy preserving parts of the algorithm. The basic idea here is that every participant works as a machine learning neuron within a cluster concept of neurons, similar to results from biological brain research, in the decentralized network. Information is partitioned and distributed by threshold secret sharing. The project will investigate and try out what the practical options for encryption are in such a scenario.

Visit https://NLnet.nl/project/combsee

NGI0 Discovery **Decentralised** **P2P**

Sonar: a modular peer-to-peer search engine for the next-generation web

Sonar

Search and discovery are some of the most important and essential use cases of the internet. When you are in school and need to give a presentation or write a paper, when you are looking for a job, trying to promote your business or finding relevant commercial or public services you need, most of the time you will turn to the internet and more importantly the search bar in your browser to find answers. Searching information and making sure your name, company or idea can be discovered is crucial for users, but they actually have little control over this. Search engines decide what results you see, how your website can be discovered and what information is logged about your searches. What filters and algorithms are are used remains opaque for users. They can only follow the rules laid out for them, instead of deciding on their own what, where and how to find the information they are looking for.

This project takes a radically different approach to privacy-friendly and decentralized search and discovery by ensuring that the search engine is decentralized by design. Even the current search engines don't run the work on a single machine - the web is way to big for that. So it is already a distributed task. By harnessing the combined power of many small systems made available by their users, such a collective approach can measure up to the services traditionally delivered by monolithic search companies. The reverse also holds: without enough participants working together and contributing, the use will likely be limited. For now, the first concern is the availability of the essential technical building blocks. Of course, a decentralized search engine should still provide users with relevant results that are at least on par with what proprietary search algorithms can offer, which this project aims to do with machine learning.

The P2P nature has attractive features. It ensures users that there can be no central point of control, not

now nor in the future. It is also potentially extremely robust and resilient: the computers contributing capacity are potentially widely spread across the internet. If for some reason like a major disaster, the European internet is cut off from the rest, P2P search may still work.

Technical description

Sonar is a project to research and build a toolkit for decentralized search. Currently, most open-source search engines are designed to work on centralized infrastructure. This proves to be problematic when working within a decentralized environment. Sonar will try to solve some of these problems by making a search engine share its indexes incrementally over a P2P network. Thereby, Sonar will provide a base layer for the integration of full-text search into peer to peer/decentralized applications. Initially, Sonar will focus on integration with a peer-to-peer network (Dat) to expose search indexes securely in a decentralized structure. Sonar will provide a library that allows to create, share, and query search indexes. An user interface and content ingestion pipeline will be provided through integration with the peer to peer archiving tool Archipel.

arso project — **Visit https://NLnet.nl/project/Sonar**

NGI0 Discovery **DAT** **Decentralised** **P2P**

D e l t a B o t

One of the things people enjoy the most about the internet, is that it enables them to talk to others remotely almost without limit. Internet allows anyone to keep closely connected with friends and family, and help their kids solve a math problem while they are at work. People collaborate with their colleagues from the couch of their living room, the cafe where they enjoy lunch or on their cell phone on the bus to the gym. Businesses can easily service their customers where this is most convenient to them, without having to travel themselves. This is so convenient, that some businesses have already moved entirely online. Internet communication has become the nerve center of whole neighbourhoods, where people watch over the possessions of their neighbours while these are away for work or leisure.

Tools like Whatsapp, Signal and Telegram has become a mainstay for individuals, businesses and even local governments as a low-threshold channel to reach out to people, be it for a friendly chat or customer support. The services promise their users that everything they share and discuss is shielded off from spying eyes. Nothing is said about the metadata that shows who talks to who, and where they are. And these still suffer from issues of centralized services maintained by one party, like censorship and country-wide bans. A Signal user cannot communicate with a Telegram user through either service. And all of them can be blocked easily.

Actually secure, private and decentralized chat is important to offer users but also businesses and governmental organizations a transparent and trustworthy communication channel. This is especially the case when sensitive and personal data is shared and even more so for people living in less democratic societies who run the risk of being arrested or harassed for who they talk to or what they say. Everyone has the right to confide in someone, be it a friend or a professional, and be sure what is said does not leave the (virtual) room. For journalists, activists, whistle-blowers and vulnerable minorities, this right can be a matter of life or death.

Delta Chat upholds this universal right to privacy. Instead of using a new technology that can be easily blocked, it uses the most widely used and decentralized messaging system on the internet, e-mail. That means Delta Chat is able to share encrypted messages without a central server that stores metadata or login where someones credentials can be recorded. Users can chat with anyone who has an email address: the recipient does not need to have the Delta chat app to reply. They simply reply to the e-mail they receive. With funding from NGI Zero new features for search and discovery of content elsewhere are added, and connections are made to other decentralized messaging platforms. Users can more easily find new content and interesting contacts while being sure that their search terms or results are not logged by any third party somewhere along the way. Users should not (and usually will not) give up functionality for security and privacy. This project can help make make decentralized messaging services like Delta Chat become more versatile and attractive chat alternatives.

Technical description

Why make humans be the only ones to search new content that is relevant to you, if bots can be made to do the same on your behalf? The DeltaBot project will research and develop decentralized, e2e-encrypting and socially trustworthy bots for Delta Chat (https://delta.chat). Bots will bridge with messaging platforms like IRC and Matrix, offer media archiving for its users and provide ActivityPub and RSS/Atom integration to allow users to discover new content. Our project is not only to provide well tested and documented Chat Bots in Python but also help others to write and deploy their own custom bots. Bots will perform e2e-encryption by default and we'll explore seamless ways to resist active MITM attacks.

merlinux GmbH — **Visit https://NLnet.nl/project/DeltaBot**

NGI0 Discovery **Agents** **Discovery** **SocialNetworking**

Software vulnerability discovery

Software security for many users is a given, an assumption, something you do not and should not have to think about too hard. If you open an app on your phone, install new software on your laptop or boot up your tablet, you assume the software you use is safe, secure and that the developers have done their job right. With the amount of software coming out and the tangled web of inter-dependencies that exist today, this assumption of trust is hard to live up to. Especially since software vulnerabilities are constantly hunted for by malicious parties that want to get into

our data and devices for blackmail, theft or on a larger and more dangerous scale, disruption of vital processes like power grids.

One of the ways to make sure users do not have to worry about the applications they have installed is to automate the search and discovery of software vulnerabilities. Detecting and fixing security risks automatically can help to mitigate vulnerabilities that were recently uncovered by vendors and developers. Of course there is little that can be done about so called zero-day exploits, but as soon as a problem is known developers typically start working on fixing their software. As a user you want to get those fixes as soon as possible, because the fact that a problem is now public increases the attack surface of software that companies, governments and people use to share sensitive data. Criminals can read bug reports too, and can opportunistically seize the chance to move in.

This project helps to make the internet more safe by shortening the path between software releases and the users. Installing a piece of software on a server or computer is quite simple these days. But behind the software repositories with tens of thousands of software applications, hides a lot of work and logistics. This is because a computer application typically isn't a single self-contained program, but assumes a lot of other software to be present on the computer. This helps to save your harddisk from having many copies of exactly the same file in different places, which is not just ecological waste but also a security liability. These so called "dependencies" need to be taken into account. A security issue in a major dependency can cause a lot of other application to be insecure.

So why does this need any work at all? Well, these dependencies are all independently produced by individual developers, small and large companies and communities. A significant human effort is required to monitor all kinds of software archives around the world for new versions. When a new version is discovered, so called packagers need to manually perform a number of tasks to arrive at the point where normal users can just install an update. Nixpkgs-update automatically discovers and updates software packages, and the Nix packaging system makes sure all the dependencies are properly handled.

With funding from NGI Zero this project will extend its efforts to automatically search for reported vulnerabilities in software packages, and make sure that updates which solve these issues are communicated and prioritised. The result is that users will be deploying and using the latest versions of software quicker, and can automatically install critical updates 24/7. If you are a company running a server on the public internet, that is critically important for your security and that of the rest of the net. As such, the project contributes to an operational internet that is more responsive to threats. We all need to be able to trust that the software we use is the latest, most reliable version that can be had. This project makes it possible to deliver on that assumption.

Technical description

nixpkgs-update automates the updating of software packages in the nixpkgs software repository. It is a Haskell program. In the last year, about 5000 package updates initiated by nixpkgs-update were merged. This project will focus on two improvements: One, developing infrastructure so that the nixpkgs-update can run continuously on dedicated hardware to deliver updates as soon as possible, and Two, integrating with CVE systems to report CVEs that are addressed by proposed updates. I believe these improvements will increase the security of nixpkgs software and the NixOS operating system based on nixpkgs.

Visit https://NLnet.nl/project/nixpkgs-update

NGI0 Discovery **Automation** **CVE** **SecurityUpdates**

Users have a right to internet access and should be sure that the rights they have offline are also protected online. The internet is not just a technology or a communication medium anymore, as these declarations from the United Nations show, it has become a crucial building block of our society, economy, democracy and the way we work, live and come together. And just like in the real world, this means that there should be a public space online, a digital commons: places where everyone has free access to knowledge, can join the discussion and exercise their basic human rights. In these spaces, users should be sure they can move and act freely, without worrying about being monitored, paying for access or being categorized or treated differently.

In Common is an initiative of civic activists and organizations to help people connect with public digital commons and learn them use free and decentralized communication tools, like for example the social media service Mastodon that users can host themselves. This is a collaborative effort that can benefit from more cooperation and sharing of relevant services and knowledge. People should be able to easily search for and discover public spaces in their area to enjoy a good talk or discussion, learn more about their neighborhood and how they can help and pick up some cool and useful open source services along the way. Through this project In Common wants to create an interactive map of the commons where civil activists and organizations can safely, easily and both on- and offline find each other and add relevant information. Tying public spaces closer together can make internet users more aware of the importance of (digital) commons and ultimately strengthen their online agency and data governance.

Technical description

IN COMMON emerged as a transnational European collective from a network of non-profit actors to identify, promote, and defend the Commons. We decided to start a common pool for Information Technologies with the aim to create, maintain, and share with the public geo-localized data that belong to our constituents and to articulate citizen movements around a free, public and common platform to map and act together for the Commons. IN COMMON forms a cooperative data library that provides collective maintenance to ensure data is always accurate.

Reseau Transition — **Visit https://NLnet.nl/project/InCommon**

NGI0 Discovery Cooperative Discovery Geo-localisation OSM

Search and discovery is one of the most important and essential use cases of the internet. When you are in school and need to give a presentation, when you are looking for a job, trying to promote your business or finding relevant commercial or public services you need, most of the time you will turn to the internet and more importantly the search bar in your browser to find answers. Searching information and making sure your name, company or idea can be discovered is crucial for users, but they actually have little control over this. Search engines set the terms for what results you see, how your website can be discovered and what information is logged about your searches. What terms are set remains obscure for users and they can only follow the rules laid out for them, instead of deciding on their own what, where and how to find the information they are looking for.

More transparent, customizable and privacy-friendly search puts the user in the driver seat and can provide them with meaningful results. Searx does this by aggregating results from more than 70 search services while avoiding any user tracking or profiling. With every search users can decide what engines they want to use and which they don't, what search language must be used and other options that are saved on the device and can therefore not be tracked. Users are also free to run their own instance of Searx, giving them complete control over the source code that makes that version of Searx tick (and alter it however they like) and ensure additional privacy protection.

This project gives Searx users even more control over what their own rules for search and discovery, in particular discoverability of sensitive or personal information. Right now Searx only searches on the internet and does not look for information on for example the computer you use. Instead of users having to upload information to makie it findable (and giving away control over where the data will end up and who gets to see and use it), Private Searx allows users to find results both online and offline on their local computer or network from the same search bar.

Technical description

Searx is a popular meta-search engine letting people query third party services to retrieve results without giving away personal data. However, there are other sources of information stored privately, either on the computers of users themselves or on other machines in the network that are not publically accessible. To share it with others, one could upload the data to a third party hosting service. However, there are many cases in which it is unacceptable to do so, because of privacy reasons (including GPPR) or in case of sensitive or classified information. This issue can be avoided by storing and indexing data on a local server. By adding offline and private engines to searx, users can search not only on the internet, but on their local network from the same user interface. Data can be conveniently available to anyone without giving it away to untrusted services. The new offline engines would let users search in local file system, open source indexers and data bases all from the UI of searx.

Visit https://NLnet.nl/project/PrivateSearx

NGI0 Discovery **LocalSearch** **MetaSearch** **PersonalArchive**

---

elRepo.io - Resilient, human-centered, distributed content sharing and discovery.

Culture is the glue that ties global and local communities together. The words people use, food they share and songs they sing makes up a collective language that says 'this is who we are, this is what our culture means to us'. To satisfy this very human need, communities need some common infrastructure that brings its members together in a space where they can share their culture, without any (commercial or governmental) interference. Internet technology can help create such a public space, a digital commons, through networks that communities host themselves and use to share their culture among peers.

elRepo.io can help communities everywhere build their own networked home where they can safely store and share audio, video, text and other file formats. The network is resistant to any form of central censorship and can even be used when internet connectivity is down, as content is stored and can be exchanged locally. To make such networks become more relevant to, for example new members of a community, this project will add a content search function to the existing distributed peer-to-peer platform.

Technical description

In this project AlterMundi and NetHood collaborate to develop a critical missing part in decentralized and distributed p2p systems: content search. More specifically, this project will implement advanced search for elRepo.io, the self-hosted and distributed culturesharing platform currently under active development by AlterMundi and partners. Search functionalities will expand on the already proven coupling of thelibxapian searching and indexing library and turtle routing. The distributed search functionality will be implemented to be flexible and modular. It will become the meeting point of three complementary threads of on-going work: Libre technology and tools for building Community Networks (LibreRouter & LibreMesh), fully decentralized, secure and anonymous Friend2Friend software (Retroshare), and a transdisciplinary participatory methodology for local applications in Community Networks (netCommons).

NetHood + AlterMundi — **Visit https://NLnet.nl/project/elrepo_io**

NGI0 Discovery **LocalSearch** **Multimedia** **P2P**

s e a r x

Search and discovery is one of the most important and essential use cases of the internet. When you are in school and need to give a presentation, when you are looking for a job, trying to promote your business or finding relevant commercial or public services you need, most of the time you will turn to the internet and more importantly the search bar in your browser to find answers. Searching information and making sure your name, company or idea can be discovered is crucial for users, but they actually have little control over this. Search engines set the terms for what results you see, how your website can be discovered and what information is logged about your searches. What terms are set remains obscure for users and they can only follow the rules laid out for them, instead of deciding on their own what, where and how to find the information they are looking for.

More transparent, customizable and privacy-friendly search puts the user in the driver seat and can provide them with more meaningful results. Searx does this by aggregating results from more than 70 search services while avoiding any user tracking or profiling. With every search users can decide what engines they want to use and which they don't, what search language must be used and other options that are saved on the device and can therefore not be tracked. Users are also free to run their own instance of Searx, giving them complete control over the source code that makes that version of Searx tick (and alter it however they like) and ensure additional privacy protection.

This project can make Searx an even more customizable transparent search alternative by working towards its first 1.0 release while addressing user suggestions and feature requests. There will also be effort put into preparing Searx for wider deployment to show users they have more options and agency when searching for what they need online.

Technical description

Searx (/sɜːrks/) is a free metasearch engine, available under the GNU Affero General Public License version 3, with the aim of protecting the privacy of its users. Across all categories, Searx can fetch and combine search results from more than 80 different engines. This includes major commercial search engines like Bing, Google, Qwant, DuckDuckGo and Reddit, as well as site-specific searches such as Wikipedia and Archive.is. Searx is a self hosted web application, meaning that every user can run it for themselves and others - and add or remove any features they want. Meanwhile, numerous publicly accessible instances are hosted by volunteer organizations and individuals alike. The project will consolidate the many suggestions and feature requests from users and operators into the first full-blown release (1.0) for Searx, as well as spend the necessary engineering effort in making the technology ready for even wider deployment.

Visit https://NLnet.nl/project/SearxRelease

NGI0 Discovery Metasearch Proxy Self-hosted

Transparency Toolkit

When you get up in the morning, and read a fine piece of investigative news about a financial scandal, you don' t really stop to think much about how news is produced and what the human cost of its production is. Every year, dozens of journalists around the world get killed, because of what they write and who they talk to. Even in democratic countries, people can run the risk of intimidation and retribution. If you happen to be a courageous journalist writing about corruption, gangs or some other social wrong, protecting your sources is more than a matter of principle - it can be a matter of life and death for all parties concerned. So journalists and other vulnerable groups like civil society groups need to be very careful.

But at the same time they of course need to collaborate. Investigative reporting it is often the combined intelligence and data gathering of many that allows them to see otherwise invisible or indiscernible patterns. That means people will have to deal with significant if not massive amounts of documents and data. As a collective, they need to find their way inside these materials to discover the information they need. But of course no conventional search engine can help them, because the resources they have are not all public and could actually cause real trouble to for instance whistleblowers inside corrupt institutions should they leak to the wrong people.

Transparency Toolkit provides journalists, activists and other actors that need to control their communication with a closed off searchable database within their browser. Users can setup their own database and fill it with various documents and file formats which contents can be further analyzed and searched. To make these databases even more resistant to censorship, the archived documents will be stored across various locations to avoid central points of failure.

Technical description

Transparency Toolkit is building a decentralized hosted archiving service that allows journalists, researchers, and activists to create censorship-resistant searchable document archives from their browser. Users can upload documents in many different file formats, run web crawlers to collect data, and manually contribute research notes from a usable interface. The documents are then OCRed (when needed) and indexed in a searchable database. Transparency Toolkit provides a variety of tools to help analyze and understand the documents with text mining, searching/filtering, and manual collaborative analysis. Once users are ready, they can make some or all of the documents available in a public searchable archive. These archives will be automatically mirrored across multiple instances of the software and the raw data will be stored in a distributed fashion.

Visit https://NLnet.nl/project/TransparencyToolkit

NGI0 Discovery Crawling P2P PersonalArchive Scraping

It has been several decades since the first internet connection was made and we still have not solved the issue of free, safe and controlled file sharing. Common channels like email set strict file size limits and leave possibly sensitive data strewn about inboxes and servers. File hosting and sharing services keep users in the dark about what happens to their uploads and do not keep files up for long. Torrent environments are fraught with illegally uploaded or malicious content that may be harmful for users, who have no tools to verify or authenticate anything or anyone.

To solve this issue, users should be able to transparently host and share files, control access to uploaded content and know precisely where there files are online. The alternative internet architecture SCION, short for (Scalability, Control, and Isolation on Next-Generation Networks), offers users (as well as internet service providers or ISPs) the overview and control this requires. SCION can ensure path-aware communication where only senders, receivers and ISPs are allowed to set the rules for how internet traffic should be routed across networks and servers.

The goal of this project is to use this transparency of internet traffic for hosts and users through SCION for decentralized data storage and retrieval. Combining the level of control, privacy protection and overview of path awareness and construction with hosting and sharing files that are decentralized and not held or maintained by a single party gives the user full agency over their data and access to it.

Technical description

With the amount of downloadable resources such as content and software updates available over the Internet increasing year over year, it turns out not all content has someone willing to serve all of it up eternally for free for everyone. And in other cases, the resources concerned are not meant to be public, but do need to be available in a controlled environment. In such situations users and other stakeholders themselves need to provide the necessary capacity and infrastructure in another, collective way.

This of course creates new challenges. Unlike a website you can follow a link to or find through a standard search engine and which you typically only have to vet once for security and trustworthiness, the distributed nature of such a system makes it difficult for users to find the relevant information in a fast and trustworthy manner. One of the essential challenges of information management and retrieval in such a system is the location of data items in a way that the communication complexity remains scalable and a high reliability can be achieved even in case of adversaries. More specifically, if a provider has a particular data item to offer, where shall the information be stored such that a requester can easily find it? Moreover, if a user is interested in a particular information, how does he discover it and how can he quickly find the actual location of the corresponding data item?

The project aims to develop a secure and reliable decentralized storage platform enabling fast and scalable content search and lookup going beyond existing approaches. The goal is to leverage the path-awareness features of the SCION Internet architecture to use network resources efficiently in order to achieve a low search and lookup delay while increasing the overall throughput. The challenge is to select suitable paths considering those performance requirements, and potentially combining them into a

multi-path connection. To this end, we aim to design and implement optimal path selection and data placement strategies for a decentralized storage system.

OVGU Magdeburg — **Visit https://NLnet.nl/project/SCION-Swarm**

NGI0 Discovery **DataPlacement** **Decentralised** **P2P**

---

i p f s - s e a r c h . c o m

On the web it can happens that someone who is hosting a small website that is valuable to you no longer wishes operate it. Or is unable to bear the costs. Or is censored. She turns off the server without warning (or someone does it for her). And all the information ever submitted is lost. If that information is for instance not just some web app, but an original resource that is valuable from a personal, cultural or historical point of view, that feels like a waste. We often learn a lot while browsing through old books, brochures, news papers and advertisements - but also from resurfacing our own notes from the past. Surely, valuable institutions like the Internet Archive can have potentially archive anything. But they won't be able to capture everything, and furthermore an archive is a frozen state. What if this was actually a community resource that you a user contributed a lot of time and content to, and have a stake in wanting to keep it alive? Wouldn't it be great if you as a contributor had the "source" all along, and were able to bring it back to life?

IPFS, or the InterPlanetary File System, is different from the original web that assumes and thus assigns all the rights to a single server and owner. It is like a giant virtual storage cluster spread across the computers of all the people that use it. You can use it to publish things, such as web pages. One thing is different: as long as there are people that retain a copy of those pages, it will remain available. IPFS does not per se have a central place of storage, any copy is enough to give others access to this data by sharing links to the content. That means you as a user can help curate the cultural, social, technical heritage you care about. The approach is similar to torrenting, but IPFS offers more transparency and security since users can look at past versions of edited data and content is uniquely identified and permanently stored.

One significant issue though for users is that conventional search engines at present do not engage with the InterPlanetary File System just yet. This is because it is challenging to work with such a distributed system, especially for a search engine. It is a constantly changing, complex and vibrant space, very much like the real world is. Easily finding things you need is vital to making any information systm usable, especially if the system in question is huge (and "interplanetary" definitely has some ambitions there).

This is where the project ipfs-search.com comes in. It aims to let users search the Interplanetary File System through a regular search engine interface, that is nevertheless distributed over the IPFS-system itself. The actual indexing of files, which is automatically done by something called a crawler, can also be decentralized. This would ensure that search and discovery on IPFS would not become a single point of failure that can get taken down, attacked or censored. But actually belongs to and is operated by the

same people that make the IPFS possible in the first place, namely its users.

Technical description

ipfs-search.com is a Free and Open Source (FOSS) search engine for directories, documents, videos, music on the Interplanetary Filesystem (IPFS), supporting the creation of a decentralized web where privacy is possible, censorship is difficult, and the internet can remain open to all.

ipfs-search — **Visit https://NLnet.nl/project/IPFS-search**

NGI0 Discovery Crawling Decentralised IPFS Indexing P2P

L i b r e - S O C

libreSOC

Consumers and businesses overpay for computer hardware, because the market is not working well. When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component, the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

This ambitious project wants to deliver the first completely open computer processor in history - one you don' t have to merely trust, because you can verify and modify everything about it. All of the technology included, from top to bottom, will become available for inspection, and can be tuned by anyone technically capable enough. In practical terms, this will mostly be done by experts, but there is one major difference: anyone can become an expert, and noone has to ask permission to experiment. This will significantly contribute to the creation a new generation of computer technologies, as well as more energy efficient and cheaper devices.

Technical description

It is 2019 and it is not possible to buy a mass-produced laptop, tablet or smartphone and replace all of its software (with software that a user can trust) without loss of functionality. Processor boot-loaders are DRM-locked; WIFI, 3D Graphics and Video Processors are proprietary, and Intel's processors contain problematic features and intransparent elements such as the "Management" Engine. The most logical

way to restore and engender trust is to literally make a new processor - one that is developed transparently and may be independently audited to the bedrock. The project develops a low-power, mobile-class, 64-bit Quad-Core OpenPower SoC at a minimum 800mhz clock rate, suitable for tablet, netbook, and industrial embedded systems. Full source code files are available for the operating system and bootloader, and the actual processor, its peripherals and its 3D GPU and VPU. Details at https://libre-soc.org/3d_gpu/

n/a — **Visit https://NLnet.nl/project/Libre-RISCV**

NGI0 PET **Hardware** **LibreSoC** **OpenHardware** **Risk-V**

Sylk Client

One of the things people enjoy the most about the internet, is that it enables them to talk to others remotely almost without limit. Internet allows anyone to keep closely connected with friends and family, and help their kids solve a math problem while they are at work. People collaborate with their colleagues from the couch of their living room, the cafe where they enjoy lunch or on their cell phone on the bus to the gym. Businesses can easily service their customers where this is most convenient to them, without having to travel themselves. This is so convenient, that some businesses have already moved entirely online. Internet communication has become the nerve center of whole neighbourhoods, where people watch over the possessions of their neighbours while these are away for work or leisure.

However, users have a hard time to understand how privacy is impacted if they use the wrong technology. Because internet works almost everywhere, the natural privacy protection of the walls of a house, a school or an office is gone. Unlike the traditional phone companies, many of the large technology providers run their business not on delivering an honest service but on secretly eavesdropping on their users and selling information to others. It is mostly not about what you say, so it is relatively easy for providers to allow some form of privacy by encrypting messages. The more interesting parts are who talks to whom, when, and where they are in the real world while they meet on the internet. if you want to be reachable across the internet, you have to constantly let the communication provider follow you wherever you go. This makes the private and professional lives of citizens an open book to companies that with the help of AI and other technologies make billions from selling 'hidden data' normal people are completely unaware of even exists. And of course in societies that are not so democratic, this type of information is critical to bring down opposition and stifle human rights.

Users assume the confidentiality and privacy when they communicate, and they are morally justified to do so. There is nothing natural or final about internet communication providers having access to all this

very personal information - or going down the dark path of selling data about customers. The cost of this in terms of internet usage and computer power needed is actually negligible, and so all it takes it the availability of open alternatives that people can use. Sylk is clearly one part of the puzzle: it is a mature open source videoconferencing tool that anyone can install anywhere for free. Businesses like the internet provider or the IT company around the corner can run it for their customers, and individuals can run it themselves from their home. Among other things, the project will add the last missing critical component, encrypted group chat, to Sylk. It will not force a new standard, but instead uses internet standards to do so. This means it contributes to a richer ecosystem, where people do not have to use a single piece of software to communicate with others - and anyone can innovate. And by switching, people can regain their privacy and make communicating via the internet as secure and confidential as we all need it to be.

Technical description

Internet communications privacy is important to users, and there is a limited set of encrypted multiparty audio and videoconferencing solutions available to consumers and businesses today. The market, predominantly occupied by proprietary services that often require risky plugins, lack introspection and transparency, proved to expose users to significant security and privacy issues. This trend must be counteracted by better open source equivalents.

SylkSuite, composed by SylkServer and SylkClient is a clean and elegant open source multiparty conferencing solution for both the client and a server written in Python. SylkSuite allows groups of users to communicate privately with rich multimedia, accessed through different protocol stacks. SylkSuite allows bridging SIP clients, XMPP endpoints and WebRTC applications by using Janus backend.

The developers have a focus on strong interoperability based on the use of open standards.

AG Projects — **Visit https://NLnet.nl/project/SylkClient**

NGI0 PET **Encryption** **Videocalling** **WebRTC**

Conversations

One of the things people enjoy the most about the internet, is that it enables them to talk to others remotely almost without limit. Internet allows anyone connected to keep closely connected with friends and family, and help their kids solve a math problem while they are at work. People collaborate with their colleagues from the couch of their living room, the cafe where they enjoy lunch or on their cell phone on the bus to the gym. Businesses can easily service their customers where this is most convenient to them, without having to travel themselves. This is so convenient,

that some businesses have already moved entirely online. Internet communication has become the nerve center of whole neighbourhoods, where people watch over the possessions of their neighbours while these are away for work or leisure.

However, users have a hard time to understand how privacy is impacted if they use the wrong technology. Because internet works almost everywhere, the natural privacy protection of the walls of a house, a school or an office is gone. Unlike the traditional phone companies, many of the large technology providers run their business not on delivering an honest service but on secretly eavesdropping on their users and selling information to others. It is mostly not about what you say, so it is relatively easy for providers to allow some form of privacy by encrypting messages. The more interesting parts are who talks to whom, when, and where they are in the real world while they meet on the internet. if you want to be reachable across the internet, you have to constantly let the communication provider follow you wherever you go. This makes the private and professional lives of citizens an open book to companies that with the help of AI and other technologies make billions from selling 'hidden data' normal people are completely unaware of even exists. And of course in societies that are not so democratic, this type of information is critical to bring down opposition and stifle human rights.

Users assume the confidentiality and privacy when they communicate, and they are morally justified to do so. There is nothing natural or final about internet communication providers having access to all this very personal information - or going down the dark path of selling data about customers. The cost of this in terms of internet usage and computer power needed is actually negligible, and so all it takes it the availability of open alternatives that people can use. Conversations is part of the puzzle: it is a mature open source messaging client that anyone can install anywhere for free. Businesses like the internet provider or the IT company around the corner can run the associated services for their customers, and individuals can run it themselves from their home. Among other things, the project will add the last missing critical component, video calling, to Conversations. It will not force a new standard, but instead uses internet standards to do so. This means it contributes to a richer ecosystem, where people do not have to use a single piece of software to communicate with others - and anyone can innovate. And by switching, people can regain their privacy and make communicating via the internet as secure and confidential as we all need it to be.

Technical description

Conversations is an Android client for the federated, provider independent network of instant messaging servers that use the Extensible messaging and Presence Protocol (XMPP). It aims to provide a feature set and a user experience that is on par with other well known messaging services. While Conversations is capable of sending end-to-end encrypted text messages, images, short videos and voice messages it currently lacks the ability to make voice and video calls. This project is about adding A/V call capabilities to Conversations in a manner that is compatible to other XMPP clients. To achieve compatibility Conversations will implement the Jingle protocol extensions including XEP 0353 (Jingle Message Initiation) for a smooth user experience across multiple devices.

Visit https://NLnet.nl/project/Conversations

NGI0 PET Encryption Federation Mobile Videocalling XMPP

Email was designed without privacy or security in mind, which is amazing for such a popular service. When you send an email, anyone that can gain access to your mail server or the mail server of the recipient can read your mail, from top to bottom. And copy it, for later usage. It is often compared to sending a post card, and of course in many cases there may be little harm in others reading what the weather is like in Paris. But what if you want to use email to send something confidential, something you do not want to share with others? Like a love letter, a political rant or an important contract? And what if you can't actually trust the mail man, for instance because the other party is using a free email service known to search through everything? Or what if you don' t like the fact that your writings are stored in a country you have never been, with different laws that may not be compatible with your thoughts about the world? Or what if you live in a country that has an unhealthy interest in bringing down certain political voices, or are part of a cultural minority that is at risk?

Computer specialists have been protecting their email with encryption for decades. This is the equivalent of putting your message very carefully in the blender, pressing the button before anyone else has read your mesage, shredding it up and sending a packet of shreds over to the other end. The amazing thing about cryptography is that you can magically (or rather mathematically) make it possible for your secret love - and not anyone else - to recreate the message from the shreds, and know it was you - and not anyone else - that sent it. For the rest of the world, the message would be meaningless garble pretty much forever.

However, the solution they came up with is not easy for normal people to work with. You need a lot of patience and technical skill to make use of it. Many people have tried, and could not get it to work or gave up because it hindered them. It was in fact too hard to turn it on by default. This means that most people are probably not even aware that it is possible to protect the contents of their email with cryptography. And so, unfortunately, normal citizens and business have been left behind - exposed to people reading their email messages, and (in the absence of other security measures) potentially also receiving fake or manipulated messages.

Autocrypt is a major contribution to make it far more convenient for people to use cryptography with email. It provides a specification for software to do most of the hard work (hence the portmanteau Autocrypt, which comes from Automatically Encrypt), and thus help also normal users protect the privacy and security of their mail. In the project funded by NGI Zero, they will create a plugin for one of the most popular desktop email clients, Thunderbird. The plugin spots whether the other side is 'autocrypt' aware, and if so will start working straightaway to shred up your mail. Don' t worry: the message comes back at the other end, and you will be safe knowing that noone else can read your mail. Obviously, after this, more email clients will need to gain these superpowers!

Technical description

Autocrypt is a specification that provides guidance for e-mail clients on how to achieve a seamless user experience. It does so by transparently exchanging keys, almost entirely automating public key management. This reduces the UI to "single click for encryption". The project will create an extension for the Thunderbird e-mail client that brings this experience to its users. The goal is to provide a new extension with a streamlined user experience that requires as little user interaction as possible, without "poweruser" features and performing practical user testing to identify open pain points. The extension will be based on OpenPGP.js, since this can be packaged directly. This will simplify installation and maintenance a great deal.

Confidential Technologies GmbH — **Visit https://NLnet.nl/project/Autocrypt**

NGI0 PET **E-mail** **Encryption** **PGP** **UI**

B r i a r

When you get up in the morning, and read a fine piece of investigative news about a financial scandal, you don' t really stop to think much about how news is produced and what the human cost of its production is. Every year, dozens of journalists around the world get killed, because of what they write and who they talk to. Even in democratic countries, people can run the risk of intimidation and retribution. If you happen to be a courageous journalist writing about corruption, gangs or some other social wrong, protecting your sources is more than a matter of principle - it can be a matter of life and death for all parties concerned.

Journalists and other vulnerable groups like civil society groups as well as minorities are starting to understand they need to forsake some of the comforts of modern connectivity, in order to avoid danger to their lives and the live of others. If they use commodity internet communication tools, they will likely put themselves at significant risk. This danger lies not just in leaking the content of what they write and what other people send to them, but more so in the ability to observe who interacts with whom, when, and where they are in the real world while they meet on the internet. if you want to be reachable across the internet, you have to constantly let the communication provider follow you wherever you go. With the help of AI and other technologies much can be derived from 'hidden data' you may not have been aware of until now. Next time you use the wifi in the public library while waiting for your informer, who knows who will be sitting behind you?

Briar is a dedicated communication tool that allows for people to relay messages through people they

can mutually trust, when they cannot trust using the public internet. The best way of staying anonymous is not having to reveal your identity, as this allows for a level of indirectness that can save lives. Briar is an open source messaging application that someone in need of such a tool can start using without other technological dependencies. It creates an 'off-grid' sneakernet which can be used to convey sensitive messages, without leaving unnecessary traces or endangering the people carrying the message - they have no access to the messages of others. Among other things, the project will make it possible to not just send messages, but also in subsequent communications instruct to remove sensitive information that is no longer relevant or is too dangerous to hold onto. And of course, Briar can be used by normal people as well - if you are in a private or business situation that calls for extreme security measures, or the internet is temporarily not available you will be happy to have this app on your mobile phone.

Technical description

Briar is a secure messaging app designed for activists, journalists and civil society groups. Instead of using a central server, encrypted messages are synchronized directly between the users' devices, protecting users and their relationships from surveillance. This project will enable users of Briar to delete their private messages. Giving users control of what information their devices retain will allow them to practice defence in depth, managing their exposure if their devices are lost or compromised.

Visit https://NLnet.nl/project/Briar

NGI0 PET **Messaging** **Mobile** **Offgrid** **Online**

---

n o d e - T o r

On the internet, every computer by design gets a unique number - a so called internet protocol address (or for short IP address). This address is used to send information from your computer to the other computer you want to communicate with, and of course back. Unlike a traditional radio, you often need to send messages to receive messages on the internet. Computers are a great engineering achievement but they are certainly not magic, and thus they need to be able to somehow find each other. The IP address makes this possible. Unfortunately, the fact that every computer has a unique number opens up the possibility of abuse by dishonest actors. Because even though it is none of their business, breaking privacy is a profitable business. If they link what you do on the left side of the internet to what you do on the right side of the internet, they can create a profile and sell this to the highest bidder - with any bad luck to people that want to use it for nefarious purposes.

While work is under way to replace the design of the internet within the Next Generation Internet initiative, there are multiple ways to avoid your IP address being tracked on the current internet. A popular method to attempt to anonymise ones internet presence is to use the Tor network. Tor is a network of millions of computers and users that send messages among each other to confuse someone watching internet traffic. To use Tor, normally you have to install a specific bit of software on your

computer that runs a service in the back end. Installing and maintaining that software does require some technical skill and the rights to install software on the computer you are using, which for instance at work or in schools may not apply. So not all users can equally benefit from this.

Of course, having such software on your computer could be an argument in some countries to prosecute you. But installing it over and over again, is also not an option and requires even more technical skill - not to mention putting users at risk of installing some malware if their skills are limited. Using a so called live medium (like a thumbdrive or a CD) requires you to close every application you are running, and spend minutes rebooting the computer, every time you want to anonymously interact with some website. What if the burden of providing Tor access could be moved to someone offering a service on the world wide web? That way you could protect any interaction of all users with say a specific website, without the users needing to install anything. Node-Tor is a re-implementation of a part of the Tor protocol. It operates all but invisible to the user, and connects the user to the Tor network directly from the web browser - no configuration needed. This allows entirely new use cases for anonymisation.

Technical description

Node-Tor is an open source project and the only existing implementation of the Tor protocol in Javascript. That gives it the unique property to not just run on a server or desktop, but also inside a regular webbrowser itself as a standalone secure webapp. It must not be misunderstood for just a re-implementation of Tor network nodes: the goal is much wider, because it allows any project related to privacy/security enhancement to implement the Tor protocol in their nodes and/or inside a web page. The browser client acts as a standalone node itself communicating via web interfaces such as Websockets with servers or through WebRTC with other browsers. The use of Javascript allows to reduce very significantly the code and libraries (prone to security breaches), simplifying the integration for developers (like removing the need to maintain installation packages since standard web interfaces can be used), simplifying the use for users. This offers a lot of potential for increasing security and privacy for everybody, since the technology can be accessed from any place and any device that has a browser or can run Javascript, including mobile devices.

Nais - Informatique Telecom — **Visit https://NLnet.nl/project/node-Tor**

NGI0 PET **Clientside** **OnionRouting** **P2P**

IMSI Pseudonymization

The mobile phone has become the very center of our digital life at a spectacular and very impressive rate. People literally go to sleep and wake up next to their mobile phone, and some clutch their phone in their hands for large parts of the day. On average they touch their phone more than they have physical contact with their combined friends and family, and even more so than any other object they own - apart from the clothes they wear. The amount of households in Europe without a mobile subscription has shrunk to a mere one in twenty, and globally over 5 billion people now use cell phones - and the number continues to grow.

When we turn on our cell phone, it safely connects to the network within seconds - ready for us to catch up on news while riding public transport, order food, play games, find our way around a city and keep connected with our social environment. When we move around, the phone invisibly detects this and seamlessly hands over the current session to the closest tower. This is done so smoothly, that most people are unaware of such a handover when it happens. But is the way a phone connects really as secure and trustworthy as we assume it is? As it turns out, mobile phone standards fail to protect the privacy of users due to a flaw in the core design which opens up a number of options of abuse. The GSM specification makes it mandatory for your phone to reveal its identity to the network. Mobile phones use radio waves to establish a connection with the closest base station, meaning that this information is received and broadcast over the air. Since the network is presumed to be operated by a regulated and licensed network provider, there is no requirement for the network to prove to the phone that it is legit. Phones have been programmed to trust the network, even if this is not justified. This design flaw is actively abused by so called IMSI catchers, which are devices used to eavesdrop on mobile phone users.

The idea behind an IMSI catcher is as simple as it is hard to notice: a fake device somewhere sends out signals impersonating a cell phone tower from one of the existing providers. Mobile phones by design want to optimize the signal they receive, and choose the strongest signal among the different reachable candidates. If the attacker has a strong enough signal, all phones in its neighbourhood connect to the fake device instead of to the real network. At that point, all the attacker needs to do is to send a standard signal for all the phones to expose their unique mobile subscriber identity. An attack only takes minutes, is very difficult to spot and can be used to expose the identity of anyone close by that has their phone turned on. In fact, such an attack also works on devices like sensors in bridges, roads and inside companies that use the mobile network to communicate. Subsequent forms of abuse may follow.

In this project the well-known security researcher and software developer Harald Welte, founder of the Osmocom project, is redesigning the mobile protocols in such a way that the phone of the user would no longer have to reveal any information before it establishes an encrypted session. This would immediately prevent a number of currently unstoppable privacy attacks targeting IMSI exposure. The project will submit these proposals into the international 3GPP standardization process, to make it available to all networks - so that when in the future you turn on the phone, it can keep its secrets safe until trust is established. As such, this project promised to be an important contribution to protecting the privacy of consumers, and the confidentiality of business and continuity within the public sector.

Technical description

The IMSI Pseudonymization project will design a specification and provide a reference implementation of a mechanism to conceal the IMSI (international mobile subscriber identity) of a mobile subscriber on the radio interface. The IMSI is used to uniquely identify each subscriber in a (2G, 3G, 4G, 5G) cellular network. However, the privacy of users is not really well protected: current specification require to transfer the IMSI in plain-text at various times before an encrypted connection can be set up. The present project will specify, implement and evaluate a method by which the IMSI will be concealed on the air interface with no modifications to existing mobile phones or any network elements of the operator beyond the HLR/HSS (which implements the authentication on the network side). The project will further submit this proposal into the 3GPP standardization process and attempt to make it at least an optional extension that operators (even MVNOs) can deploy.

Osmocom project — **Visit https://NLnet.nl/project/IMSI-pseudonymisation**

NGI0 PET MobileInfrastructure Observability StandardSetting

Verifpal

Secure communication over the internet is critical. Humans however are not infallible, and the same holds for the humans that design the protocols that should make our internet traffic safe. Internet engineers and software developers need to handle a lot of complexity, and even a small oversight or a very improbable scenario or combination of factors can mean breaking part or whole of the protection required The secure technologies we depend on to keep internet communications secure are frequently found to suffer from fundamental design vulnerabilities as well as implementation errors. Truth is, while trust is a fundamental human trait, we should not just trust human intuition to get everything right.

This is where computers can come to help us out, to see if we can underpin that trust in a systematic way. Computers have no problem to exhaustively try out all options, even if it takes them millions and millions of tries. When instructed in the right way, that means their endless combinatorial capabilities can be used to simulate even the most unlikely of events. Again and again, if necessary. A lot of awesome computer science brain power has gone into so called formal proofs. Formal proofs use very strict mathematical modelling to take everything that could possibly happen into account, and prove that the software or protocol at hand does what it is assumed to do. However, as you may imagine, this modelling can get pretty complex and as such is an art in itself - restricting the usage to a very limited set of experts. However, once you have the models right you can actually go a lot further than just prove the protocol: from the model you can automatically generate secure software libraries that you can be sure implement the protocols involved exactly right. This is a guarantee that no human programmer can give.

Noise Explorer is the first of a new generation of open source tool that is helping to democratise these proofs, and bring together community knowledge about protocols and proofs at the same time. It conveniently assists those designing secure channels based on the so called Noise Protocol Framework, which is used in some of the largest messaging tools in the market to protect the confidentiality of the messages sent around. The creators of Noise Explorer have precomputed many different options, and so developers can just take the proofs instead of having to model their protocol and spend a lot of time on setup and computation.

Verifpal is the logical next step. It expands the scope of Noise Explorer to make it applicable for many more protocols that need to be secure. Whether you successfully connect to a wireless network, or see the little green padlock next to a website address - we all want to trust the security behind that. VerifPal is creating a unique tool specifically designed to make it easier to make protocols that will not let us users down.

Technical description

Noise Explorer is an online engine for reasoning about Noise Protocol Framework (revision 34) Handshake Patterns. Noise Explorer allows you to design Noise Handshake Patterns, and immediately obtain validity checks that verify if your design conforms to the specification. For visually oriented people, it provides a convenient visualisation in your browser. Noise Explorer can also generate Formal

Verification Models and Software Implementations. This allows to instantly generate full symbolic models in the applied pi calculus for any Noise Handshake Pattern that you enter. Using ProVerif, these models can be analyzed against passive and active attackers with malicious principals. The model's top-level process and sophisticated queries are specifically generated to be relevant to your Noise Handshake Pattern, including tests for strong vs. weak forward secrecy and resistance to key compromise impersonation Noise Explorer also automatically generates a secure implementation of your chosen Noise Handshake Pattern design, written in Go. In addition the users can explore a Compendium of Formal Verification Results. Since formal verification for complex Noise Handshake Patterns can take time and require fast CPU hardware, Noise Explorer comes with a compendium detailing the full results of all Noise Handshake Patterns described in the original specification. These results are presented with a security model that is even more comprehensive than the original specification, since it includes the participation of a malicious principal.

Symbolic Software — **Visit https://NLnet.nl/project/VerifPal**

NGI0 PET **CodeGeneration** **Cryptography** **FormalVerification**

A proof of concept of identity-based encryption

Email was designed without privacy or security in mind, which is amazing for such a popular service. When you send an email, anyone that can gain access to your mail server or the mail server of the recipient can read your mail, from top to bottom. And copy it, for later usage. Or modify it. It is often compared to sending a post card, and of course in many cases there may be little harm in others reading what the weather is like in Athens. But what if you want to use email to send something confidential, something you do not want to share with others? Like a love letter, a political rant or an important contract? And what if you can't actually trust the mail man, for instance because the other party is using a free email service known to search through everything? Or what if you don' t like the fact that your writings are stored in a country you have never been, with different laws that may not be compatible with your thoughts about the world? Or what if you live in a country that has an unhealthy interest in bringing down certain political voices, or are part of a cultural minority that is at risk?

Computer specialists have been protecting their email with encryption for decades. This is the equivalent of putting your message very carefully in the blender, pressing the button before anyone else has read your mesage, shredding it up and sending a packet of shreds over to the other end. The amazing thing about cryptography is that you can magically (or rather mathematically) make it possible for your secret love - and not anyone else - to recreate the message from the shreds, and know it was you - and not anyone else - that sent it. For the rest of the world, the message would be meaningless garble pretty much forever.

However, the rather technical solution computer scientists came up with is far from easy for normal people to work with. In particular, both the person sending an email but also the recipient need to carefully prepare themselves upfront in order to even be able to securely exchange messages. As a sender you need to somehow obtain a digital "key" from the intended recipient. Without it, it is not possible to use classical email encryption. To make it even more cumbersome for all parties involved, the exchange cannot in most cases take place using email. Otherwise an peeping tom that would be able to intercept these emails, could easily figure everything out and still intercept messages or prior to that swap keys out - after all, someone having unapproved access to your email was the very problem statement we started with.

Certainly, this rather fundamentally unpractical requirement rules out many different use cases. People tend to send lots of email to people they have never had contact with before. So you cannot just turn on encrypted email and let the software handle the rest: a new key will need to be obtained for every new person you have contact with. This results in rather unpractical arrangements, which is the main reason there is still so little adoption of something as useful as encrypted email.

The Identity-Based Encryption project by professor Jacobs is going to explore an interesting alternative to this traditional way of doing things. Not because the current technology isn't secure, but because the user experience promises be so much better. Learning from recent advances in academic research, the project aims to resolve the need to manually retrieve keys across the internet for every contact - as a eager sender you can just generate that key yourself for every recipient. That means you can start sending encrypted email to anyone with a mail address straightaway, without all the impracticalities of the outback. All they need to do when your encrypted email arrives in their mailbox, is go and pick up the corresponding secret key somewhere. And this is a one time task for them, after someone has actually sent them an encrypted email which they will want to be able to read. This is perhaps the equivalent of showing your passport or driver license when you go to the post office to pick up a package. A working solution in this space would be a fundamental building block for the next generation internet. The proposed scheme requires less patience and far less technical skill from the users, and less margin for error. That would signficantly help democratising the encryption of email. We need to make safe email available to anyone, and the sooner we are able to do this the better.

Technical description

The project aims to extend the existing attribute-based identity platform IRMA with easy-to-use encryption. The kind of encryption is called Identity-Based. Its main advantage is that key management is simple, so that encryption becomes easy to use, via a plugin to an email client (only Thunderbird in this proof of concept project). The plugin computes the public key of the recipient of a message, from some uniquely identifying attribute of the recipient (typically an email address, but phone number, or citizen registration number could work as well). The receiver of the message will have to prove, via IRMA, possession of the uniquely identifying attribute to some Trusted Third Party (TTP), which will then provide the corresponding private key. Within this project a working set-up will be built. Turning it into a widely usable product will require more work, in follow-up projects.

Privacy by Design — **Visit https://NLnet.nl/project/IBE**

NGI0 PET **Attributes** **Foundation** **IdentityBasedEncryption**

Often, on the internet, we give away a lot more information than necessary. Imagine a situation where a student wants to claim a discount offered to students by an online book store. That means she will somehow need to prove to in fact be entitled to the discount. When you walk into a normal book store, you are able to buy a book without telling any of the staff who you are. And yet, online, it is somehow perceived as reasonable for the student in our example to have to upload a picture of their student ID in order to qualify for the very same discount. This ID contains lots of unnecessary GDPR protected information in addition to the student affiliation. Think about your name and especially it also contains sensitive biometric information. The online book store does not have a legitimate interest in the color of the students skin, or even in their name - often, that means they are just a search engine away from knowing a lot more about you than you care to think about. All they care about is not having to give unnecessary discounts to people other than students.

Technically, it is of course entirely feasible to minimise what is shared. What if you were to find some credible organisation that would be willing and able to vouch for your claims? If you trust me, and I say to the book store owner I know for a fact you have a valid student card. Would you still need to see the card yourself? Of course there are many claims (in technical terms called "attributes") people may need to fulfil: being a student or entitled to a discount for seniors, being legally adult to see some movie or a verified minor to be allowed in a kids chatroom, being a journalist to attend a press webstream or being unemployed to qualify for benefits. With the open source IRMA project (IRMA stands for "I reveal my attributes") you as a user are in full control who gets to see what very specific attributes, and you don't have to worry about the rest. Many applications may claim such broad capabilities, but this unique open source application can actually deliver. It has a solid academic base, with over a decade of research of top cryptography experts backing up the technology.

The technical basis may be extremely solid, but there is a sociotechnical dimension to this as well. The open source project is now at the point where it is getting deployed at some scale out there in the wild. And of course, as with any new technology, not all users fit within the original design. Users do not always behave predictably. And the diversity among users is huge - and so is their technical skill set. Elderly, children, people with disabilities - they all bring in new usability requirements. Without good usability and inclusive design, even the best technology will fail in the market. The next phase of the IRMA open source project is all about iterating on the technology to make it useful to a broad audience. The project entails implementing a number of experimental new designs, iterate over different tradeoffs and carefully studying how actual users interact with them and which options achieve the best result. The most reliable protection of data is to have as little of it out there. IRMA has the potential to be a game changer for privacy.

Technical description

Authentication methods, like passwords, often involve a trade-off between usability and security. Secure

passwords are a hassle to use, and easy-to-use passwords are often also easy to guess or to brute force. Clearly, there is a need for authentication methods that are both secure and user-friendly. The IRMA mobile app can fill this gap. It was originally developed with a strong focus on providing secure and privacy-friendly authentication. This project will focus on making IRMA easy to use for everyone. We will conduct a formal large-scale evaluation of IRMA that focuses on usability in general as well as on accessibility (i.e. for users with disabilities) in particular. By doing so, usability hindrances can be identified and improved, making IRMA user-friendly and accessible for users with the widest range of capabilities.

Radboud University — **Visit https://NLnet.nl/project/IRMA-made-easy**

NGI0 PET **Attributes** **InclusiveDesign** **UX**

CryptPad

Collaboratively writing a document together in real-time with others is still a bit magic. Someone else, perhaps on the other side of the planet, is typing something. And within a fraction of a second, the text magically appears on your screen. If you insert some text in the text just typed, this travels to all people you are in the session with. This amazing technology is the ideal companion for say an online meeting - everyone can contribute, and correct any flawed minutes without much effort.

For this kind of collaboration in real-time, there is a limited set of options in the market you can use. Most available services in the market like Google Docs, Microsoft Office or LibreOffice Online share one very undesirable characteristic: you need to fully trust the company running the service you use. Whomever has access to the servers used to connect everyone together, can read everything you have written - and deleted. That means that if you need to work on something confidential like an important contract, you may want to reconsider using the service. If you by accident cut and paste a password in the wrong window, you probably need to change it.

Especially if you write about sensitive topics like corruption, money laundering or state surveillance this open backend you cannot control is a really significant problem. If the server is located in another jurisdiction, you probably want to watch carefully what you write - you may inadvertently violate some laws you are literally unaware of.

Cryptpad is different: it is free and open source software you can run anywhere you want yourself. This means you can choose someone you really trust, rather than being forced to trust. But even better, CryptPad will make everything you do undecipherable to the outside world before anything is sent to the service to be distributed among all the participants. The infrastructure is consciously left ignorant of anything having to do with the content you write: it just diligently routes all the different contributions from you and your fellow collaborators across the internet. It cannot look inside the traffic. All of this is

done without bothering the user. From a user perspective it works as any other application. That means CryptPad puts you square back in control. In the project, the researchers will make it more easy to work with CryptPad as a team rather than as an ad hoc group - adding advanced group management capabilities to the system. This will allow you to add and remove collaborators, and thus to improve operational security when you use the software intensively.

Technical description

Cryptpad is a secure and encrypted open source collaboration platform. The CryptPad teams project will fund the development of a number of group-focused features to Cryptpad. We'll improve our current implementation of encrypted shared folders to display the permissions possessed by team members for different documents. The capacity to remove a member from a group is difficult in an encrypted system, as the knowledge of encryption keys cannot be taken away once given. We'll implement key-rotation protocols, and develop encrypted mailboxes to facilitate the delivery of new keys to authorized members. The same mailbox system will enable the development of notifications, allowing users to request additional permissions for documents, to invite new members to a group or session, or to inform friends that a document has been updated. Teams organize in many ways, and with the technical components available we'll focus on interfaces which support different modes of coordination, whether the team is hierarchical or self-organizing. Overall, we hope to make it so that the most intuitive way to collaborate is also the most secure.

XWiki SAS — **Visit https://NLnet.nl/project/Cryptpad**

NGI0 PET AccessControl ClientSideEncryption Collaboration TextEditor

---

Padding Machines for Tor

On the internet, every computer by design gets a unique number - a so called internet protocol address (or for short IP address). This address is used to send information from your computer to the other computer you want to communicate with, and of course back. Unlike a traditional radio, you often need to send messages to receive messages on the internet. Computers are a great engineering achievement but they are certainly not magic, and thus they need to be able to somehow find each other. The IP address makes this possible. Unfortunately, the fact that every computer has a unique number opens up the possibility of abuse by dishonest actors. Because even though it is none of their business, breaking privacy is a profitable business. If they link what you do on the left side of the internet to what you do on the right side of the internet, they can create a profile and sell this to the highest bidder - with any bad luck to people that want to use it for nefarious purposes.

While work is under way to replace the design of the internet within the Next Generation Internet initiative, there are multiple ways to avoid your IP address being tracked on the current internet. A

popular method to attempt to anonymise ones internet presence is to use the Tor network. Tor is a network of millions of computers and users that send messages among each other to confuse someone watching internet traffic. Of course, this is an arms race between those that want to be anonymous when they visit some webpages and those that want to achieve the opposite goal.

Researchers found out that while the actual content can be well obscured with lots of intricate math operations, no activity is still observably different from some activity. That means sometimes the patterns of usage would still put users at risk. This is the background of this project. It will attempt to create fake network activity that is realistic and plausible, in such a way that an attacker will not be able to infer much anymore about Tor users. Tor is used a lot by ordinary people but also by journalists, whisteblowers, dissidents, diplomats and others for who the loss of their anonymity while using the internet can have very dramatic consequences. The project therefore contributes to both privacy and security of internet users.

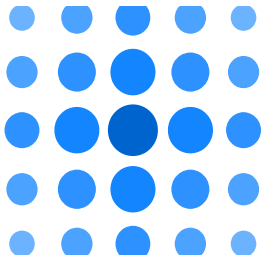
Technical description

Tor is the worlds largest anonymity network with about eight million daily users around the world who use Tor to browse the web anonymously, access onion services, and circumvent censorship. The project Padding Machines for Tor will design and implement padding machines---as part of a new framework in Tor for generating fake padding traffic---to defend against website fingerprinting attacks. A website fingerprinting attack is a type of traffic analysis attack where an attacker attempts to determine websites visited by a target Tor user by analysing encrypted traffic. The results of the project will be both open source and open access, with the goal of contributing to effective and efficient defenses deployed by default in Tor against website fingerprinting attacks.

Karlstad University — **Visit https://NLnet.nl/project/TorPaddingMachines**

NGI0 PET **Deanonymisation** **Fingerprinting** **OnionRouting**

Universal DID Resolver and Registrar



One of the oldest questions on the internet is: how do you adequately prove you are you? Or perhaps the reverse formulation offers a better mental model: how do you prevent others from succeeding in pretending they are you? Now lets flip this question around once more: how would you like to see this managed yourself, if you could? How heavy-weight or convenient do you want to be proven that you are you, to allow you to get into your own environment or have something done on your behalf? And what is it worth to you in terms of effort? Would you be willing to spend a minute to have some clever secure device you have in your pocket involved? Authenticate via your mobile phone? And what if you are in a rush, or on the go? Are you happy with some company like

your email provider or a large social network having the ability to make that judgement, based on a user login a few hours ago? And what if that company is based in some other jurisdiction, and could be forced to let others in as well? Or would you rather choose your own identity, and formulate direct rules to have complete control at any given point?

As could be guessed, individual people have a need for different levels of confidence and security in different contexts. A security breach matters perhaps less if you just want to login to a music service to change a playlist. After all, the worst that can happen is that someone messes things up and you have to create a new one. It matters a great deal more if you want to do a significant financial transaction at work, or open the door of your house remotely to let the babysitter in while you are delayed in traffic. Perhaps you can think of scenarios where you want even more control.

So what proof to use as the basis of your trust, and the subsequent actions taken? Historically people rely on some authority they collectively trust. Such an authority has typically taken high tech countermeasures to make the channel through which that trust is conveyed hard to fraud. A passport or banknote are quite tricky to fabricate due to the use of special techniques. Online we have only a very limited amount of trust "anchors" of varying quality. The domain name system is such an anchor, digital certificates or customer relationships are another. Today, having access to a certain mail account or phone which is known to be yours is the most common proof used. Email is often called the "poor man's solution" to identity management, and it is what most organisations and businesses fall back on. Can't log in? We will send you an email to reset your login. Just click on the link. And of course, email was never designed to be safe. It kind of works, but really we can do better.

Perhaps your use cases require more strict proof than that of normal consumers, or less strict proof. Even for a single large service provider, it would hard to figure this out satisfactorily for all users. For the same reason people write their own testament to document what should happen with things they own or control after they die, you want to document what should happen with things you own or control what happens when you are physically absent. There is no universal will that is acceptable to all, nor is there a universal policy that satisfies all use cases.

So what if you yourself would be able to create and control your own identity, and determine your own proofs and methods? In order to function in a global internet, you would need to be able to convey your requirements and demands in a portable way. There would be no central authority dictating you what to do here. That would mean you you yourself would have to make things explicit upfront in a foolproof way - so that elsewhere on the internet people and services would know what you expect them to do to distinguish the real you from fraudsters.

This is the starting point of the DID Resolver and Registrar project. These two applications are being designed to help you create a machine processable travel document for identity management tied to your data and the services you use or provide. They are part of an initiative within the web standards community to create suitable standards for handling decentralised identities. The software will help to tell others exactly how you want to see things handled. The outcome is expected to contribute both to the standard, and to an actually working solution where users can design and manage their own decentralised identities. The project is led by one of the authors of the W3C specification.

Technical description

The Universal DID Resolver and Registrar are open-source software components that implement Decentralized Identifiers (DIDs). DIDs lie at the heart of an emerging technical and social paradigm known as "self-sovereign identity" (SSI), which allows individuals, organizations, and things to create and

manage their digital identities without dependence on any central authority or intermediary. This technology is highly aligned with Next Generation Internet values such as human-centricity, openness, trust, and reliability. DIDs as a building block for protocols are of similar importance to Internet infrastructure as other identifiers such as domain names or e-mail addresses. The Universal DID Resolver and Registrar are aligned with corresponding W3C community group specification efforts. Development and maintainance of the code takes place in close collaboration with relevant community and industry stakeholders such as the Decentralized Identity Foundation, uPort, Jolocom, Sovrin, Civic, Veres One, Blockstack, ERC725 Alliance, etc.

Visit https://NLnet.nl/project/UniversalDID

NGI0 PET **DID** **SelfSovereignIdentity** **W3C**

Opaque Sphinx

For many people, one of the things that is most difficult when using the internet is remembering all the passwords. As a result, they will act unwise. Some use very simple passwords that extremely are easy to guess. Or reuse the same password all the time, so that a single leak of one of the services they use on the internet will expose all the others. This makes it very easy for those that want to abuse the accounts of these people to do so. Hackers have started collecting all passwords that have ever been publicly exposed in a hack. There is an alarming success rate if you start trying these out one at a time - with a simple bot net you can hammer the servers at a rate of many thousands of attempts per second. But instead of blaming the victims for reckless or uninformed behaviour, we can see if we can solve the challenge in another way.

Make a 180 degree turn, and take the position of a service that needs to authenticate a user. You get an account name and a password, and compare this to what you have on record. If it is correct, you let the user enter. If not, you reject. But what if you could make it so that you as a service provider never get to see the actual passwords of any of your users, but instead can register and negotiate in a more clever way than with a bare password? With the right technical design (which in technical terms is called a "Password Authenticated Key Exchange"), you don't need to bother the user with all kinds of additional requirements for this. You can just handle everything in the software at both ends. This novel solution is based on advanced modern cryptographic research, and really deserves to be deployed everywhere, but isn't - yet.

If you as a service provider can verify your users have access to the right password without you actually getting to ever see it, you do not run the risk of ever exposing their data. Even if the user uses the same password everywhere, neither you or anyone else can ever find out. Implementations already exist for desktop operating systems like Linux and Windows, but the Opaque Sphinx project will make it possible to effectively use this technology on Android phones too.

Technical description

Opaque Sphinx is a project that aims to secure password-based authentication by deploying the state-of-the-art SPHINX and OPAQUE cryptographic protocols to eliminate almost all common attack vectors - such as weak guessable passwords, password reuse, phishing, password databases, offline dictionary attacks, database leaks - plaguing current solutions. These protocols provide the strongest available cryptographic properties with cryptographic proofs. The project intend to port its already existing free software SPHINX implementation - besides already existing support for Linux and Windows - to Android so it can also be used on smartphones.

Visit https://NLnet.nl/project/OpaqueSphinx

NGI0 PET **Encryption** **Mobile** **PasswordStore**

ValOS Cryptographic Content Security project

Internet technology is perceived as rather complex, more than is probably necessary. That is why people tend to let other people be in control of the technology they use, even though noone knows what they need better than they do themselves. While there are millions of professional and amateur developers capable of creating applications on the web, that leaves billions that cannot. There is no technology currently well suited to bring development to the average person in the street. ValOS (Valaa Open System) aims to simplifies software development and makes web apps inherently more secure by default with no or little effort from developer,.

It does so by creating a much simpler model to work with. The content that the users create, and the application that enables them to do so, are brought together from different sources inside the browser - not before. The architecture of the system assumes the worst possible environment, because creating security and resilience when everybody is honest and connections are perfect is just unrealistic. No one should need to trust anyone else by default. Clients can and will crash at any moment, but the user expects her data back. Sessions can get lost and devices can lose connectivity, go offline without crashing, keep creating new commands into an outgoing queue and expect to survive coming back online.

ValOS aims to enable a new paradigm ecosystem where applications are secure by default with no or little effort from developer, further enabling the creativity of everyone. By design data can remain located securely in someones phone, under the direct control of the user. It aims to make the system robust in most imaginable scenarios and to allow it to fail securely in outlier cases. This is a highly experimental but visionary project with a lot of potential.

Technical description

ValOS (Valaa Open System) is a project pushing programming to become a civic skill. It’s a decentralized

software development architecture that empowers beginners with little training or prior experience to create practical web applications. ValOS applications and data are created, stored and distributed as event streams. ValOS Gateway is a JavaScript library that acts like a browser: it connects to event streams, reduces them into applications and provides means to induce new events. ValOS Cryptographic Content Security project focuses on enhancing the infrastructure level security of ValOS through event log hash chaining, end-to-end encryption and other features.

Valaa Technologies Ltd — **Visit https://NLnet.nl/project/ValOS-crypto**

NGI0 PET **IdentityManagement** **Protocol**

Distributed Private Trust

When we read some news on the internet, how do we know that someone we don't know is not a fake. Can we somehow distinguish between 20.000 likes for a fake "Black Lives Matter" or "Gilets jaunes" social media account created for political subversion with a pyramid of fake accounts, and a network of real humans concerned about real issues? If we get a technically valid email message with all proper technical assurances from europaa.eu from a hacker (don't worry, we claimed it), is there a proper way to distinguish it from europa.eu - a domain that has been active for nearly two decades and which has sent out billions of email messages? The answer is of course that we would need a globally scalable reputation system. And that is a really hard problem to solve.

We actually have some reputation systems in the commercial world, but these give us a lot of questions as well. What does it mean if a hotel or a taxi driver gets an average "2 out of 5" stars for their service through some online service? Will they ever do business again? How many people get blackmailed with such a threat? Who actually sets the criteria, and who vets that all the responses and numbers are real? Does a user giving a really positive valuation give it to the virtual taxi company running a website or to the individual driver? What if that driver wants to work somewhere else, or is unhappy about the policies and high fees charged the website? Can a competing website use the data?

Outside of a single context, it is hard to agree on criteria and policies for reputation. One countries political activist or dissident is another countries public enemy. A privacy-invasive commercial business that is perfectly legit in the USA may be illegal in Europe. And yet we want to be able to delegate trust and accept trustworthy information from others. The individual human brain does not scale to real-world trust knowledge about billions of people, companies and resources at internet scale. There is unlikely to ever be a single system, and if there ever would be it would be too powerful. This is why decentralised reputation systems are probably among some of the most anticipated technologies for the NGI.

Decentralised reputation would add scalability to trust delegation, which is required for any social system to scale to internet size. This is not easy, in many way. Very little is more private than the trust conveyed in others, especially when trust has been damaged but there is still a professional or social dependency. There is a taboo on saying a colleague is horrible, and should never be allowed to work with something important ever again. Yet, if that person asks, social pressure will kick in. Incoming negative trust affects outgoing trust.

If trust delegation is too naive, people will try to game the system. If you probe around a little, you will find ways online to buy likes, mentions, SEO links. This projects wants to create a mechanism that isn't so easy to game. This is a small, exploratory project that is confident it can build a first prototype. The project will use something called "secure multi-party computation" to calculate aggregate ratings without having to reveal individual users ratings to any other party.

Technical description

The project "Distributed Private Trust" wants to develop a prototype for a trust and reputation system that does not rely on a centralized trusted party and provides users with more privacy than current systems. It uses secure multi-party computation to calculate aggregate ratings without having to reveal individual users ratings to any other party. The project also applies techniques from mechanism design to make the system robust to malicious behaviour of participants, for example by diminishing incentives to submit dishonest ratings.

Visit https://NLnet.nl/project/DistributedTrust

NGI0 PET **MultiPartyComputation** **Reputation** **Trust**

Finish porting Replicant to a newer Android version

Replicant

Consumers that go shopping for a new cell phone or tablet these days, at the surface have quite a choice. Even the cheapest of mobile phones sold today, is surprisingly powerful compared to that of a couple of years ago. All that seems left for consumers to do is to match their own sense of style and of course budget. If they are really eager, they might compare a limited set of technical specifications: How long does the battery last? How big and bright is the screen? And do games and movies run smoothly? Most users tend to not even bother about that, eager to jump straight to the app stores filled with more applications than a human could feasibly install in their life. What more could a mere user want?

Somewhere in the back of our minds there may be lingering some larger, less happy thoughts. What about security and privacy? Who really is in control of our devices? It is not easy to connect the joyous occasion of our (often much anticipated) purchase of a really cool new gadget with societal resilience, our collective future well-being or any other of the larger economic effects of our individual choices...

In the early GSM era, there wasn't a single dominant operating system from a single vendor. The market was competitive and rather straightforward from todays perspective. Major efforts like Symbian (which ran on the very popular phones of erstwhile market leader Nokia, but also on those of Siemens, Alcatel, Bosch, Sharp, Sony Ericsson etc) were the result of a pragmatic collaboration on more or less equal footing of many manufacturers. These had a shared development responsibility, and equal opportunities. None of them knew how their users actually used the phones they created: that was the

business of the customer.

The subsequent rise of the smartphone resulted in market disarray, because the dynamics of the new situation were so different. It wasn't so much a difference in technical quality that set the new masters of the universe apart, it was a complete change of the underlying business model and value proposition few people properly understood - if any.

The real-world cost of developing and maintaining the first generation of mobile platforms was non-trivial, and price competition in the devices was heavy. And then suddenly a no-visible-cost and feature-rich smartphone operating system appeared on the market. It wasn't produced by any of the current competitors or by an open consortium. The source was a single company that had heavily invested into this for strategic reasons. In parallel Apple was able to launch its own effort, take its slick iPod music player and its strong media presence and market visibility in the desktop space. Their premium iPhone line addressed the most luxurious part of the market - also with the help of Google. The CEO's of both companies even sat on each others boards, so the strategy was certainly aligned.

It was a perfect coup. Among the two of them they effectively levered the possibilities of the mobile smartphone platforms, media stores and restricted-access platform-owned app stores to take ownership and control of large parts of the software and content ecosystems at global scale. Traditional phone manufacturers (many of which were European due to the success of the pioneering GSM standard) had historically been just selling a phone at competitive margins (with "no strings attached"). The whole economy of their operations and ecosystem of collaboration was effectively pushed aside by this audacious new strategy. The new Android operating system was funded not by the sale of the product itself, but by the promise of future user data gathering without real limits or much oversight - which had elsewhere proven to be able to create giant revenues. And unlike a desktop computer, a phone is nearly always on. It moves wherever the user goes, and thus it is always near. It has a camera, a microphone and lots of sensors. When users search for something, they use the default search bar which you control.

So effectively the new "smart" phone was primarily a vehicle for extensive data gathering about users, which could be resold and monetized later on. The manufacturers could get the operating system for free. The small margins that could be made on selling the software to they were negligible compared to the advantages later on. And of course at the time there was still a generation adoration of these "tech darlings" - press wrote lovingly about the "reality distortion field" around Apple's CEO Steve Jobs.

Right from the start this conceiled play was extremely profitable for both of them, allowing lots of subsequent investment - into their platforms, into the developer tools, into marketing and into legislative lobby. The "mobile first" strategy actually worked out better than anyone would have imagined, especially because the mobile phone operating system produced by Google turned out to be more than just a "loss leader". The market funnel of the free option it provided only became visible at the end. Technically advanced and more fair platforms appeared, but were unable to counter the "winner takes all" development in time. At present the vast majority of the phones are sold using one of only two operating systems: Android and iOS. In the absence of effective policy and legislative efforts to curb this unfortunate situation, that market dominance is a hard problem to solve at a technical level.

In our consumer bubble, we actively contributed and still contribute to this. The software stores of both platforms may offer consumers plenty of options at the application level. This seems quite healthy at first. But when you analyse the situation, it is far from how society should want this to be. This all starts with the fact that users do not have to manually install all applications. Apple has full control and puts its own software in pole position. Google is able to make the manufacturers do the same through contractual obligations. The result is the same: a strategic choice of end user applications is preinstalled alongside the platform, and effortlessly available to all users.

Many of us have meanwhile become used to these omnipresent "free" but closed "blockbuster"

applications that ship alongside the dominant platforms. As we know from history, for instance through the famous European anticompetition cases against dominant technology companies taking control over web browsers, media players and portable runtimes (Java/C#), preinstalled applications have a huge competitive advantage. Not all users are as technically competent, and this creates enough inertia with consumers to keep manufacturers on a leash. The huge market share of platform 'defaults' like Android's default browser have a deep impact on the market, leaving little room for web developers to follow pretty much all what Google implements - even if they disagree or would actually like to follow proper web standards as produced by W3C. Who can affort for their website or web application to look worse on an operating system with the majority of market share?

Apple holds all the cards closely to its chest, and keeps full control. As long as it has Google as competitor, it feels secure of anti-competition measures. Their main strategy to even increase control is to buy suppliers, or make them sign exclusive contracts keeping others at bay. The defense strategy of Google is publishing most of Android source code. Manufacturers can and have tried to build alternative versions based on that. But in the market real-world control remains tightly with Google through the critical applications which need the "blockbuster" restrictively licensed apps and the larger infrastructure - both of which remain tightly closed. A certain percentage of users will always at some point demand these "free" applications, while others cannot withstand the social lock-in and will actively push vendors to bow down. No small time manufacturer can afford to be out.

The platforms realise this powerful position very well, and are not afraid to lever it. Either a manufacturer is all-in, or all-out: it cannot selectively allow individual users to use blockbuster applications later on. This cut-throat dillema has left the companies that make the actual phones little choice but to accept unattractive licensing conditions that restrict their freedom to innovate. And even if they do comply with all the demands including a non-disclosure agreement to seal their lips, their license can be withdraw at any time. In fact this may even happen due to geo-political pressure, as a very large Chinese manufacturer of Android found out to its great dismay in May 2019 when it was banned from future upgrades to Android. That can happen to any phone vendor using Android at any time.

Thhe rigid control over the platform and the app stores was originally meant as a way to secure access to consumer data. These days, it is actually making an awful lot of money on its own. Consumers are paying a huge and very direct cost for the 'free platform' deal of the manufacturers. The dominant mobile platforms both charge developers up to an incredible 30% of their revenues (more than any VAT rate around the world!).

If your company wants to sell enough apps to make a living, you will want to use the default sales channel with the most users. This of course is the platform app store, which comes preinstalled on the prime spot. In fact, most users would not know how to install apps any other way, or are warned against that with scary messages. Selling through the app store means you have to pay up and at the same time obey all kinds of rules. The companies behind the mobile platforms themselves can at any time see an interesting market emerging. At that point there is a clear unequality of arms: if they want, the next update will put their own applications preinstalled on hundreds of millions of devices. This giving them a clear and unfair business advantage over anyone else in the market. Meanwhile developers ironically pay for the privilege of being allowed to exclusively develop for the platform concerned, and sell the outcome in the default (and most restrictive) app store. The platform almost certainly has a higher more profit margin from the average developer, even if it is a direct competitor. But what can developers do? Their investment into the software they wrote is hard-wired to the initial choice of platform...?

Non-trivial applications that run on one mobile platform do not run on another, and require additional effort to write in a way where they can. This invisible 'cost of diversity' to the larger ecosystem of creators (which is orders of magnitude bigger) contributed significantly to the "winner takes all" scenario at

platform level. When the European Commission orders some app to be developed for citizens to access its services, crowdsource data gathering or inform them of passenger rights, it does not care about creating someting for the users of the innovative Finnish mobile platform Sailfish from Jolla - or in fact anyone else. If you look at the apps officially published by the European Commission on the app stores, you will not find any app for any European mobile platform ever published there. The same 'selfish' short term considerations will of course be made even more frequently by smaller actors with less deeper pockets, like independent publishers. As a result the market will make the largest platforms larger, and will completely ignore the rest.

In the new mobile world we live in now, control as a user is limited to the very surface of things. Significant privacy and security issues start directly below that surface. You don't really know what the platform actually does while executing apps, and more importantly, who sees your data - or if you are a business, looks at the data of your customers. When you use one of the hundreds of thousands of existing apps and games, you only see the service they provide. But you can't inspect or even see what more they take. What does an app do exactly when you click on the pretty icon? This is very much unlike for instance interacting with a web page, which is fully transparent. As it turns out, mobile apps do lots of things users do not know about, and would not agree with if they did. In some cases literally hundreds of companies have been known to get access to data on the phone.

A consumer-friendly platform should empower the user to notice and take action, or even make it technically impossible. However, the companies that produce the operating systems seem to have other interests. Have you ever wondered why everyone tells you your desktop computer needs a firewall and you are allowed full control to see everything happen. Now stop and think about why your cell phone does not have the very same level of firewall capabilities, but only very much simplified and less capable?

So what can we as a society do in the face of such a complex situation of market failure, anti-competitive practices, perverse incentives and general confusion? How do we give control back to the users? How do we create equal opportunities for European phone manufacturers? How do we stop the unfair "platform tax" on app developers, stimulating employment and startups?

One reasonable direction is to try and lay the ground work for creating viable alternative platforms. Such a fundamental approach is necessary in order to end these extractive practices and the resulting lack of consumer freedom. Smart phones are really just small computers. This means we can build upon plenty of meanwhile mature building blocks and technical work done over decades. In fact, both Android and iOS followed the same path. They were not created from scratch, but based on existing open source projects for desktop and server operating systems. There is nothing magical, it is just engineering work. This is what this project contributes to: it provides an alternative to stock Android. The Replicant project has been building a variant of Android without any unknown parts, unlike the Android which is preinstalled on most phones: all the source code is available for inspection and collaborative improvement as a matter of principle. As new versions of Android emerge, their software needs to be synchronised to keep up with consumer expectations and remain compatible with new applications emerging. Otherwise the user would pay for regaining control by being locked to outdated functionality - which would not really contribute to more users making the choice for more privacy.

Technical description

Replicant is the only fully free operating system for smartphones and tablets. All the other operating systems for smartphones and tablets use nonfree software to make some of the hardware components work (cellular network modem, GPS, graphics, etc). Replicant avoids that, either by writing free software replacement, by tweaking the system not to depend on it, or, as the last resort by not supporting the

hardware component that depends on it. However it is based on Android 6, which is not supported anymore, thus it has way too many security issues to fix, so keeping using this version is not sustainable. This project consists in finishing to port Replicant to Android 9, which now has standardised an interface for the code that makes the hardware components work. Once done, it will also make the free software replacement automatically work on future Android versions.

Replicant and the FSF — **Visit https://NLnet.nl/project/ReplicantUpdate**

NGI0 PET **MobileOS**

YunoHost and the Internet Cube

InternetCube

If you upload a document to a service like Dropbox or Youtube, or collaborate with someone on a document through an online office application, your content probably ends up somewhere on the other end of the world. When you want to retrieve it, you will need to be online. For software and services to function properly, you need to be sufficiently well connected to the internet.

If you happen to be in an rural area where this is an issue, you might not be so lucky. But fortunately there is no need to create this dependency. You can also run software and services locally, meaning that you will always have access to them. The Internet Cube is an initiative to allow users to run services on a small device. It does not fully depend on always having internet connectivity to the outside. In fact, you can take it outdoors and operate it on a simple battery or solar panel - making it a portable and extremely resilient solution. In case of a disaster, the cloud will not be of much use to coordinate work and help bring everything back up. And as a bonus, your data and contents stays close to you - safe from nosy people, and under your immediate and full control. You can easily install additional open source software and services.

Technical description

YunoHost is a free and open-source server distribution that provides a self-hosted alternative to commercial centralized services, and allows people to take back control over their data. Yunohost aims to make server administration accessible to the general public and ultimately make personal servers as common as desktop computers. Based on YunoHost, the Internet Cube project develops an affordable plug-and-play server that can be bought and easily deployed at home by the general public. In addition to its self-hosting capabilities, it provides a privacy-enhancing WiFi hotspot which protects its users from censorship and metadata leaks. And because it is low-power, it can be used even in remote and offline situations.

Support Self-Hosting (association) — **Visit https://NLnet.nl/project/InternetCube**

NGI0 PET **DNS-over-TLS** **OnionRouting** **PersonalServer** **VPN**

Mobile NixOS

Consumers that go shopping for a new cell phone or tablet these days, at the surface have quite a choice. Even the cheapest of mobile phones sold today, is surprisingly powerful compared to that of a couple of years ago. All that seems left for consumers to do is to match their own sense of style and of course budget. If they are really eager, they might compare a limited set of technical specifications: How long does the battery last? How big and bright is the screen? And do games and movies run smoothly? Most users tend to not even bother about that, eager to jump straight to the app stores filled with more applications than a human could feasibly install in their life. What more could a mere user want?

Somewhere in the back of our minds there may be lingering some larger, less happy thoughts. What about security and privacy? Who really is in control of our devices? It is not easy to connect the joyous occasion of our (often much anticipated) purchase of a really cool new gadget with societal resilience, our collective future well-being or any other of the larger economic effects of our individual choices...

In the early GSM era, there wasn't a single dominant operating system from a single vendor. The market was competitive and rather straightforward from todays perspective. Major efforts like Symbian (which ran on the very popular phones of erstwhile market leader Nokia, but also on those of Siemens, Alcatel, Bosch, Sharp, Sony Ericsson etc) were the result of a pragmatic collaboration on more or less equal footing of many manufacturers. These had a shared development responsibility, and equal opportunities. None of them knew how their users actually used the phones they created: that was the business of the customer.

The subsequent rise of the smartphone resulted in market disarray, because the dynamics of the new situation were so different. It wasn't so much a difference in technical quality that set the new masters of the universe apart, it was a complete change of the underlying business model and value proposition few people properly understood - if any.

The real-world cost of developing and maintaining the first generation of mobile platforms was non-trivial, and price competition in the devices was heavy. And then suddenly a no-visible-cost and feature-rich smartphone operating system appeared on the market. It wasn't produced by any of the current competitors or by an open consortium. The source was a single company that had heavily invested into this for strategic reasons. In parallel Apple was able to launch its own effort, take its slick iPod music player and its strong media presence and market visibility in the desktop space. Their premium iPhone line addressed the most luxurious part of the market - also with the help of Google. The CEO's of both companies even sat on each others boards, so the strategy was certainly aligned.

It was a perfect coup. Among the two of them they effectively levered the possibilities of the mobile smartphone platforms, media stores and restricted-access platform-owned app stores to take ownership and control of large parts of the software and content ecosystems at global scale. Traditional phone manufacturers (many of which were European due to the success of the pioneering GSM standard) had historically been just selling a phone at competitive margins (with "no strings attached"). The whole economy of their operations and ecosystem of collaboration was effectively pushed aside by this audacious new strategy. The new Android operating system was funded not by the sale of the product itself, but by the promise of future user data gathering without real limits or much oversight - which had elsewhere proven to be able to create giant revenues. And unlike a desktop computer, a phone is nearly always on. It moves wherever the user goes, and thus it is always near. It has a camera, a microphone and

lots of sensors. When users search for something, they use the default search bar which you control.

So effectively the new "smart" phone was primarily a vehicle for extensive data gathering about users, which could be resold and monetized later on. The manufacturers could get the operating system for free. The small margins that could be made on selling the software to they were negligible compared to the advantages later on. And of course at the time there was still a generation adoration of these "tech darlings" - press wrote lovingly about the "reality distortion field" around Apple's CEO Steve Jobs.

Right from the start this conceiled play was extremely profitable for both of them, allowing lots of subsequent investment - into their platforms, into the developer tools, into marketing and into legislative lobby. The "mobile first" strategy actually worked out better than anyone would have imagined, especially because the mobile phone operating system produced by Google turned out to be more than just a "loss leader". The market funnel of the free option it provided only became visible at the end. Technically advanced and more fair platforms appeared, but were unable to counter the "winner takes all" development in time. At present the vast majority of the phones are sold using one of only two operating systems: Android and iOS. In the absence of effective policy and legislative efforts to curb this unfortunate situation, that market dominance is a hard problem to solve at a technical level.

In our consumer bubble, we actively contributed and still contribute to this. The software stores of both platforms may offer consumers plenty of options at the application level. This seems quite healthy at first. But when you analyse the situation, it is far from how society should want this to be. This all starts with the fact that users do not have to manually install all applications. Apple has full control and puts its own software in pole position. Google is able to make the manufacturers do the same through contractual obligations. The result is the same: a strategic choice of end user applications is preinstalled alongside the platform, and effortlessly available to all users.

Many of us have meanwhile become used to these omnipresent "free" but closed "blockbuster" applications that ship alongside the dominant platforms. As we know from history, for instance through the famous European anticompetition cases against dominant technology companies taking control over web browsers, media players and portable runtimes (Java/C#), preinstalled applications have a huge competitive advantage. Not all users are as technically competent, and this creates enough inertia with consumers to keep manufacturers on a leash. The huge market share of platform 'defaults' like Android's default browser have a deep impact on the market, leaving little room for web developers to follow pretty much all what Google implements - even if they disagree or would actually like to follow proper web standards as produced by W3C. Who can afford for their website or web application to look worse on an operating system with the majority of market share?

Apple holds all the cards closely to its chest, and keeps full control. As long as it has Google as competitor, it feels secure of anti-competition measures. Their main strategy to even increase control is to buy suppliers, or make them sign exclusive contracts keeping others at bay. The defense strategy of Google is publishing most of Android source code. Manufacturers can and have tried to build alternative versions based on that. But in the market real-world control remains tightly with Google through the critical applications which need the "blockbuster" restrictively licensed apps and the larger infrastructure - both of which remain tightly closed. A certain percentage of users will always at some point demand these "free" applications, while others cannot withstand the social lock-in and will actively push vendors to bow down. No small time manufacturer can afford to be out.

The platforms realise this powerful position very well, and are not afraid to lever it. Either a manufacturer is all-in, or all-out: it cannot selectively allow individual users to use blockbuster applications later on. This cut-throat dillemma has left the companies that make the actual phones little choice but to accept unattractive licensing conditions that restrict their freedom to innovate. And even if they do comply with all the demands including a non-disclosure agreement to seal their lips, their license can be withdraw at

any time. In fact this may even happen due to geo-political pressure, as a very large Chinese manufacturer of Android found out to its great dismay in May 2019 when it was banned from future upgrades to Android. While part of this was retracted later, the fact is that such a thing could happen to any phone vendor using Android at any time.

Thhe rigid control over the platform and the app stores was originally meant as a way to secure access to consumer data. These days, it is actually making an awful lot of money on its own. Consumers are paying a huge and very direct cost for the 'free platform' deal of the manufacturers. The dominant mobile platforms both charge developers up to an incredible 30% of their revenues (more than any VAT rate around the world!).

If your company wants to sell enough apps to make a living, you will want to use the default sales channel with the most users. This of course is the platform app store, which comes preinstalled on the prime spot. In fact, most users would not know how to install apps any other way, or are warned against that with scary messages. Selling through the app store means you have to pay up and at the same time obey all kinds of rules. The companies behind the mobile platforms themselves can at any time see an interesting market emerging. At that point there is a clear unequality of arms: if they want, the next update will put their own applications preinstalled on hundreds of millions of devices. This giving them a clear and unfair business advantage over anyone else in the market. Meanwhile developers ironically pay for the privilege of being allowed to exclusively develop for the platform concerned, and sell the outcome in the default (and most restrictive) app store. The platform almost certainly has a higher more profit margin from the average developer, even if it is a direct competitor. But what can developers do? Their investment into the software they wrote is hard-wired to the initial choice of platform...?

Non-trivial applications that run on one mobile platform do not run on another, and require additional effort to write in a way where they can. This invisible 'cost of diversity' to the larger ecosystem of creators (which is orders of magnitude bigger) contributed significantly to the "winner takes all" scenario at platform level. When the European Commission orders some app to be developed for citizens to access its services, crowdsource data gathering or inform them of passenger rights, it does not care about creating someting for the users of the innovative Finnish mobile platform Sailfish from Jolla - or in fact anyone else. If you look at the apps officially published by the European Commission on the app stores, you will not find any app for any European mobile platform ever published there. The same 'selfish' short term considerations will of course be made even more frequently by smaller actors with less deeper pockets, like independent publishers. As a result the market will make the largest platforms larger, and will completely ignore the rest.

In the new mobile world we live in now, control as a user is limited to the very surface of things. Significant privacy and security issues start directly below that surface. You don't really know what the platform actually does while executing apps, and more importantly, who sees your data - or if you are a business, looks at the data of your customers. When you use one of the hundreds of thousands of existing apps and games, you only see the service they provide. But you can't inspect or even see what more they take. What does an app do exactly when you click on the pretty icon? This is very much unlike for instance interacting with a web page, which is fully transparent. As it turns out, mobile apps do lots of things users do not know about, and would not agree with if they did. In some cases literally hundreds of companies have been known to get access to data on the phone.

A consumer-friendly platform should empower the user to notice and take action, or even make it technically impossible. However, the companies that produce the operating systems seem to have other interests. Have you ever wondered why everyone tells you your desktop computer needs a firewall and you are allowed full control to see everything happen. Now stop and think about why your cell phone does not have the very same level of firewall capabilities, but only very much simplified and less capable?

So what can we as a society do in the face of such a complex situation of market failure, anti-competitive practices, perverse incentives and general confusion? How do we give control back to the users? How do we create equal opportunities for European phone manufacturers? How do we stop the unfair "platform tax" on app developers, stimulating employment and startups?

One reasonable direction is to try and lay the ground work for creating viable alternative platforms. Such a fundamental approach is necessary in order to end these extractive practices and the resulting lack of consumer freedom. Smart phones are really just small computers. This means we can build upon plenty of meanwhile mature building blocks and technical work done over decades. In fact, both Android and iOS followed the same path. They were not created from scratch, but based on existing open source projects for desktop and server operating systems. There is nothing magical, it is just engineering work. This is what this project contributes to: it will use the most powerful software packaging system currently available as a basis, and will attempt to make it run on standard mobile phones. This will bring many fundamental building blocks along for free. Of course there is much work needed after that to create something suitable for end users, but it will significantly lower the threshold for the community and provides a great starting point for anyone to join in.

Technical description

The mobile-nixos project seeks to provide a coherent tool to produce configured boot images of NixOS GNU/Linux on existing mobile devices (cellphones, tablets). The goal is to provide a completely integrated mobile operating system, allowing full use of the hardware's capabilities, while empowering the user to exercise their four software freedoms to use, study, share and improve the software.

Visit https://NLnet.nl/project/mobile-nixos

NGI0 PET **MobileOS** **Reproducibility**

Graphics acceleration on Replicant

Replicant

Consumers that go shopping for a new cell phone or tablet these days, at the surface have quite a choice. Even the cheapest of mobile phones sold today, is surprisingly powerful compared to that of a couple of years ago. All that seems left for consumers to do is to match their own sense of style and of course budget. If they are really eager, they might compare a limited set of technical specifications: How long does the battery last? How big and bright is the screen? And do games and movies run smoothly? Most users tend to not even bother about that, eager to jump straight to the app stores filled with more applications than a human could feasibly install in their life. What more could a mere user want?

Somewhere in the back of our minds there may be lingering some larger, less happy thoughts. What about security and privacy? Who really is in control of our devices? It is not easy to connect the joyous occasion of our (often much anticipated) purchase of a really cool new gadget with societal resilience,

our collective future well-being or any other of the larger economic effects of our individual choices...

In the early GSM era, there wasn't a single dominant operating system from a single vendor. The market was competitive and rather straightforward from todays perspective. Major efforts like Symbian (which ran on the very popular phones of erstwhile market leader Nokia, but also on those of Siemens, Alcatel, Bosch, Sharp, Sony Ericsson etc) were the result of a pragmatic collaboration on more or less equal footing of many manufacturers. These had a shared development responsibility, and equal opportunities. None of them knew how their users actually used the phones they created: that was the business of the customer.

The subsequent rise of the smartphone resulted in market disarray, because the dynamics of the new situation were so different. It wasn't so much a difference in technical quality that set the new masters of the universe apart, it was a complete change of the underlying business model and value proposition few people properly understood - if any.

The real-world cost of developing and maintaining the first generation of mobile platforms was non-trivial, and price competition in the devices was heavy. And then suddenly a no-visible-cost and feature-rich smartphone operating system appeared on the market. It wasn't produced by any of the current competitors or by an open consortium. The source was a single company that had heavily invested into this for strategic reasons. In parallel Apple was able to launch its own effort, take its slick iPod music player and its strong media presence and market visibility in the desktop space. Their premium iPhone line addressed the most luxurious part of the market - also with the help of Google. The CEO's of both companies even sat on each others boards, so the strategy was certainly aligned.

It was a perfect coup. Among the two of them they effectively levered the possibilities of the mobile smartphone platforms, media stores and restricted-access platform-owned app stores to take ownership and control of large parts of the software and content ecosystems at global scale. Traditional phone manufacturers (many of which were European due to the success of the pioneering GSM standard) had historically been just selling a phone at competitive margins (with "no strings attached"). The whole economy of their operations and ecosystem of collaboration was effectively pushed aside by this audacious new strategy. The new Android operating system was funded not by the sale of the product itself, but by the promise of future user data gathering without real limits or much oversight - which had elsewhere proven to be able to create giant revenues. And unlike a desktop computer, a phone is nearly always on. It moves wherever the user goes, and thus it is always near. It has a camera, a microphone and lots of sensors. When users search for something, they use the default search bar which you control.

So effectively the new "smart" phone was primarily a vehicle for extensive data gathering about users, which could be resold and monetized later on. The manufacturers could get the operating system for free. The small margins that could be made on selling the software to they were negligible compared to the advantages later on. And of course at the time there was still a generation adoration of these "tech darlings" - press wrote lovingly about the "reality distortion field" around Apple's CEO Steve Jobs.

Right from the start this conceiled play was extremely profitable for both of them, allowing lots of subsequent investment - into their platforms, into the developer tools, into marketing and into legislative lobby. The "mobile first" strategy actually worked out better than anyone would have imagined, especially because the mobile phone operating system produced by Google turned out to be more than just a "loss leader". The market funnel of the free option it provided only became visible at the end. Technically advanced and more fair platforms appeared, but were unable to counter the "winner takes all" development in time. At present the vast majority of the phones are sold using one of only two operating systems: Android and iOS. In the absence of effective policy and legislative efforts to curb this unfortunate situation, that market dominance is a hard problem to solve at a technical level.

In our consumer bubble, we actively contributed and still contribute to this. The software stores of both

platforms may offer consumers plenty of options at the application level. This seems quite healthy at first. But when you analyse the situation, it is far from how society should want this to be. This all starts with the fact that users do not have to manually install all applications. Apple has full control and puts its own software in pole position. Google is able to make the manufacturers do the same through contractual obligations. The result is the same: a strategic choice of end user applications is preinstalled alongside the platform, and effortlessly available to all users.

Many of us have meanwhile become used to these omnipresent "free" but closed "blockbuster" applications that ship alongside the dominant platforms. As we know from history, for instance through the famous European anticompetition cases against dominant technology companies taking control over web browsers, media players and portable runtimes (Java/C#), preinstalled applications have a huge competitive advantage. Not all users are as technically competent, and this creates enough inertia with consumers to keep manufacturers on a leash. The huge market share of platform 'defaults' like Android's default browser have a deep impact on the market, leaving little room for web developers to follow pretty much all what Google implements - even if they disagree or would actually like to follow proper web standards as produced by W3C. Who can affort for their website or web application to look worse on an operating system with the majority of market share?

Apple holds all the cards closely to its chest, and keeps full control. As long as it has Google as competitor, it feels secure of anti-competition measures. Their main strategy to even increase control is to buy suppliers, or make them sign exclusive contracts keeping others at bay. The defense strategy of Google is publishing most of Android source code. Manufacturers can and have tried to build alternative versions based on that. But in the market real-world control remains tightly with Google through the critical applications which need the "blockbuster" restrictively licensed apps and the larger infrastructure - both of which remain tightly closed. A certain percentage of users will always at some point demand these "free" applications, while others cannot withstand the social lock-in and will actively push vendors to bow down. No small time manufacturer can afford to be out.

The platforms realise this powerful position very well, and are not afraid to lever it. Either a manufacturer is all-in, or all-out: it cannot selectively allow individual users to use blockbuster applications later on. This cut-throat dillema has left the companies that make the actual phones little choice but to accept unattractive licensing conditions that restrict their freedom to innovate. And even if they do comply with all the demands including a non-disclosure agreement to seal their lips, their license can be withdraw at any time. In fact this may even happen due to geo-political pressure, as a very large Chinese manufacturer of Android found out to its great dismay in May 2019 when it was banned from future upgrades to Android. That can happen to any phone vendor using Android at any time.

Thhe rigid control over the platform and the app stores was originally meant as a way to secure access to consumer data. These days, it is actually making an awful lot of money on its own. Consumers are paying a huge and very direct cost for the 'free platform' deal of the manufacturers. The dominant mobile platforms both charge developers up to an incredible 30% of their revenues (more than any VAT rate around the world!).

If your company wants to sell enough apps to make a living, you will want to use the default sales channel with the most users. This of course is the platform app store, which comes preinstalled on the prime spot. In fact, most users would not know how to install apps any other way, or are warned against that with scary messages. Selling through the app store means you have to pay up and at the same time obey all kinds of rules. The companies behind the mobile platforms themselves can at any time see an interesting market emerging. At that point there is a clear unequality of arms: if they want, the next update will put their own applications preinstalled on hundreds of millions of devices. This giving them a clear and unfair business advantage over anyone else in the market. Meanwhile developers ironically pay

for the privilege of being allowed to exclusively develop for the platform concerned, and sell the outcome in the default (and most restrictive) app store. The platform almost certainly has a higher more profit margin from the average developer, even if it is a direct competitor. But what can developers do? Their investment into the software they wrote is hard-wired to the initial choice of platform...?

Non-trivial applications that run on one mobile platform do not run on another, and require additional effort to write in a way where they can. This invisible 'cost of diversity' to the larger ecosystem of creators (which is orders of magnitude bigger) contributed significantly to the "winner takes all" scenario at platform level. When the European Commission orders some app to be developed for citizens to access its services, crowdsource data gathering or inform them of passenger rights, it does not care about creating someting for the users of the innovative Finnish mobile platform Sailfish from Jolla - or in fact anyone else. If you look at the apps officially published by the European Commission on the app stores, you will not find any app for any European mobile platform ever published there. The same 'selfish' short term considerations will of course be made even more frequently by smaller actors with less deeper pockets, like independent publishers. As a result the market will make the largest platforms larger, and will completely ignore the rest.

In the new mobile world we live in now, control as a user is limited to the very surface of things. Significant privacy and security issues start directly below that surface. You don't really know what the platform actually does while executing apps, and more importantly, who sees your data - or if you are a business, looks at the data of your customers. When you use one of the hundreds of thousands of existing apps and games, you only see the service they provide. But you can't inspect or even see what more they take. What does an app do exactly when you click on the pretty icon? This is very much unlike for instance interacting with a web page, which is fully transparent. As it turns out, mobile apps do lots of things users do not know about, and would not agree with if they did. In some cases literally hundreds of companies have been known to get access to data on the phone.

A consumer-friendly platform should empower the user to notice and take action, or even make it technically impossible. However, the companies that produce the operating systems seem to have other interests. Have you ever wondered why everyone tells you your desktop computer needs a firewall and you are allowed full control to see everything happen. Now stop and think about why your cell phone does not have the very same level of firewall capabilities, but only very much simplified and less capable?

So what can we as a society do in the face of such a complex situation of market failure, anti-competitive practices, perverse incentives and general confusion? How do we give control back to the users? How do we create equal opportunities for European phone manufacturers? How do we stop the unfair "platform tax" on app developers, stimulating employment and startups?

One reasonable direction is to try and lay the ground work for creating viable alternative platforms. Such a fundamental approach is necessary in order to end these extractive practices and the resulting lack of consumer freedom. Smart phones are really just small computers. This means we can build upon plenty of meanwhile mature building blocks and technical work done over decades. In fact, both Android and iOS followed the same path. They were not created from scratch, but based on existing open source projects for desktop and server operating systems. There is nothing magical, it is just engineering work. This is what this project contributes to: it will help for alternative operating systems to use the graphics hardware inside modern phones to lower their power usage. The Replicant project has been building a variant of Android without any unknown parts, unlike the Android which is preinstalled on most phones: all the source code is available for inspection and collaborative improvement as a matter of principle. This means R&D is necessary to replace all the closed components with secure, open ones. Smooth graphics make a lot of difference to users, but they are also important to reduce energy usage: without the ability to use the dedicated graphics module of the phone, the computations will be done (rather

inefficiently) by the core processor of the phone. This means the user would pay for regaining control by additional battery usage - which is obviously not really an option for most consumers, given the already problematic battery usage dependency....

Technical description

The project aims to create a free software graphics stack for Replicant 9 that is compatible with OpenGL ES (GLES) 2.0 and can do software rendering with a decent performance, or GPU rendering if a free software driver is available. Replicant is a fully free software Android distribution that puts emphasis on freedom, privacy and security. It is based on LineageOS and replaces or avoids every proprietary component of the system. Replicant is so far the only distribution for smartphones that is endorsed by the Free Software Foundation as meeting the Free System Distribution Guidelines. Due to its strict commitment to software freedom, Replicant does not use the proprietary GPU drivers that shipped within other Android distributions. The project aims to put together a new graphics stack for the upcoming Replicant 9 that is GLES 2.0 capable. The project will then focus on improving the performance by fine tuning its OpenGL operations and leveraging hardware features. At last, focus will swift into the integration of the Lima driver, a free software driver for ARM Mali-4xx GPUs, which will allow to offload some GLES operations to the GPU. This will greatly increase graphics performance and thus usability.

Replicant — **Visit https://NLnet.nl/project/Replicant-graphics**

NGI0 PET **MobileOS**

TLS-KDH mbed

Imagine you would work in an organisation with thousands of employees, like a government. It would be important to properly manage who gets to access which computer systems. A large part of this would need to happen automatically: if you want to print out a document on an printer in the hallway, or visit the intranet to view the menu, you do not want users to have to log in every time. Luckily, people have worked out powerful mechanisms that allow you to log in when you get in the office in the morning, and which will negotiate everything else automatically without bothering the user.

Now think about the internet. That is much much larger and way complex than a single organisation. And yet it does not have any mechanism to manage who gets to access which computer systems. So we do have to log in every time. And users are very bothered by this. The ARPA2 project has successfully produced a working solution (called TLS-KDH, hence the name of the project). This is a very creative and for some unexpected combination of a number of robust proven technologies that can together deliver a highly secure and extremely fast mechanism to authenticate users. It also offers anonymous encryption of a connection before revealing identities of clients and servers.

The ARPA2 community is now aiming for IETF standardisation of this technology. In order to make this possible, it needs an independent second implementation of the new protocol. The project will deliver this implementation, in a popular open source library aimed at embedded systems. Combined with work on peer-to-peer mechanisms, this will potentially allow devices to securely discover and connect to each other.

Technical description

TLS-KDH (http://tls-kdh.arpa2.net/) is a mechanism that adds Kerberos authentication to the Transport Layer Security (TLS) network protocol. TLS-KDH is developed under the flag of ARPA2 (www.arpa2.net) and is formalized in the form of a draft Internet specification. Furthermore, a successful prototype implementation has been built and integrated into GnuTLS. Making this prototype code production ready is well underway and in its final stage.

In order for TLS-KDH to become an Internet Standard the IETF requires at least two working implementations. To provide the IETF with two TLS-KDH implementations and to address the embedded world with a TLS-KDH capable TLS library we chose MbedTLS as our second library. The TLS-KDH mbed project's goal is to implement the TLS-KDH functionality in the MbedTLS library.

But why do we want to implement Kerberos authentication in the first place? Well first of all, the Kerberos protocol is quantum computer proof. That means that we can use this mechanism in the (future) presence of quantum computers. Since TLS is one of the most widely used security protocols on the present Internet having such mechanism would be a welcome addition. Secondly, Kerberos employs a centralized architecture as opposed to X.509 which is distributed. Adding TLS-KDH gives the user a choice which architecture (and implied pros and cons) to use. For a more extensive overview of advantages of TLS-KDH we refer to the project's homepage (http://tls-kdh.arpa2.net/).

Visit https://NLnet.nl/project/TLS-KDH-mbed

NGI0 PET **Cryptography** **EmbeddedSystems** **IETF** **TLS**

S A S L X M S S

Digital signatures are very convenient, for consumers, governments and businesses alike. Most documents that need to be signed these days are 'digitally born': they first exist inside a computer. Signing in the conventional way (on a piece of paper) is both very time-consuming and eco-unfriendly. Each document has to be sent to a printer, someone needs to collect the printout and get it back to their desk, find a pen that works (sigh) and sign it. And in many cases the document at hand will need to be rescanned shortly after, in order to be sent by mail.

Digital signatures are also more secure. Signatures are basically just a few lines of ink from a pen. When

you look close, no two signatures from the same person are the same. The natural variance means the origin and history (and thus the authenticity) of those "ink proofs" can be really hard to technically verify properly. With a little practise, a fake signature is easy to create - in fact, in most cases any signature will do. What can people use to verify? While the actual proof is not so good, in a lot of practical cases we have other reasons why we trust a document. For instance because we got it in person, or know the document has been securely locked away by a trusted party. The common practise to scan a "real" signature and cut and past it as an image inside a document, operates on that same premise: we get the document from a trusted source, and so we can trust it - making the addition of the signature more of a ritual.

However, on the internet we do not have such guarantees. As countless phishing mails from banks and credit card companies will show, cloning some existing legitimate document is trivial. On the internet trust and trustworthiness is low, while speed of acting is high. That is why we need digital signatures as a basis to delegate trust: to sign software, documents, etc.. A digital signature is often used at points where you hand over some control to other, so it is really important to get this right.

Digital signatures use advanced math to guarantee authenticity. We are considering trusting something, but we need to make sure the person or organisation that has supposedly signed something, in fact did so. Conventional computers as we use today in our offices and homes would need many thousands of years to break most digital signatures. This is more than orders of magnitude of a human lifetime, as well as the lifetime of most of human dealings. So digital signatures have been recognised as a practical and convenient way to work, with much better security than their ink predecessors. It is no wonder that digital signatures continue to increase in adoption everywhere.

One urgent problem with todays digital signatures however, is that a new type of computer technology is threatening some of the assumptions we made above. These new devices (so called "quantum computers") are assumed to be capable of performing some common types of calculations in parallel at such a speed, that it would be possible to fake current types of digital signatures much faster. So much faster in fact, that people rightfully worry about important things they sign today. It would not be the first time that the pace of development of computers takes people by surprise.

The answer is of course to recognise the threat and innovate, by making the digital signatures smarter. Not all calculations can be sped up by the new quantum computers as well, as least that is the common assumption with computer scientists. So the strategy devised is often called shifting to "quantum-proof" of "post-quantum" solutions, though the latter name is a bit weird given that the quantum computers will continue to exist in parallel with normal computers.

The project SASL XMSS will take an innovative new digital signature type from top European scholars that is "quantum-proof". The math in XMSS works in such a different way, that the quantum computers are not supposed to be able to crack them. And the size of the digital signature is reduced to less than 25% compared to the best alternative we have today. The project connects this new digital signature to an existing standard called SASL, which is the most prevalent internet framework for authentication and security. And it aims to implement this solution in a popular open source libary. That means when people install the latest version of that library, along with the update they will automatically get the exciting new capabilities that this project brings. That double pronged strategy will make the new digital signature type become available to many applications at once. This should help tremendously with adoption. The expected end result is a great degree of trustworthiness, meaning that users can continue to trust others on the internet with confidence - and without additional hassle.

Technical description

Simple Authentication and Security Layer (SASL) is an authentication and data security framework. The framework defines a structured interface to which SASL mechanisms must comply. These mechanisms can then be used by application protocols in a uniform manner. XMSS provides cryptographic digital signatures without relying on the conjectured hardness of mathematical problems. Instead, it is proven that it only relies on the properties of cryptographic hash functions. XMSS provides strong security guarantees and is even secure when the collision resistance of the underlying hash function is broken. It is suitable for compact implementations, is relatively simple to implement, and naturally resists side-channel attacks. Unlike most other signature systems, hash-based signatures can so far withstand known attacks using quantum computers. The SASL XMSS project's goal is to implement the XMSS system as a SASL mechanism in one of the publicly available open source SASL libraries.

ARPA2 — **Visit https://NLnet.nl/project/SASL-XMSS**

NGI0 PET **Cryptography** **IETF** **Post-Quantum** **SASL**

E G I L S C I M c l i e n t

These days organisations work with lots of external services via the internet. When those services require a login, and most do, there is some significant book-keeping to do. Sometimes your organisation will pay a flat rate to a service provider, but many other times you will pay per user or per use. The user may need to receive a personal environment, coupled to their role in your organisation. A manager or teacher may get another view than a student or a flex worker. Other times you as an organisation use the service in such a way, that you need to be able to exchange information with other systems from other provider. That means service providers need adequate and up to date information, in order for their service to properly work.

And all of this happens at quite a pace. New people join. Tasks shift. People leave. Policies change. Someone has to fill in for someone else. Laws and regulations change. Etcetera. In practical terms this means a lot of work adding and removing accounts and approving changes everywhere. Of course, you can go the easy route. If you relax on quality, you can reduce some overhead and just work with a few big vendors. Of course there is a price you pay for that convenient consolidation. You will likely not get the best of breed solutions at the best price. You are less open to innovation, only from the parties you work with. There is also a price in terms of dependency and the privacy of your users as well. A large service provider gets to learn more about your users, more than you might like - think of companies that combine advertising and user tracking with regular services. There is also of course legislation like the General Data Protection Regulation which you need to observe.

If you want to work with the best services, at the best price, you want to keep your options open. But you also want to automate, because otherwise you would not just go crazy - you would inadvertently make mistakes. This is why you will want to use interoperable standards over private arrangements. You only need to implement the right standard once, and then you can reuse it across services. Standards make management and auditing a lot easier. One young but very promising standard is SCIM, which stands for System for Cross-domain Identity Management. This already works very well inside many organisations to exchange information between your internal IT system and that of vendors.

However, there are use cases where you need one additional layer of exchange. In education, many schools and institutions like universities work together through so called federations. That means they are able to match up each others users, and have organised a smooth way of sharing resources among the different organisations. The SCIM standard is not yet able to handle this kind of aggregate collaboration, at least not in a standardised way. But of course that can be changed. This project will take a number of very experienced internet engineers that already have several internet standards on their track record. It will add a very concrete use case with real stakeholders (such as the Swedish National Agency for Education) and an open source system which is already used by many schools. With those ingredients and the grant from NGI Zero it aims to set a new internet standard for adding federation to the SCIM standard - part of which means implementing it in quality open source libraries that anyone can then adapt and reuse without limits for any purpose whatsoever.

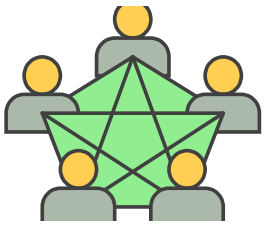
This systemic capability will increase consumer choice and agency, and will reduce the pressure of the market towards consolidation with a few large suppliers by making it just as easy to work with small vendors. This way, it will stimulate and nurture future innovation in a structural way - which is an enormous contribution for what is on the outside a modest project with a modest budget compared to most efforts.

Technical description

Managing student information in an effective, secure and GDPR compliant way is crucial for the digitalized school. EGIL is an open source client that facilitates the exchange of student information to external providers of study material or administrative services in a standardized way. It supports attributes based on SCIM (RFC 7642-7644) and extensions, it provides an interface to common directory services and supports federated solutions between a large number of school principals and service providers. This project will improve EGIL's federative capabilities, submit an Internet-Draft on the subject federated accounts provisioning, as well as providing a proof of concept for using SCIM as the standard for exchange of student information. This will eliminate the problems caused by using several different exchange protocols and formats between school principals and service providers.

The Internet Foundation In Sweden — **Visit https://NLnet.nl/project/EGIL-SCIM**

NGI0 PET **Education** **Foundation** **GDPR** **Interoperability**



According to the Universal Declaration of Human Rights established just after the second world war, all humans have a number of inalienable rights. These are quite fundamental rights such as the freedom of association, freedom of movement and freedom of speech. These rights were instituted to protect normal people from those who can wield power over them, such as governments. The rights enable people to collectively counter injustice, corruption (in particular within institutions that are supposed to be incorruptible such as the police, the court and government) and inequality. Surely, it is not a goal in itself of most governments to curb individual rights, at least not in Europe. But when these rights are not respected, things can get out of hand quickly. Human rights are part of social hygiene.

In the datacene era, however, these rights seem to be under threat. People are constantly observed from all angles of the internet. A lot of data and metadata is being collected about their behaviour. Who is talking to whom, when and where. That data can unexpectedly pop up years later, in a completely different context - and be weaponised against them at that time. This pervasive social logging has a chilling effect on the mind of the individual: if everything you do can be used against you at any point in time, it is best to not do anything. So the piling up of data and observational metadata from corporate and private surveillance hurts the collective ability for groups of like-minded people to express themselves and pursue their common interests.

In order to really enjoy these fundamental human rights in the computer age, we need to take countermeasures. We need to restore confidentiality within social groups as a technical principle. Of course, the possibility of regaining confidentiality is also useful for businesses and governments that sometimes need to hold their card close to their chest to prevent espionage. The project will build a communication tool based on sound scientific principles. The tool will continually shuffle 'black boxes' of unknown digital content among a group of people. In some cases, there might be a message inside the box, in other cases the box will be empty. The boxes are all the same size and weight, so you cannot tell which is which from the outside. The content is protected by using complex mathematical operations and protocols that prevent snooping. For a human, this would be very tiring and unpractical. But for a computer, this is just another job it can do in the background. This setup creates so called 'plausible deniability': you cannot know what is being discussed. You cannot know if someone actively participated in a discussion, or just passively participated. And if you use some other level of indirection that hides the remaining data traces like your IP address, no-one (outside the group) can learn anything. Such a scheme significantly reduces the amount of available metadata to outside observers. If we are serious about fundamental rights such as the freedom of association, freedom of movement and freedom of speech in the digital world, tools like this can make a world of difference.

Technical description

The aim of the proposed project is to design and implement an open source library that implements the

so-called Dining Cryptographer's network or DCnet (first proposed by David Chaum in 1998). Existing implementations suffer from poor efficiency (e.g. high computation and/or communication cost) or limited security (e.g. when a malicious participant can disrupt the communication). The project will produce cryptographic primitives and protocols that help to bring untraceable communication (e.g. untraceable instant messaging, file transfer, IP telephony) closer to practice. We will implement the most recent advances in cryptographic research (e.g. zero-knowledge proofs) and engineering (e.g. highly optimized arithmetic on elliptic curves and finite fields) into account to maximize both security and efficiency.

University of Luxembourg — **Visit https://NLnet.nl/project/DCnets**

NGI0 PET DCnet InstantMessaging Metadata Observability

R e o w o l f

Reo

Many of the underlying core technologies we use on computers, date from an era when the internet was in its infancy. Security wasn't a primary concern, and thus wasn't part of the design decisions. Sockets are such a technology, dating back to the early eighties. Sockets are a convention that used by all the software that needs to communicate across a network. A socket basically is a placeholder of the network connection inside the computer. Applications will send traffic to that placeholder - and the operating system will take care of the rest .The technical design was flexible enough to survive the intermittent decades, but offered users almost no control or insight as to what is happening. Essentially it functions as a software hose connecting the inside of the operating system with the outside world.

The key problem we face today is the fundamental security and trust issues which that design is completely ignorant of. It doesn't understand that not every application should be allowed to do the same things. In particular, as soon as a user is allowed to use a socket because of some legitimate application, all the applications that belong to the same user can use it. Multiple applications can use the same socket at the same time, and none of them would be able to see what other applications are doing. Because data is actually being sent from your computer to the outside, that can become a critical issue real fast. This design is part of pretty much every operating system currently on the market. And worse: the technology is not just present, but is actually still heavily used.

The Reowolf project is geared to providing a next generation solution to make network connections safer. Instead of a hose that bits are pumped trhough, it provides a smart connector. That connector allows to synchronise data from multiple sources from inside to outside and vice versa. The key difference between the technology developed within Reowolf and a classic socket is that Reowolf will allow to do so in a controlled way. Unlike a socket, such a connector allows for high-level verification, compilation and optimization techniques. So you can clean up, or selectively filter, the incoming and outgoing traffic. This significantly and directly improves the control the user has over the connections the computer makes across the internet, and allows for many interesting user benefits such as dynamic configuration. Reowolf thus offers a systems level solution for the next generation internet.

Technical description

The Reowolf project aims to replace a decades-old application programming interface (BSD-style sockets) for communication on the Internet. In this project, a novel programming interface is implemented at the systems level that is interoperable with existing Internet applications. Currently, to increase quality of service (e.g. intrusion detection, latency and throughput) non-standard techniques are applied. Internet service providers resort to deep packet inspection to guess applications intent, and BSD-style socket programming is error-prone and tweaking is fragile. This project resolves these problems: it provides support to middleware to further improve quality of service without having to give up on privacy, and makes programming of Internet applications easier to do correctly and thus more reliable.

CWI — **Visit https://NLnet.nl/project/Reowolf**

NGI0 PET **FormalProofs** **Fundamental** **POSIX** **SystemsProgramming**

G N U M e s

When you start up your computer, you will probably think twice before you download some random piece of software from the internet and run it. You know that doing so could allow unwelcome guests to your computer and your data. Your computer might even end up in a bot net. So when you see some nice piece of software, you will ask yourself the question: can I really trust the software? Perhaps you will check the origin it comes from. Better safe than sorry.

Did you miss checking something, though? What about the software that is already on your computer before you started? A computer is not much use without an operating system. While most computers are sold with an operating system, actually you have the choice to remove that and install something different. Have you thought about the trustworthiness of that fundamental piece of software - your most fundamental travel companion on the wild west of the internet? Trustworthiness is essential. When an operating system has a so called 'back door' (either intentionally or not), someone could extract whatever user data - like personal pictures or home movies - from your computer. And the worse thing: without you ever finding out. The operating system guards all the other software, and warns you when you install software from the internet. But itself, it doesn't have to ask for permission. Ever. It doesn't just have "access all areas": in fact, it runs the whole show.

With commercial software like Microsoft Windows or Mac OS X that you get delivered when you buy a computer, trust in what their closed operating system does will of course always be a leap of faith: as a user you essentially are given no choice. In proprietary systems you do not have the freedom to study the source code, or to control what really happens. So you either trust the vendor, or you'd better not use it.

For an increasing amount of people, after the revelations from whistleblowers like Edward Snowden, that "leap of faith" is not so obvious anymore. They prefer to use free and open source operating systems like GNU Linux, FreeBSD and OpenBSD. These are technology commons: the people that wrote the software allow you to inspect the source code. Even more so, they give you the source code to do anything with it that you like. So you don't just blindly have to take their word for it and trust them, you can take matters into your own hands.

But until now, there were some parts that would escape introspection. You would have to trust them, not because the people involved didn't want to share everything with you - but because they couldn't. When an operating system is loaded, you need to get the computer into a state from where it can manage itself. The necessary software is poetically called a "binary seed", because it is, well, a very very long string of bits. In fact, a few hundreds of millions of bits. And of course, that amount of information without any hints or cues as to how they interact are rather hard to grasp - and thus a potential point of risk.

What if we could get the computer into the right state through a different path? The GNU Mes project aims to replace the traditional "binary seed" by something orders of magnitude smaller. The really clever and innovative part is that they will add the more complex parts to a "second stage", which is being created from scratch by the project, in a human understandable programming language. This two stage approach allows to make all of computing more trustworthy, in a very controlled way - and will grant our future selves the ability to use computers without taking a leap of faith. If the project succeeds, it will make a very fundamental contribution to the security of the next generation internet.

Technical description

GNU Mes was created to address the security concerns that arise from bootstrapping an operating system using large, unauditable binary blobs, which is common practice for all software distributions. Mes is a Scheme interpreter written in a simple subset of C and a C compiler written in Scheme and comes with a small, bootstrappable C library. The Mes bootstrap has halved the size of opaque binaries that were needed to bootstrap GNU Guix, a functional GNU/Linux distribution that focusses on user freedom, reproducibility and security. That reduction was achieved by replacing GNU Binutils, GNU GCC and the GNU C Library with Mes. The final goal is to help create a full source bootstrap for any interested UNIX-like operating system. After three years of volunteer work this funding will enable us to take another big step forward and reach an important new milestone in creating more auditable secure software distributions.

joyofsource.com — **Visit https://NLnet.nl/project/GNUMes**

NGI0 PET **BinarySeed** **Bootstrap** **Minimisation** **OperatingSystem**

Social media are important for many people to stay in touch, but they also influence the way we act ourselves. Their persistent presence in our lives creates a lot of pressure to be someone else then we are. This is not a coincidence: the developers have designed the software to keep us 'engaged' at any cost. If we sleep less well because of a message we got just before we went to sleep, that is fair game. We may wake up with a hundred messages we "missed", which creates a lot of stress. But it keeps us engaged, and keeps the advertising cash flowing.

Some people question whether that social frenzy is actually the best design for our society. Long ago, before we even had steam engines, people used to make long business voyages with sailing boats. The sailors would be without contact with home for weeks, sometimes even months. Messages home were not really possible, other than by relay - boats going into the other direction. Even in such an extreme situation, people were able to enjoy life.

Manyverse is a social application that allows for people to relay messages through friends, and discover new ones. Manyverse is part of a global open source messaging movement and ecosystem called Secure Scuttlebutt (often shortened to SSB). This network has all the features you would expect: posts, threads, likes, profiles, etc. Someone in need of such a tool can start using Manyverse (or any of the other tools that work with this same protocol), without any other technological dependencies or registering an account somewhere. There is no central registry, simply because it is not needed. Manyverse was created to allow users of smartphones to be part of the Secure Scuttlebutt network.

Manyverse works 'off-grid' as well as on a (local) network, so you can use it pretty much any way you want. And it can be used to convey sensitive messages, without leaving unnecessary traces or endangering the people carrying the message - by design they have no access to the messages of others. If you want, your message will never touch the internet unencrypted. Secure Scuttlebutt is meant as a different, human-centric social network, which acts more in line with how people should want to engage - rather than how they are expected and pushed to engage by companies. Unhappy with the social media you are surrounded with, or need a solution that allows you to communicate offline? Why not try and see if this more relaxed, human-size model fits better with your way of life?

Technical description

Manyverse is a social networking mobile app, implemented not as a typical cloud service, but instead on a peer-to-peer network: Secure Scuttlebutt (SSB). The mobile app locally hosts the user's database, allowing them to own their personal data, and also use the app when offline. Data can sync from one mobile device to another, via Bluetooth, Wi-Fi, or Internet. Free and open source software.

Manyverse — **Visit** **https://NLnet.nl/project/Manyverse**

NGI0 PET MobileApp Offgrid Online SocialNetworking

DNSSEC Key Signing Suite

You mobile phone doesn't really understand what for instance "NLnet.nl" or "www.wikipedia.org" mean, when you type either name into a web browser. Being a web browser, it will not come as a surprise that the software will assume you want to visit some website. But it doesn't really know where that website is located on the internet. It doesn't need the physical place of course, but it needs the number that unique identifies the web server so it can connect.

All your mobile phone does know, is how to ask that question to other, specialised computers. These computers actually also probably don't know, unless they have recently answered the same question for another user. Names can change really fast for good reasons, so you would need to refresh this data a lot - otherwise users would end up on the wrong computer. The computers you send your question to, will have a good working understanding how the so called "domain name system" of the internet works. More in particular, the name we asked for needs to be cut up in smaller pieces that need to be read backwards.

There is a short code at the end, which points to a country - or provides some other meaningful clue as to where more information can be learned about the still unknown parts of the name. The short code (which people tend to call a "top level domain") is uniquely managed by a single professional organisation. It is actually called a registry because that is literally what it does: it registers all the names people use. One organisation registers names which end in ".nl", others take care of ".org" or ".eu". There is an invisible list that has all the top level domains on it. This list is called the "root zone" of the internet, and it is quite important because everything that uses a name will need to start its search there.

It is the registry organisation which can provide additional details about the segment next up, in this case "wikipedia" or "NLnet". But it will still not know all the answers itself, so your question will travel to yet more computers. We are getting close now to the computers that these organisations have selected to take care of their domain name. In the case of NLnet this computer will be able to give the right answer straightaway, and this answer needs to be sent back across the entire chain of computers. In the case of wikipedia, the fact we still have a "www" part to look for, could mean that inside Wikimedia foundation there would still be another computer which could be responsible for everything under that label. The same could go for fr.wikipedia.org or ro.wikipedia.org - the label www is only meant for human consumption, but computers actually don't need it. After just a few steps, we started getting part of the answer we were looking for, and all of these parts are sent back to your phone. And at some point in time, we have the entire answer.

Now how do we know that the answer we obtained in this recursive way really can be reliably traced back to the right computers running the root zone of the internet - the so called root servers? Simple, because there are digital signatures on each part of the answer. For the root zone, there is a so called cryptographic key which is distributed widely - there is only one for the whole world. Chances are you have that key on your phone or computer, and your internet provider certainly has. When the question arises where .org is, this digital signature will make sure you know the right internet address to go. There you can ask the organisation that is responsible for the next part of the answer. For each computer that

gives another level of detail, new signatures are added. So in the end you should have a complete proof for every step: or in other words, a trust chain.

Those signatures on the answers are really important: your computer has nothing else to underpin trust. If someone is able to falsify these signatures, they could use this to manipulate answers for everything "below". This includes not just domain names, but also other things people have put into the DNS like certificates. So great effort is spent on making sure everything happens in a really safe way, leaving nothing to chance. And as a matter of technical hygiene, the cryptographic key needs to be changed regularly. For the root of the internet, there is in fact a grandiose ceremony which involves flying in people from all over the world to closely watch how the keys are replaced. The event is attended by journalists and observers. Of course this kind of public event is really expensive, but there is only one root zone of the internet and it only happens once every couple of years - so it is kind of a special event.

Organisations running a top level domain, also need a thorough procedure. They may not have the same budget, however. True, some of the larger organisations may have multi-million euro annual budgets, but others certainly do not. So far there was not a canonical procedure shared among these organisations, meaning that there was room for ambiguity and misinterpretation that could have serious consequences for the economy and society alike. Also, policy makers responsible for national and regional policies were unsure what was expected from them.

This project aims to fill this hiatus. It will design a tight and secure procedure that gathers all the best practices for key signing for domain name registries. The project is a collaboration between European experts that are responsible for the software that has been running on some of the root servers of the internet for many years, and a not-for-profit from the USA that actually operates several top level domains across the world. Their combined experience and technical expertise will make this a very important contribution to establishing trustworthy and secure operational practices on the internet.

Technical description

DNSSEC provides trust in the DNS by guaranteeing the authenticity and integrity of DNS responses. As DNS is of fundamental importance to most Internet communication, this is a vital function that needs safeguarding. Beyond providing trust in the DNS, DNSSEC is a key enabler for other technologies that improve the security, privacy and trust of Internet users. In the DNSSEC Key Signing Suite project we build a set of tools, scripts and guidelines (a playbook) to facilitate simple key signing with a standardised ceremony that has automated checks and audits where possible. The impact of this will be twofold. First, it leads to reliable, predictable and verifiable key ceremonies, which improves the trust in DNSSEC. Second, it will significantly ease the burden of operation, bringing the use of a validated and trustworthy signing procedure within reach for many more DNSSEC operators than today (e.g. smaller or less profitable top-level domain operators).

Stichting NLnet Labs — **Visit https://NLnet.nl/project/keysigningsuite**

NGI0 PET DNSSEC Foundation IETF KSK KeyManagement TLD

We live in an time where it seems there are video camera's everywhere. Not just on a payment terminal, a traffic sign or at the entrance of an office building either. Every day hundreds of millions of people take their phone from their pocket and start filming themselves and others. And they do not just record it for their personal use, they tend to put it online. But in many instances you are not the only one captured on screen. The person next to you might not want to be put online, for whatever reason. They might be underage. They might have their credit card out. In the "selfie" mindset, those concerns may not matter much. After all, what can they do? All they want is to capture the moment, and you just happened to be in the wrong place at the wrong time.

Now combine this with the ever increasing power of computers to process video. Todays capabilities go far beyond face recognition in pictures. Computers can now almost realtime process video recordings and live streams. Connect that in your mind with the insane amount of information about people already available online, for instance through social media. This could end up as the 'coup de grace' for privacy in the public sphere: point your camera to a girl in a bar, and their name, income, hobbies and past sex life automatically pop up. The creep factor of this is enormous, and so is the potential for abuse. That is probably the reason why some of the largest privacy violators on the planet have invested literally billions of dollars in building technical capabilities to that effect.

All of a sudden, you end up with a spillover from the digital world into the real world. If we want to protect privacy in the public sphere, we will need to create technologies that can counter this trend of crowd-sourced public surveillance. It is probably infeasible and not desirable to ban camera's. There are many beneficial uses for them too, after all. So the next best thing is to create technologies that can protect the privacy of people that would be captured on video against their will. We can use video recognition technology the reverse way: wipe out everything concerning the people you did not explicitly get permission from. It doesn't really matter to your vlog viewers if you cannot see the faces of the people walking behind you - or randomise their clothes. All we need it the technology to do this automatically. This is the goal of the VFRAME project: it aims to build a first, exploratory tool as a first step to restore visual privacy. In the end, this is probablyu something that should be turned into legislation so that all consumer devices behave like this by default. So that next time that first kiss on a bench in the park, will not be part of internet history.

Technical description

Visible data shares many of the same risks as wireless data yet visual privacy is often overlooked in the field of information security studies as separate and less relevant. As computer vision becomes increasingly adept at understanding the visual domain, differences between existing protocols for processing wireless data and emerging protocols for processing visible data (computer vision) become less apparent. Ultimately, images and video are wireless data too, and they are exposed to an increasing

number of attacks on visual information privacy with less technologies for protection. Visual Defense Tools will explore and prototype computer vision methods for visual privacy through visual obfuscation and minimization techniques, mostly related to biometrics. The goal will be to build a conceptual road map and functional open-source prototypes to stimulate future development of more accessible visual privacy technologies.

Visit https://NLnet.nl/project/Vframe

NGI0 PET **Biometric** **Deanonymisation** **ImageProcessing** **VisualPrivacy**

ARPA2 resource ACL and HTTP SASL modules for NGINX

For some use cases, web servers need to be a bit smarter. They are really good at serving up web pages really fast, which is the core of their task. Yet out of the box they understand very little of what they are doing, or who they are interacting with. That is pretty much left to the applications running on such a server. In some instances, it could be quite beneficial if some of these responsibilities could be delegated to the webserver. That way, developers can focus on applications themselves rather than on keeping unwanted or unauthorised visitors out.

We all want websites to be as secure as possible. We also want to grant users as much privacy as we can. Technical measures can of course be taken at the level of the application, as is done traditionally. But it is quite easy to make mistakes, and a lot of work. An awful lot of work. Developers waste a lot of time on implementing the same steps over and over. It would be a lot easier if some web tool can just assume that only valid users would enter, and that some reliable source would authoritatively tell them what rights they need to get.

This project from the ARPA2 community wants to deliver such a solution. It has already developed open source software libraries that offer an easy way to distinguish between who is entitled to see something and who isn't. This solution can already be used with all kinds of existing software, because it is compatible with the most popular standards organisations use to keep this data. And you can even work with all kinds of roles and pseudonyms, so unlike most traditional solutions their work isn't completely hardwired to individual people. The latter often leads to people giving their overpowered user credentials to others to quickly get stuff done. In the project, they will now implement it in such a way that all users of the most popular webserver of the moment can take advantage of the power of these libraries. This will help developers outsource one of their headache tasks to a simple and trustworthy open source server component, written by specialists with a focus on security, auditability and standards support. This will in turn simplify applications, will reduce their cost and improve their performance. And of course the small codebase will be significantly easier to analyse in terms of security.

Technical description

In most of our daily interactions with a remote server we depend on the application running on the server to properly authenticate the user within the browser session, and to manage who can do what. However, if we want to enforce stronger guarantees with regards to restricted resources and tasks, our options are much more limited. This project from the ARPA2 community wants to move the state of the art in access control forward by combining the extensible SASL standard with a well-defined generic ACL mechanism that also allows for pseudonimity. The project will produce a self-contained library and two modules for a popular web server (NGINX) that use the new library. With the NGINX HTTP SASL module a user-agent can authenticate to the web server using any SASL mechanism the server supports. With the NGINX ARPA2 ACL module the web server can determine whether an authenticated user has authorization for the request that he/she sent. I.e. a user makes the request: "DELETE /messages/10" and the server can then decide based on the authenticated user, the action and resource whether this is allowed or not.

Netsend — **Visit https://NLnet.nl/project/arpa2-nginx**

NGI0 PET **AccessControl** **Auditability** **IETF** **Middleware** **SASL** **Webserver**

W i r e g u a r d

The internet is unfortunately not only populated just by kind and careful people. And it wasn't designed to be secure either. This is a dangerous and rather unfortunate combination of circumstances, and one you should take into account when you use the internet. When you go online outside of your house or office, chances are you use whatever wireless network you can find to get online. Mobile internet subscriptions are still expensive, so it is logical people seize every opportunity to connect for free. While this is a daily habit for many millions of people, and often nothing bad happens, it does expose you to serious risks.

This is because your computer doesn't really know a lot about the world. It depend on information it gets from the networks it connects to. If the network happens to cheat, the computer often has very little defenses. If you use an untrustworthy network to connect you to the internet, you move yourself into the middle of enemy territory. While you happily enjoy your free bits it provides as a way to buy money, it has ample opportunity to exploit all kinds of security weaknesses against you. Essentially, if you got away scot free with connecting to an unsafe network, it probably wasn't your security that held anyone back. You were just lucky that no-one serious tried to do anything - this time. So it is recommended to not connect over wifi networks you don't know.

Unless of course you've arranged for a secure tunnel that allows you to teleport your internet traffic across the unsafe local network to the real internet unscathed. The concept is surprisingly simple: individual messages sent through the Internet, called packets, are encrypted using some mechanism, and this encrypted message then substitutes the original one, making all communications sent through the tunnel unreadable to eavesdroppers and unalterable to attackers. Proven cryptography protects the integrity of the traffic flowing through the tunnel. And once the packets reach the other end of the tunnel, they can be unpacked. From that point onwards they may continue their life as normal internet traffic. Travelling the path in reverse path is of course also possible: packets sent from the internet to your computer are protected in exactly the same way.

In anticipation of better technologies that should arrive with the next generation internet, such tunnels are a key technology to guarantee consumer safety. They play a major role in protecting users both from snooping and malicious traffic injection. Sadly, the tools to create these secure tunnels is rather cumbersome (if not plain hard) to work with. This has prevented mass adoption.

WireGuard is a completely new entrant to the field, and it is praised widely by technologists for its very high quality. Its goal is to be the most secure and easiest to use VPN solution available. Wireguard has many attractive traits: it is fast, simple and lean. It can run on embedded interfaces and super computers alike, and is fit for many different circumstances. Wireguard makes it very easy to set up a secure tunnel with modern technologies. It employs formally verified cryptographic constructions and has best in class performance. So you can more safely browse the web without annoying delay, even from potentially unsafe networks.

WireGuard starts from scratch with modern cryptography and best-practice defense-in-depth implementation strategies. It is suitable and easily deployable for both end users and in data centers across the world, and provides an essential core building block for making the Internet safer. Within the project the team will continue the effort to make WireGuard land within the Linux kernel, upgrade some parts of the cryptography inside the Linux kernel because the current options are flawed, and do a comparative analysis of Wireguard protocol implementations on Windows, iOS and Android so quality and reliability can be assured across implementations.

Technical description

WireGuard is a next generation VPN protocol that uses state of the art cryptography. One of the most exciting recent crypto-networking developments, WireGuard aims to drastically simplify secure tunneling. The current state of VPN protocols is not pretty, with popular options, such as IPsec and OpenVPN, being overwhelmingly complex, with large attack surfaces, using mostly cryptographic designs from the 90s. WireGuard presents a new abuse-resistant and high-performance alternative based on modern cryptography, with a focus on implementation and usability simplicity. It uses a 1-RTT handshake, based on NoiseIK, to provide perfect forward secrecy, identity hiding, and resistance to key-compromise impersonation attacks, among other important security properties, as well as high performance transport using ChaCha20Poly1305. A novel IP-binding cookie MAC mechanism is used to prevent against several forms of common denial-of-service attacks, both against the client and server, improving greatly on those of DTLS and IKEv2. Key distribution is handled out-of-band with extremely short Curve25519 points, which can be passed around in the likes of OpenSSH. Discarding the academic layering perfection of IPsec, WireGuard introduces the idea of a "cryptokey routing table", alongside an extremely simple and fully defined timer-state mechanism, to allow for easy and minimal configuration; WireGuard is actually securely deployable in practical settings. In order to rival the performance of IPsec, in addition to cross-platform implementations, WireGuard is implemented inside the Linux kernel, but

unlike IPsec, it is implemented in less than 4,000 lines of code, making the implementation manageably auditable. These features converge to create an open source VPN utility that is exceedingly simple, yet thoroughly modern and secure.

Visit https://NLnet.nl/project/wireguard-scaleup

NGI0 PET **Cross-Implementation** **Cryptography** **Kernel** **Protocol** **Security** **VPN**

The Open Green Web

THE GREEN WEB FOUNDATION

Climate change is the most crucial (and divisive) issue of our time. The world needs to switch from fossil fuels to renewable sources of energy as soon as possible, not to avoid the consequences of a rising worldwide temperature, but to limit the irrevocable damage it will cause to our environment, food and vital infrastructures as much as we can. Among those vital infrastructures are the internet and the world wide web, that billions of people around the world everyday and have become the foundation under our economies, societies and even democracies. But what fuels these digital lifelines? When you think of sustainability, grey factories billowing grey smoke may come to mind first, or a highway jam packed with cars. What you may not know is that the world wide web is quite power-hungry itself, taking up a sizable chunk of the electricity produced worldwide and as the technology keeps growing, so does its environmental impact. We need to green the web to make a technology so important to modern life actually durable and sustainable.

So what can a single user do to green the worldwide web? Think of internet users as consumers. Vegetarian food is quickly becoming more popular because consumers buy vegetarian, favor vegetarian restaurants and food manufacturers and restaurant holders hop onto the trend and offer more plant-based products. The same thing is (slowly) happening in other market places and consumer environments. Now think of the world wide web: users are online consumers and hosting companies are product manufacturers. Some hosting companies are seemingly not worried about the environment and 'host grey', fueling their power consumption with fossil-based electricity. Other web hosting services understand the dire consequences of climate change and 'host green', relying on renewable energy sources to keep their sites in the air. How can an environmentally aware user-consumer know that the sites they visit are hosted green and avoid grey hosting services?

Since 2006 the Green Web Foundation has made it clear how the web is powered and over the years has built the largest worldwide dataset which sites are driven by renewable energy. Users can see whether their favorite sites are hosted grey or green with browser plugins, developers can connect their website to the Green Web Feed API and automatically verify sustainable hosted sites and hosting companies can learn how to green their services. So far the Green Web project has resulted in 1.5 billion lookups since 2011.

Seeing which sites are hosted grey or green is one way to help internet users make environmental choices online, but what if they could choose to only visit sustainable websites? The Open Green Web

wants to develop a search engine with ethical filtering, that only show green hosted results drawn from the Green Web Foundation dataset. This green search engine will be based on the fully customizable and open-source search engine Searx, that aggregates results from a variety of search engines and protects your privacy against search tracking. Any Searx instance will be able to use the ethical filter and introduce green search to its users. On top of this, the dataset of 1.5 billion grey and green hosted sites will be opened up to academic research and all the software developed by the Green Web Foundation (like the browser plugins and the API) will be open-sourced.

To green the web, users and hosting services first need to know what parts of it are grey, and how these can become more sustainable. Because search engines for many users are the everyday entry point to the world wide web, adding ethical filtering here can have a significant impact on the sustainability of users' online lives.

Technical description

The world wide web has become a mainstay of our modern society, but it is also responsible for a significant use of natural resources. Over the last ten years, The Green Web Foundation (TGWF) has developed a global database of around 1000 hosters in 62 countries that deliver green hosting to their customers, to help speed a transition away from a fossil fuel powered web. This has resulted in roughly 1.5 billion lookups since 2011 - through its browser based plugins, manual checks on the TGWF website and its API, provided by an open source platform. But what if you want to take things one step further? This project will create the world's first search engine with ethical filtering, that will exclusively show green hosted results. In addition to giving a new choice of search engine to environmentally conscious web users, all the code and data will be open sourced. This creates a reference implementation for wider adoption across industry of search providers, increasing demand and visibility around how we power the web. The project build upon the open source search engine Searx, and will collaborate with the developers of that search tool to make "green" search an optional feature for all installs of Searx.

Stichting The Green Web Foundation — **Visit https://NLnet.nl/project/GreenWebSearch**

NGI0 Discovery **CarbonNeutrality** **Ecology** **EthicalFiltering** **Foundation** **Hosting** **Sustainability**

---

n e u r o p i l

Neuro:pil
Secure Interaction for Things

Search and discovery are some of the most important and essential use cases of the internet. When you are in school and need to give a presentation or write a paper, when you are looking for a job, trying to promote your business or finding relevant commercial or public services you need, most of the time you will turn to the internet and more importantly the search bar in your browser to find answers. Searching information and making sure your name, company or idea can be discovered is crucial for users, but they actually have little control over this. Search engines decide what results you see, how your website can be discovered and what information is logged about your searches. What filters and algorithms are are used remains opaque for users. They can only

follow the rules laid out for them, instead of deciding on their own what, where and how to find the information they are looking for.

Neuropil is a project that wants to turn the tables on online search and discovery: instead of search solutions calling the shots, data owners decide what content is publicly searchable in the first place. They can do this through a new messaging layer that is private and secure by design. Data owners can send cryptographic and unique so-called intent messages that state what specific information can be found where. The access to the actual information or content is also controlled by data owners, for instance to provide either paid or public free content. Instead of a search engine accessing and logging everything on a website to make it searchable, neuropil lets a data owner decide on their own what can be found on their site and how users can access it. This is communicated through a messaging layer that fits easily into standard servers and can connect to users browsers, eliminating the need for search engines or third parties indexing search results entirely.

Technical description

Neuropil is an open-source de-centralized messaging layer that focuses on security and privacy by design. Persons, machines, and applications first have to identify their respective partners and/or content before real information can be sent. The discovery is handled internally and is based on so called "intent messages" that are secured by cryptographic primitives. This project aims to create distributed search engine capabilities based on neuropil, that enable the discovery and sharing of information with significantly higher levels of trust and privacy and with more control over the search content for data owners than today's standard.

As of now large search engines have implemented "crawlers", that constantly visit webpages and categorize their content. The only way to somehow influence the information that is used by search engines is by using a file called „robots.txt“. Other algorithms are only known to the search engine provider. By using a highly standardized "intents" format that protects the real content of users, this model is reversed: data owners define the searchable public content. As an example we seek to implement the neuropil messaging layer with its extended search capabilities into a standard web server to become one actor and to handle and maintain the search index contents of participating data owners. By using the Neuropil messaging layer it is thus possible to build a distributed search engine database that is able to contain and reveal any kind of information in a distributed, concise and privacy preserving manner, without the need for any central search engine provider.

pi-lar GmbH — **Visit https://NLnet.nl/project/Neuropil**

NGI0 Discovery DeviceQuerying Discovery DistributedSearch EmbeddedSystems IoT P2P

Practical Decentralised Search and Discovery

Users have a right to internet access and should be sure that the rights they have offline are also protected online. The internet is not just a technology or a communication medium anymore, as these declarations from the United Nations show, it has become a crucial building block of our

society, economy, democracy and the way we work, live and come together. But in practice, the right to internet access is still a privilege because connectivity definitely is not global (yet). Search and discovery is also unequally distributed: if you want to use one of the major search engines, your computer needs to connect to a server operated by this service provider. But not every region or local community has a stable and uncensored internet connection, so a lot of 'free' advertisement-based online search services are then out of reach.

At their core, communities are built on networking. To be a part of a community, you need to know where you can get your daily groceries, who can fix your car, where to go to if you want to meet new people, how you can connect with your friends. Search and discovery in this sense fulfill basic human needs and should not only have to rely on centralized services, especially when the stable internet connection needed to run these services is not there.

The Serval Project is a extensive effort to create local networks that are extremely fault- and disaster-resistant and can work independent from a centralized infrastructure like the internet. Community members only need their own devices, like their phones, to create and maintain a network together so they can call, send messages etcetera, all without a central service provider or connection to the internet. This project will add search and discovery to these local networks (so-called mesh networks) so community members can more easily find each other and local enterprises can promote their services. This can help strengthen the ties in local, remote and rural communities and improve the quality of service for its users.

Technical description

Internet search and service discovery are invaluable services, but are reliant on an oligopoly of centralised services and service providers, such as the internet search and advertising companies. One problem with this situation, is that global internet connectivity is required to use these services, precisely because of their centralised nature. For remote and vulnerable communities stable, affordable and uncensored internet connectivity may simply not be available. Prior work with mesh technology clearly shows the value of connecting local communities, so that they can call and message one another, even in the absence of connectivity to the outside world. The project will implement a system that allows such isolated networks to also provide search and advertising capabilities, making it easier to find local services, and ensuring that local enterprises can promote their services to members of their communities, without requiring the loss of capital from their communities in the form of advertising costs. The project will then trial this system with a number of pilot communities, in order to learn how to make such a system best serve its purpose.

Visit https://NLnet.nl/project/MeshSearch

NGI0 Discovery DisasterRecovery Mesh Offgrid Resilience ServiceDiscovery

One of the things people enjoy the most about the internet, is that it enables them to talk to others remotely almost without limit. Internet allows anyone to keep closely connected with friends and family, and help their kids solve a math problem while they are at work. People collaborate with their colleagues from the couch of their living room, the cafe where they enjoy lunch or on their cell phone on the bus to the gym. Businesses can easily service their customers where this is most convenient to them, without having to travel themselves. This is so convenient, that some businesses have already moved entirely online. Internet communication has become the nerve center of whole neighbourhoods, where people watch over the possessions of their neighbours while these are away for work or leisure.

Email to this day is among the most popular online communication services and is used by governments, companies and organizations to talk to clients and share files. Even though email was designed without privacy or security in mind. When you send an email, anyone that can gain access to your mail server or the mail server of the recipient can read your mail, from top to bottom. And copy it, for later usage. Or modify it. It is often compared to sending a post card, and of course in many cases there may be little harm in others reading what the weather is like in Athens. But what if you want to use email to send something confidential, something you do not want to share with others? Like a love letter, a political rant or an important contract? And what if you can't actually trust the mail man, for instance because the other party is using a free email service known to search through everything? Or what if you live in a country that has an unhealthy interest in bringing down certain political voices, or are part of a cultural minority that is at risk?

Users should be able to send, store and search their email privately, without giving up ease of use or important email features. Mailpipe does exactly this with a personal webmail client and search engine that users can host themselves. Mailpipe supports user-friendly email encryption and helps users sort through their usually overflowing inboxes with a powerful search engine. This project opens up the Mailpipe email client and search engine to other applications, while the user can still control what data can actually be accessed. Such features can make Mailpipe a more complete user-centric email alternative to existing 'free' email services that leak unnecessary amounts of information and data.

Technical description

Mailpile is an e-mail client and personal e-mail search engine, with a strong focus on user autonomy and privacy. This project, "Mailpile Search Integration", will adapt and enhance Mailpile so other applications can make use of Mailpile's built-in search engine and e-mail store. This requires improving Mailpile in three important ways: First, the project will add fine-grained access control, so the user can control which data is and isn't exposed. Second, enabling remote access will be facilitated, allowing a Mailpile running on a personal device to communicate with applications elsewhere on the network (such as

smartphones, or services in "the cloud"). And finally, the interoperability functions themselves (the APIs) need to be defined (building on existing standards wherever possible), implemented and documented.

Mailpile ehf — **Visit https://NLnet.nl/project/Mailpile**

NGI0 Discovery AccessControl Email Metasearch Search SelfHosted

---

M y n i j

We have come to associate search and discovery of digital content with online search engines. Somewhere on the planet there is an army of all-knowing machines waiting day and night for our inquiries, ready to point us to wherever we need to be - if we ask them nicely. However, this tremendous luxury comes with quite a heavy real-time dependency for internet users: it requires us to have an active connection to the internet whenever we need to find something. As our use of the internet has become more nomadic over the years due to the rise of mobile phones, there are in fact many situations that we find ourselves in where our use of the internet is very restricted or even temporarily cut off. Like when you are on a train, in a busy city centre where the wifi is completely saturated, in a remote area with limited coverage, or when you've ran out of your monthly mobile data plan. Or something more serious, when the network is offline for a longer time due to a cascading network failure or cyberattack.

All of a sudden, we are at a loss. It feels we are thrown back in time. We cannot find anything anymore outside of the files and documents we have stored on our devices. Our on-line search engines are all out of reach and of no use to us. Our many questions will have to wait: there is nothing we can do until we get back online. Such a real-time dependency on a critical resource is not only annoying for users (and sometimes downright disadvantageous when you really need to look up something like a manual or an important reference document). It is also not necessary. Devices like the smartphone in your pocket have such massive storage capacity that you can easily use them to store on-line data of interest, so you can search through it off-line later. This is why Mynij is creating a search engine that can fit on a smartphone, providing accurate results whether on-line or off-line. Not only does this make search more functional and resilient, because the content you search is local and already tailored to your interest, it can even be quicker to use.

Moving search and discovery offline can also give the user more control over their privacy and circumvent forms of online censorship. Because the user stores the information that will be searched themselves, they do not have to worry anymore about some third party tracking every search query (or limiting what sites they can and cannot go through). Search and discovery in this sense works better, is more user-centric and less dependent on third parties and commercial service providers.

Technical description

People feel lost when their connection to the internet is cut. All of a sudden, they cannot search for some reference or quickly look up something online. At the other end, hundreds of millions of servers are 'always on', awaiting the user to come online. Of course, this is neither very resilient nor economic. And it is also not necessary. In the 60s, computers used to occupy a large room. Nowadays, with smartphones, they fit in your hand. A complete copy of the Web (10 PB) already fits on 100 SSDs of 100 TB occupying a volume similar to an original IBM PC. A partial copy of the Web optimised for a single person will thus soon fit on a smartphone.

Mynij believes that Web search will eventually run offline for legal, technical and economic rationale. This is why it is building a general purpose Web search engine that runs offline and fits into a smartphone. It can provide fast results with better accuracy than online search engines. It protects privacy and freedom of expression against recent forms of digital censorship. It reduces the cost of online advertising for small businesses. It brings search algorithms and information presentation under end-user control. And you control its availability: as long as you have a copy and a working device, it can work.

Mynij — **Visit https://NLnet.nl/project/Mynij**

NGI0 Discovery Indexing Mobile Offline Resilience Search

P2Pcollab

Search and discovery are some of the most important and essential use cases of the internet. When you are in school and need to give a presentation or write a paper, when you are looking for a job, trying to promote your business or finding relevant commercial or public services you need, most of the time you will turn to the internet and more importantly the search bar in your browser to find answers. Searching information and making sure your name, company or idea can be discovered is crucial for users, but they actually have little control over this. Search engines decide what results you see, how your website can be discovered and what information is logged about your searches. What filters and algorithms are are used remains opaque for users. They can only follow the rules laid out for them, instead of deciding on their own what, where and how to find the information they are looking for.

Centralizing online search around just a few search engines creates a host of problems, ranging from user privacy and nontransparent filtering to misinformation and fake news. The algorithms of search engines can be misused to show millions of users incorrect and discrediting information and stories about the topic or person they were looking up. This is done to influence elections or to shape the public opinion around specific topics, like refugees and climate change. The reach of these search engines (and the social media networks that are exploited for the same goal) is enormous and once a story goes viral, it is hard if not impossible to take it offline, let alone combat the misinformation with correct reports. At their core, search engines focus on a website's popularity when they filter search results, not information

accuracy. All of this creates a perfect storm for fake news to spread incredibly quickly online.

Because search and discovery for many users is the starting point of their online day, information accuracy is an issue that search solutions should address. One of the ways to establish what information can be trusted and what is false, is to let users assess the data they share first. Peer-to-peer collaboration is a way for internet users to connect and work together directly, without the need for a central authority or in-between layer. Search and discovery in this way can be crowd-sourced, instead of organized by one central party (a search engine) that is more vulnerable to attack and misuse.

Together, peers can publish data, subscribe to other people's messages and documents, recommend and disseminate information and news and tag correct and informed articles and stories, that can then be searched by others. The group filters what data and information should be spread wide and far and what should be forgotten, not a third party (i.e. the search engine provider) that will not give access to its search algorithm to protect their commercial interests. An actually open internet requires transparent, user-centric search and discovery, which is what this project will help to build.

Technical description

This project is working towards creating a more decentralized, privacy-preserving, collaborative internet based on the end-to-end principle where users engage in peer-to-peer collaboration and have full control over their own data, enabling them to collaborate on, publish & subscribe to content in a decentralized way, as well as to discover & disseminate content based on collaborative filtering, while allowing local, offline search of all subscribed & discovered content. The project is researching & developing P2P gossip-based protocols and implementing them as composable libraries and lightweight unikernels with a focus on privacy, security, robustness, and scalability.

Visit https://NLnet.nl/project/P2Pcollab

NGI0 Discovery CollaborativeFiltering Discovery P2P PrivateSearch

---

Free Software Vulnerability Database

NGI ZERO

Software security for many users is a given, an assumption, something you do not and should not have to think about too hard. If you open an app on your phone, install new software on your laptop or boot up your tablet, you assume the software you use is safe, secure and that the developers have done their job right. With the amount of software coming out and the tangled web of inter-dependencies that exist today, this assumption of trust is hard to live up to. Especially since software vulnerabilities are constantly hunted for by malicious parties that want to get into our data and devices for blackmail, theft or on a larger and more dangerous scale, disruption of vital processes like power grids.

Search and discovery of software vulnerabilities is an issue of oversight. There are various databases that record critical risks and issues, but the tools that developers can use to go through these databases tend to focus only on a few sources. Software security should be a collective effort and developers need a

complete view of any insecurities they need to deal with. This project wants to create new free and open source (FOSS) tools that aggregate software vulnerabilities from all possible sources and organize them in a standardized way. This makes secure software development more transparent and ultimately contribute to more solid tools and services for endusers.

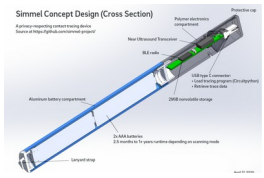
Technical description

"Using Components with Known Vulnerabilities" is one of the OWASP Top 10 Most Critical Web Application Security Risks. Identifying such vulnerable components is currently hindered by data structure and tools that are (1) designed primarily for commercial/proprietary software components and (2) too dependent on the National Vulnerability Database (from US Dept. of Commerce). With the explosion of Free and Open Source Software (FOSS) usage over the last decade we need a new approach in order to efficiently identify security vulnerabilities in FOSS components that are the basis of every modern software system and applications. And that approach should be based on open data and FOSS tools. The goal of this project is create new FOSS tools to aggregate software component vulnerability data from multiple sources, organize that data with a new standard package identifier (Package URL or PURL) and automate the search for FOSS component security vulnerabilities. The expected benefits are to contribute to the improved security of software applications with open tools and data available freely to everyone and to lessen the dependence on a single foreign governmental data source or a few foreign commercial data providers.

Visit https://NLnet.nl/project/vulnerabilitydatabase

NGI0 Discovery **Automation** **Discovery** **FOSS** **Responsiveness** **Software** **Vulnerabilities**

S i m m e l



The SARS-CoV2 pandemic has triggered many different government measures across the world, including unprecedented lock-downs of various degrees of many millions of people. These measures buy time to get ourselves better organised. Because of the immense effects on societies, economies and individuals, such measures can only be temporary in nature, after which societies will have to reboot into a new operational mode as long as there are no effective vaccins.

In such situations of societal crisis, a large array of technologies will be proposed as immediate ad hoc interventions. There is after all significant pressure to just do something, while there is limited oversight whether these technologies would actually help. At the same time, impact on society can be high - when moving fast, things could go horribly wrong too. Move fast, break things, may hold for companies but not for whole societies.

We also need to recognise the significant reserve from the side of citizens that such a situation will

somehow be abused. We have seen such a similar fallout after the 9/11 attacks. An old political paradigm dating back to Machiavelli is after all: "Never waste the opportunity offered by a good crisis.". The same holds for cybercrime: new crises will eventually happen, and the opportunistic ability to improvise is also present in those meaning less well. Organisations and individuals in panic mode tend to not make properly weighted decisions, which can result in serious cybersecurity risks but also the disruption of democratic order.

One of the solutions that has emerged - and provoked a lot of discussion - is "contact tracing". Contact tracing means that you use close distance sensing as an approximation of physical proximity, and thus potential risk of infection. There are a number of ways by which this can be done, and most people think of so called "corona apps". But what if the privacy offered by a mobile phone application are not good enough? What if you have an older type of phone incompatible with the features required? What if you do not even have a phone?

Simmel is a small portable contact scanning device, that is crafted specifically to serve this use case. It can do the same thing or more than a phone can do, but in a much more controlled way. It is open hardware, so one can transparently tweak every aspect of the device from top to bottom - in hardware, in firmware and in software. This meaning that (unlike for instance with third party mobile phones) you do not need permission from a platform provider or operator to tweak any features to satisfy the highest security and privacy criteria.

The Simmel project does not endorse any specific contact tracing design, and its designs are free to be used by any initiative. By purpose it is not compatible with proposed technologies that rely on the constant upload of data to the cloud. The owner of the device should always be in control, because trustworthiness is critical to large scale adoption. Simmel is designed for citizens and for science, not for anything else.

Technical description

Simmel is a platform that enables COVID-19 contact tracing while preserving user privacy. It is a wearable hardware beacon and scanner which can broadcast and record randomized user IDs. Contacts are stored within the wearable device, so you retain full control of your trace history until you choose to share it.

The Simmel design is open source, so you are empowered to audit the code. Furthermore, once the pandemic is over, you are able to recycle, re-use, or securely destroy the device, thanks to the availability of hardware and firmware design source.

The contact tracing algorithm is programmed using CircuitPython, to facilitate ease of code audit and community participation. The Simmel project does not endorse a specific contact tracing platform, but it is inherently not compatible with contact tracing proposals that rely on the constant upload of data to the cloud.

Visit https://NLnet.nl/project/Simmel

NGI0 Discovery **HardwareIsolation** **OpenHardware**

Wireguard Windows client

Amebis — **Visit https://NLnet.nl/project/WG-Windows**

VPN Fund DesktopApp VPN Wireguard

Web Shell

As soon as you sign up with a free email provider or install an operating system, you usually get some cloud storage space. Accessing your data through an online environment has become commonplace, both for business as for individuals. You can share everything from a grocery list with your significant other or even store sensitive documents in the cloud so you can access them from work or on the road. But to be able to have users upload, edit or create files online and pass them around, many cloud services require full access to users data. Like the extensive (and sometimes unreadable) privacy policies you already had to wade through to open your 'free' account, users do not have many options. Either you click yes and the cloud is yours, or you deny the apps all access and are left to your own devices.

The fact that you want to access your data online, does not mean you need to store it in a place where the provider requires you to hand over the keys. Especially when it is unclear who can use the keys to look around in your documents, or analyze sensitive documents simply for the sake of personally profiled advertising, which quite certainly is not what you signed up for. You would want to know where your data is, lock the doors to it yourself yourself and keep the keys somewhere safe (without any intricate key management or cryptographic busywork).

WebShell combines these features of complete data control and user-friendly data access in an environment you are already familiar with: an online version of your desktop. Just like when you start up your laptop or home computer, you start up WebShell and go through your files, open an application to edit something, and switch it off again. Instead of these apps requiring full access to your data that is hosted somewhere in the world, you can self-host your data vault (or choose a hosting service you know and trust) and apps never operate on your files directly. The web desktop securely opens and saves your files and only with your express permission.

Technical description

The WebShell project aims to define and implement a new secure dataflow and the accompanying APIs for allowing users to use their files in Web apps without authorizing the apps to access the user's file storage. At its core, WebShell consists of a container single-page application which can open remote components (primarily apps and file-system adapters) in sandboxed iframes and communicate with them through HTML5 message channels using the defined APIs. WebShell provides for file operations and the required UI (file menus, toolbars, dialogs) to support the familiar file operations (new, open, save, etc.) while apps merely implement serialization and deserialization of an individual file's content, after

the user's explicit request. The project will build a fully-featured WebShell Desktop container, as well as a minimal WebShell container for testing and easy deployment of single apps. In addition, we will integrate a starter set of editor apps for common file types and a starter set of file system adapters, concentrating primarily on self-hosting and non-commercial web storage solutions like remotestorage.io and Solid storage.

Ljudmila Art And Science Laboratory — **Visit https://NLnet.nl/project/Webshell**

NGI0 PET **Containerisation** **Self-Hosted** **UX** **WebApp**

EteSync - iOS application

EteSync

People and organisations use both free and paid online services to manage their private address books, calendars and tasks. These services allow them to back up their data and share the same information across different devices - so they can add an appointment or new contact while they are on the mobile phone at the train station, or on the couch at home, and it magically emerges on their desktop calendar. Other tools allow our loved ones to know where we are at any given moment in time. Given how personal and confidential such information is, use of these convenient services can make users vulnerable to all kinds of abuse.

That risk is not necessary. Service providers can perform the core services (sharing and backup) just as well without any knowledge about user data. Given how normal encryption has become elsewhere on the internet, for instance in instant messaging, it is high time that we start applying it to the information we store about the people we meet, the places we go and the things we do. The overarching goal of the open source EteSync project is to enable users to end-to-end encrypt all of their information, and the expected outcome of this project is to make EteSync available for users on the iOS platform (an often requested feature) - thus making it possible for many more users to switch off the unprotected storage and regain their privacy. Similar to the existing version for other platforms, the new application will support integrated sync for contacts, calendars and tasks, including seamless integration with the native OS as if it was a normal DAV account. This allows users to safely store calendar events, tasks, personal notes and location data ("find my phone") without having to be a computer science wizard.

Technical description

EteSync is an open source, end-to-end encrypted, and privacy respecting sync solution for contacts, calendars and tasks with more data types planned for the future. It's currently supported on Android, the desktop (using a DAV adapter layer) where it seamlessly integrates with existing apps, and on the web for easy access from everywhere.

Many people are well aware of the importance of end-to-end encryption. This is evident by the increasing popularity of end-to-end encrypted messaging applications. However, in today's cloud-based world, there is much more (as important!) information that is just left exposed and unencrypted, without people even realising. Calendar events, tasks, personal notes and location data ("find my phone") are a few such

examples. This is why the overarching goal of EteSync is to enable users to end-to-end encrypt all of their data.

The purpose of this project is to create an EteSync iOS client which will seamlessly integrate with rest of the system and let the many currently uncatered for iOS users securely sync their data.

Visit https://NLnet.nl/project/Etesync-iOS

NGI0 PET **Calendaring** **ClientSideEncryption** **Cryptography** **DAV** **E2EE** **MobileApp** **Sync**

Virtualizing device firmware

The impact of cybercrime is increasing and the attacks on individuals, businesses and crucial infrastructure are becoming more advanced and creative. At the same time we use more 'smart' devices in our homes, offices and streets that are connected to the internet while lacking fundamental security. A camera connected to the internet is not just a camera you can control from your phone, it is also a device that, without certain protection measures, can be manipulated to attack specific servers, trying to take down specific servers which can be immensely harmful, let alone dangerous when crucial infrastructures are the target. To bring the pervasive insecurities of the internet of things closer to home, how about a company selling smart home software that uses the same access details for every house, which can simply open the 'smart' front door lock of every user?

As the internet of things grows and connected devices become cheaper and more commonplace, we need to fix vulnerabilities and close back doors as fast as possible. That means developers should learn how to think like a cybercriminal: how can my device be abused, what creative workaround can grant you access that I should fix? One of the ways to do this is to carefully monitor how a device is actually attacked. This project creates technology that can simulate how basic internet of things devices work and how malicious software will try to abuse it to attack servers. Better understanding one of the many security and privacy threats that plague the internet of things is a step forward in ensuring our devices work for us, instead of against us.

Technical description

Recent targets of attacks on infrastructure did not come from powerful computers, but instead from consumer electronics devices. The most widely known example of this is the Mirai botnet, where consumer grade IP cameras were infected, added to a botnet and then used in wide scale attacks in a rather devious way: the original functionality of the device was left untouched, meaning that users either didn’t notice that their device had been taken over, or weren’t bothered by it. This projects aims to provide a way to virtualise such an IoT device and integrate it with an existing honeypot framework to see how the malware is inserted and how botnets operate. The goal is to extract a firmware from an existing device and use that as the base for the virtualisation. The same setup can also be used to

systematically check for undocumented behaviour of firmware.

Visit https://NLnet.nl/project/virtualfirmware

NGI0 PET **EmbeddedSystems** **Emulation** **Firmware** **Honeypot** **Testing** **Virtualisation**

Suhosin-NG

When you think of programming, you probably do not think of the websites and online services you use and login to everyday. Still, to make a social networking site like Facebook or a popular web content management tool like Wordpress work, users need to interact with a server that in turn should correctly access databases that hold the information needed. That is what the PHP web programming language can do and currently does for the millions of websites that use tools like Wordpress and Wikimedia.

The fact that PHP is a popular web programming language does not mean however that it entirely secure, or that it is always used with attention to security. Through the years, many web vulnerabilities have been found and attributed to bad PHP implementations or insecure default settings. Advancing the state of art in a massively used web programming language is of course non-trivial: if we want to trust some of the most visited websites and services, we should be sure that the technical backend is built securely. Suhosin is a continuous effort to update and secure new versions of PHP and guarantee that implementations leave no loose ends. Suhosin NG, which stands for Next Generation, aims to connect the work already done with a new project created from scratch to best protect PHP 7, the latest version of the web programming language.

Technical description

The PHP programming language was invented by Danish programmer Rasmus Lerdorf in 1994. The language is actively used by millions of websites through popular tools such as WordPress, Owncloud and Wikimedia. Suhosin-NG (next generation) will significantly improve the security of web applications running with PHP 7, and help thwart popular web attack vectors aimed at PHP based websites. Already existing ideas from the Suhosin project for PHP 5 will be gathered in addition to implementing a number of new ideas to improve the overall security stature of PHP 7. This concerns harnessing new features of the language, mitigating security risks in the default configuration and improvements to the runtime behaviour. In practical terms the project will implement these by extending the PHP extension Snuffleupagus, that already provides a good basis for hardening PHP 7. The project's goal is to provide software and documentation for setting up a PHP 7 environment in the most secure way possible.

SektionEins GmbH — **Visit https://NLnet.nl/project/Suhosin-NG**

NGI0 PET **Encryption** **FunctionIsolation** **Hosting** **Library** **Sanitisation**

Ricochet Refresh

When you get up in the morning, and read a fine piece of investigative news about a financial scandal, you don' t really stop to think much about how news is produced and what the human cost of its production is. Every year, dozens of journalists around the world get killed, because of what they write and who they talk to. Even in democratic countries, people can run the risk of intimidation and retribution. If you happen to be a courageous journalist writing about corruption, gangs or some other social wrong, protecting your sources is more than a matter of principle - it can be a matter of life and death for all parties concerned.

Journalists and other vulnerable groups like civil society groups as well as minorities are starting to understand they need to forsake some of the comforts of modern connectivity, in order to avoid danger to their lives and the live of others. If they use commodity internet communication tools, they will likely put themselves at significant risk. This danger lies not just in leaking the content of what they write and what other people send to them, but more so in the ability to observe who interacts with whom, when, and where they are in the real world while they meet on the internet. if you want to be reachable across the internet, you have to constantly let the communication provider follow you wherever you go. With the help of AI and other technologies much can be derived from 'hidden data' you may not have been aware of until now. Next time you use the wifi in the public library while waiting for your informer, who knows who will be sitting behind you?

Ricochet Refresh is a revisit of metadata-free communication software that people like journalists and whistleblowers can use to communicate completely anonymously. Ricochet does not reveal users' IP addresses, encrypts all message content, does not require you to sign in somewhere (and leave personal data) and any contact data is stored only on your device, making tracking of your calls and your contacts practically impossible. To do this, Ricochet uses the Tor anonymity network, which is bound for an upgrade that will make Ricochet not work as it wants to anymore. Ricochet Refresh updates the messaging software to keep it compatible with Tor and plans to implement a smartphone and document transfer feature requested by journalists and activists that use the safe communication solution. Through this project journalists, activists and whistleblowers can continue to disclose information safely and freely and keep the public informed.

Technical description

Ricochet Refresh, is a metadataless messenger for PCs (Windows, macOS, Unix) that provides anonymity as well as security. By using Tor, it allows people at risk making public interest disclosures to communicate in chat sessions with anonymity to journalists, members of parliament, regulators

protecting the environment, financial malfeasance investigators and others who have the power in society to act as corrective mechanisms to serious wrongdoing. This project will update Ricochet, reduce known security risks, and ensure continued compatibility with Tor's onion services protocol. The possibility of anonymous communication is important for everyone, but particularly vital for those who risk reprisal in their workplace or other institutions to be able to speak up. Through anonymity, Ricochet Refresh allows the focus to be on the disclosure, not on the source or whistleblower. Thus, the project provides a tool in support of evidence-based reporting in the public interest by creating a safe on-going channel for the journalist to conduct verification as the story develops.

Blueprint for Free Speech — **Visit https://NLnet.nl/project/Rico**

NGI0 PET **Anonymity** **Foundation** **InstantMessaging** **NGO** **OnionRouting** **P2P** **Whistleblowing**

Nitrokey

Email was designed without privacy or security in mind, which is amazing for such a popular service. When you send an email, anyone that can gain access to your mail server or the mail server of the recipient can read your mail, from top to bottom. And copy it, for later usage. Computer specialists have been protecting their email with encryption for decades. This is the equivalent of putting your message very carefully in the blender, pressing the button before anyone else has read your mesage, shredding it up and sending a packet of shreds over to the other end. The amazing thing about cryptography is that you can magically (or rather mathematically) make it possible for your secret love - and not anyone else - to recreate the message from the shreds, and know it was you - and not anyone else - that sent it. For the rest of the world, the message would be meaningless garble pretty much forever. However, the solution they came up with is not easy for normal people to work with. This means that most people are probably not even aware that it is possible to protect the contents of their email with cryptography.

Good cryptography begins and ends with key management: to encrypt and decrypt data you create a secret key you store safely and a public key you share with others to communicate privately and securely. Users that need to be sure no one can access their files, like journalists, activists and whistleblowers, would want to make sure their secret key are somewhere else then on their one device. They would need a failsafe, a backup plan, like a separate and protected storage device. That is what the Nitrokey can be: an open-source USB key that can store secret keys to protect emails and encrypted files and store important sensitive data. The key is PIN-protected and resistant to brute force and hardware attacks. Next to storing secret cryptographic key, Nitrokey allows its users to better protect their passwords by creating one time passwords or providing two-factor-authentication.

To make this arsenal of privacy and security measures work for as many users and services as possible, this project wants to make Nitrokey better protect your web services through encryption, which you can use for example to secure your webmail. Through adding this feature any web application can now store the private keys used to encrypt your communication and files safely on a separate, secure device.

Technical description

Nitrokey is an open source hardware USB key for data encryption and two-factor authentication with FIDO. While FIDO is supported by web browsers, using Nitrokey as a secure key store for email and (arbitrary) data encryption requires a native software. Therefore email encryption in webmail isn't possible with Nitrokey. At the same time strong end-to-end encryption in web applications all share the same challenge: To store users' private keys securely and conveniently. Therefore secure end-to-end encryption usually requires native software too (e.g. instant messenger app) or - less secure - store the user keys password-encrypted on servers. Nitrokey aims to solve these issues by developing a way to use Nitrokey with web applications. To avoid the necessity of device driver, browser add-on or separate software this project is going to utilize the FIDO (CTAP) protocol. As a result the solution will work with any modern browser (which all support WebAuthn), on any operating system even on Android. This will give any web application the option to store private keys on ones own Nitrokey devices.

Nitrokey — **Visit https://NLnet.nl/project/Nitrokey**

NGI0 PET **Authentication** **Cryptography** **FIDO** **Hardware** **WebAuthn**

M E G A 6 5 P h o n e

Do you completely understand how your computer, laptop or smartphone works? Do you know what happens behind the browser, the text editor, the operating system? Probably not, and that is not a surprise nor is it something to be ashamed of. The development of consumer electronics is like a web that becomes increasingly intricate, where new technologies added continuously without anyone checking how the wires are connected or if there is a risk for short-circuiting. All sorts of vulnerabilities and back doors have crept in software and hardware over the years that even the developers themselves are sometimes unaware of.

Users can either rely technology companies reassuring them that another patch or fix will make everything okay, or be able to switch to a device they know they can trust and that can be thoroughly and transparently verified for security and privacy. To have that option, Mega65 goes back to the early days of personal computing and has developed a completely open-source 8-bit computer that users can modify as they see fit, with full transparency. Now the project aims to develop a portable version of this computer as a smartphone that follows the same approach: technology can only be truly secure if it is and stays simple. The smartphone is open-source (which is quite hard in fact), connects to the internet like any other smartphone and instead of needing a lifeline everyday, has a battery life of some 1000 hours in standby. Users can now have a phone they can truly trust to protect their privacy and be transparently, completely secure.

Technical description

Much of the insecurity and lack of privacy is the simple result of how complex computers, the internet

and all of the protocols and technologies that they include. It seems that the majority of proposals to fix this solution consist of adding something to this complicated mess. While this has helped to reduce the symptoms of the problem, by adding complexity it has actually made the problem worse. There are simply too many places for insecurities and privacy violating software to hide in modern complex systems. Even the hardware itself is not immune, with problems like SPECTRE, MELTDOWN and vulnerabilities in the management processors of modern computers and phones showing that even the processors we use today carry significant risks due to their complexity. This project takes a contrarian approach of seeing just how simple a system can be make, that would still be useful for a core set of functionality. The project takes inspiration from the simple and effective computers of the 1980s: it explores how to retain their simplicity and transparency, and combine them with modern improvements in security and capability. The goal is to allow even a single determined person to completely verify that a device has not been compromised, and that there are no unwanted listening ears when performing privacy sensitive tasks. The project will advance its current proof-of-concept to a functioning hardware and software system that can demonstrate profoundly improved security and privacy, and in a way that allows a determined user to verify that the device is still truly under their exclusive control and serving them alone.

Moulds Team AG — **Visit https://NLnet.nl/project/Mega65**

NGI0 PET **Dumbphone** **HardwareIsolation** **MobileOS** **ModularHardware** **OpenHardware** **TrustedComputingBase**

V i t a

On the internet, every computer by design gets a unique number - a so called internet protocol address (or for short IP address). This address is used to send information from your computer to the other computer you want to communicate with, and of course back. Unlike a traditional radio, you often need to send messages to receive messages on the internet. Computers are a great engineering achievement but they are certainly not magic, and thus they need to be able to somehow find each other. The IP address makes this possible. Unfortunately, the fact that every computer has a unique number opens up the possibility of abuse by dishonest actors. Because even though it is none of their business, breaking privacy is a profitable business. If they link what you do on the left side of the internet to what you do on the right side of the internet, they can create a profile and sell this to the highest bidder - with any bad luck to people that want to use it for nefarious purposes.

Misuse of IP addresses shows just one of the ways in which the internet protocol and other important networking technologies are designed to connect, to extend, but not always to secure the traffic that is sent over it. The pioneers of the internet simply could not foresee how massive and crucial their technology would become to modern society. This project aims to add security to the core internet protocol by encrypting and codifying the information it transports so users can confidentially be online. IPsec, which stands for internet protocol security, is an older effort to protect users privacy and security

on the internet and Vita aims to update this work and make it ready for deployment at your local network operator. Fixing and securing fundamental internet technologies is a worthwhile effort for the billions of users that live and work online as we speak, but can only make a difference for people if it is actually a part of the current internet. This project can help make that a reality and raise the bar for online privacy and security.

Technical description

When the IP protocol was designed, its original authors did not add adequate security features. In 1994 the first official RFC concerning an end-to-end encrypted variant of IP called IPSEC was published after a number of years of standardisation work in the IETF. Almost a quarter of a century later, there is still a very limited set of implementations of the protocol. IPSEC is perceived by many as hard to deploy, which creates a chicken and egg situation in driving adoption. Vita is a fresh new implementation of IPSEC based on Snabb Switch, a high performance open source packet networking toolkit. The goal of Vita is to make it very easy to use IPSec on commodity hardware, and to produce a fast and compliant clean room implementation. Vita previously received funding from the Internet Hardening Fund. This project will move the deployability of Vita forward, and among others will produce a number of drivers for interfacing with e.g. high speed interfaces such as the Linux kernel. It limited size and use of an existing packet networking toolkit means it can be easily audited.

Interstellar Ventures — **Visit https://NLnet.nl/project/vita-cloud**

NGI0 PET **Compatibility** **Confidentiality** **Deployment** **IPSEC** **VPN**

---

d h c p c a n o n

Privacy is a matter of control. When you want to protect your privacy, it does not mean you never tell anyone anything, it means you want to be in control of who you share your personal information with. On the internet a lot of control is taken away from you. The technology that lets you connect to networks all around the world and find information anywhere it is stored is built around identification, both of its users and the virtual places they visit. Unfortunately, many crucial networking standards and protocols were not designed with user privacy in mind, let alone giving them any sense of control over who can see what they do online. This vacuum has been filled with all sorts of tracking and tracing schemes that can make detailed profiles of people, which can then be (mis)used for commercial or even criminal gain.

One of the protocols that is both a crucial part of how the internet works and also a potential privacy hazard is DHCP, or Dynamic Host Configuration Protocol. This protocol, like the name states, dynamically distributes important identifiers like internet procotol addresses (IP addresses) when users connect to a particular network. These identifiers can be leaked and then used to identify and track the device of a specific user. dhcpcanon gives back users some control over their online privacy by minimizing any personal information that can be disclosed through DHCP. The project helps to

implement an existing and proven technical standard on DHCP privacy protection into current networks. This way the internet community can take practical steps to make our online life more private and move forward to a more privacy-friendly technology.

Technical description

When your computer enters a new network as a guest, it will need to receive information to be able to send and receive packets. The internet standard responsible for this is called Dynamic Host Configuration Protocol (DHCP). Traditional DHCP and DHCPv6 can potentially leak information which can be abused to uniquely identify a certain device - and thus track a user. dhcpcanon is a DHCP client implementation that implements the technical standard RFC7844, DHCP Anonymity Profiles. The new standard provides guidelines for minimizing information disclosure via DHCP. This project will produce DHCP clients implementing the Anonymity Profiles for restricted devices as microcontrollers and easy integration with network management tools.

Visit https://NLnet.nl/project/dhcpcanon

NGI0 PET | Configuration | DHCP | Deanonymisation | EmbeddedSystems | IETF | Middleware

---

S p e c t r u m

How can you understand and trust a complex system, like the operating system managing the hardware and software on your computer? You can make the complexity simpler by cutting it up into parts, compartmentalizing what does what, where information is stored, which processes talk to each other. This way users can be sure their system only does what it is supposed to do and know precisely what goes in and what comes out. This can be done through virtual machines, which are isolated simulations of operating systems or programs on a computer. Simply put, you create virtual rooms where only one thing happens and only you have the keys to each door. This can give users complete control over what happens on their computer and ensures that if some malicious software finds a way in, it cannot get to the other rooms. This can be very important if your device contains sensitive information, if some ill-meaning third party tries to listen in, or when the device is part of some crucial infrastructure and is targeted for attacks.

Security by isolation sounds simple enough, but in actuality requires a lot of work and maintenance. Operating systems that can compartmentalize programs and processes are very hardware-specific and the virtual machines they run require regular and complicated upkeep. The Spectrum operating system takes a different and simpler approach: all data on the system is stored in one place and applications that need access to that data are isolated and specifically told what information they can and cannot

access, even within the same application. For example, when you want your word processor to access certain files when you are working and other documents when you are at home, you can create two versions or simulations (called instances in Spectrum) with specific access rights. Users can keep a clear overview of their system and applications, as well as the various instances they create, by simply writing all this down in a configuration text. A system called Nix takes this text and creates all the software that the user has written down. Each program can be updated separately, without worries that other parts will break or become incompatible. Users always have a clear overview on what is happening on their computer, instead of getting lost in a maze of virtual rooms. Security by isolation becomes more manageable and transparent, making it accessible for a larger audience.

Technical description

Spectrum is an implementation of a security through compartmentalization based operating system, built on top of the Linux kernel. Unlike other such implementations, user data and application state will be managed centrally, while remaining isolated, meaning that the system can be backed up and managed as a whole, rather than mixed up in several dozen virtual machines. The host system and isolated environments will all be managed declaratively and reproducibly using Nix, the purely functional package manager. This will save the user the burden of maintaining many different virtual computers, allowing finer-grained resource access controls and making it possible to verify the software running across all environments. The Linux base, and a variety of isolation technologies from containers to virtual machines, will bring security through compartmentalization to a much wider range of hardware than previous implementations, and therefore make it accessible to many more people.

Visit https://NLnet.nl/project/Spectrum

NGI0 PET **Desktop** **LiveDistro** **OperatingSystem** **Reproducibility** **TaskIsolation** **TrustedComputingBase**

Accessible security

INSURGO

Most users rely on antivirus programs to keep their system and important data safe and private. Visited sites, downloaded files, email coming in and out, everything should pass through a digital border control that keeps malware and spyware out. Perform a complete system scan every other month and most users will be reassured: I am safe. The truth is that there is more than one way into your system and not every backdoor is properly protected. Attackers can also target the BIOS (Basic Input/Output System) program that every computer has to boot up and load the operating system. The BIOS is the first process to run when you power on your computer and is usually not scanned by any antivirus or security software you have installed. Accessing the BIOS and installing malicious software on such a fundamental level gives attackers far-reaching control over a system (which is why it is used for ransomware) and the user usually does not even realize it. And updating their BIOS probably is not something they do (if they are even aware of it at all).

Fortunately, there are plenty of open-source tools developed over the years that can completely secure

your system, down from the hardware and the BIOS up to the software you use. Unfortunately, the barrier to entry of many tools is probably too high for most users, who will not now where to begin and get lost in a maze of technical details. Which program is better than the rest, how can I make tool A work with framework B? And how will all of this affect my system, can I still use my computer the way I am used to?

Security should not be a black box. Instead, users should be able to choose from plug & play solutions that work together nicely and cover most if not all exits in their systems. Or they should have a one-stop-shop solution, a big green button they can press for total security. This project aims to update, optimize and interconnect existing open-source security solutions. The end goal is to improve the security of both technical and non-technical users. In a deeply connected world, cybersecurity should be as democratic as possible so we can be sure we can actually trust our online devices.

Technical description

The "Accessible security" project's initiative was sparked by the need for usable security made available to the average citizen. Several projects are contributing a part of this bigger puzzle: QubesOS, coreboot, Heads, me_cleaner, Whonix and others. Yet the average person does not have the sophistication to integrate these software projects. With some effort we can add some missing parts, help the effected projects usability, and facilitate access to cutting-edge developments, currently only usable by developers and more sophisticated users. Bringing these projects together will reduce the amount of expertise and effort required to benefit from these projects.

Insurgo, Technologies Libres / Open Technologies — **Visit https://NLnet.nl/project/AccessibleSecurity**

NGI0 PET **BIOS** **Bootmanager** **DesktopOS** **Firmware** **Hardware** **OnionRouting** **TaskIsolation**

S O L I D D a t a W o r k e r s

In the 'real world', you instinctively know what information you should keep behind locked doors and what is safe to share. Your bank statements are stored in a folder somewhere in the attic instead of leaving them laying around on your kitchen table. You do not tell random people on the street what your phone number is, or where your children go to school. In the virtual world, this type of common sense can work differently. Users are quicker to trust service providers to keep their personal data safe from theft and prying eyes, and do not always see the dangers of storing passwords in an online text file, or sharing sensitive financial documents via email. The dangers are unmistakably there, but until someone close to you suffers the consequences of a hack or a privacy breach, the risks of online data storage are vague and its convenience is too tempting to

pass up. People are accustomed to easy, accessible and convenient online tools and services. More private and secure open-source alternatives should not exclude users because of an overly technical setup or incompatibility with existing proprietary solutions.

Solid (or Social Linked Data) is a new approach to protecting personal data initiated by Tim Berners-Lee, the inventor of the world wide web and developed in collaboration with the Massachusetts Institute of Technology (MIT). The project aims to give users back full control over their personal data, which they can store in personal online data stores (or pods) and then give applications that run on the Solid platform access rights as they see fit. Users always retain ownership over their data, decide for themselves where it is stored and can change the permissions of any application that can access the data. Eventually the Solid ecosystem should offer decentralized and user-centric alternatives to centralized social media like Facebook, Twitter, LinkedIn etcetera.

Convincing people to switch to Solid will take more than just telling them privacy horror stories. You cannot (and should not) scare someone into using your product, no matter how good your intentions might be. The alternative should be as good or even better than the original and switching should be easy and painless. That is what Solid Data Workers will provide: a toolkit that can bridge the gap between the online services you use now (email and calendars for example) and the Solid platform, keep data synchronized and allow you to import existing data to work flawlessly with the Solid technology and approach to data. Privacy is far from dead, but people usually lack the tools or technical knowledge to protect their personal data online. New and promising privacy-friendly platforms like Solid should be as inclusive as possible to actually make a difference and change the status quo of online personal data. This project can help make that change.

Technical description

Solid Data Workers is a toolkit to leverage the Solid platform (an open source project led byTim Berners-Lee) into a viable, convenient, open and interoperable alternative to privacy-hungry data silos. The aim is to use Solid as a general purpose storage for all of the user's private information, giving them a linked-data meaning to enrich the personal graph and provide a first-class semantic web experience. The project involves a PHP and a NodeJS implementation of the "Data Workers" toolkit to easy the "semantification" of the data collected from external services (SPARQL queries build, metadata retrieval and storage, relationships creation...), some sample software component to import existing data into the semantic graph and keep it synchronized with back-end sources (primarily: emails and calendars), and a proof-of-concept application to showcase the potentials of the semantic web applied to personal linked data. As Solid may be self-hosted or hosted by third-party providers, Solid Data Workers may be attached to any of those instances and to different back-end services.

Visit https://NLnet.nl/project/SOLIDdataworkers

NGI0 PET DataHarvesting Decentralisation Ontology Solid

Privacy is a matter of control. When you want to protect your privacy, it does not mean you never tell anyone anything, it means you want to be in control of who you share your personal information with. On the internet a lot of control is taken away from you. The technology that lets you connect to networks all around the world and find information anywhere it is stored is built around identification, both of its users and the virtual places they visit. Unfortunately, many crucial networking standards and protocols were not designed with user privacy in mind, let alone giving them any sense of control over who can see what they do online. This vacuum has been filled with all sorts of tracking and tracing schemes that can make detailed profiles of people, which can then be (mis)used for commercial or even criminal gain.

Remaking the internet to be secure and private by design is what the ARPA2 project has been doing for several years by using and extending existing security standards and developing identity management solutions for users. Technology developed in the ARPA2 project simplifies and centralizes encryption and integrates state of art authentication and security standards, among other things. These tools help build an internet that "treats its end users as full-blown citizens, and not as milking cows", as the ARPA2 developers explain on their website. Just like other ARPA2 initiatives this project makes existing and proven internet technology more interoperable and usable to better protect users online data and identity on the internet.

Technical description

Some protocols are far better known than others. Everyone will recognise the HTTP protocol we use to transfer web pages. LDAP is not as well known, but it is also a key technology we use on a daily basis - in fact it shapes how most organisations are organised online. LDAP is a proven technology but can be cumbersome to work with, and as a result it has seen little innovation in recent years.

This project develops a number of innovatie middleware components from the ARPA2 project. This includes a privacy enhancing middleware for LDAP (LEAF), which allows to do attribute filtering and selectively transforming of LDAP; SteamWorks, which allows for responsive large scale configuration and trust delegation; and Lillydap, a library that can be used to easily add LDAP to any application. The project also delivers on (broad)er deployability of these building blocks, by providing tools for distropackaging the innovative solutions produced by the project.

Visit https://NLnet.nl/project/LDAPmiddleware

NGI0 PET AccessControl Attributes Certificates Configuration KeyManagement LDAP Middleware Pseudonimity

Reusing passwords is a known security risk. But remembering unique and strong passwords, full of numbers and symbols, is practically impossible with the amount of logins people have. Instead of trying to memorize them all or, even worse, write them down somewhere close to the computer, password managers can also do the trick. Of course this password vault then becomes the new point of failure, especially if the manager saves and syncs the passwords online to a server somewhere. Of course any online password manager would not be worth its salt if they can be easily cracked. But a well-prepared attacker would only need to get into a server once to grab as many passwords as they can, possibly compromising countless accounts in the process.

Users can only truly trust password stores if they are practically unbreakable. This project combines two technologies that make it practically impossible for attackers to access your stored passwords, both online and offline. Sphinx adds end-to-end encryption to password storage: when you open your vault with your master password and copy or select the password you need to login to a specific service, both your master key and your other passwords are encoded and unusable for any spying third party the whole time. On top of that, the new protocol Opaque makes phishing impossible when authenticating (in other words, proving that you are who you say you are) to the server. Combining these two technologies makes password storage theoretically secure, meaning users can actually rely that their passwords and the massive amounts of personal data they give access to are as safe as possible. This project aims to further develop Sphinx and create a Sphinx server, as well as propagate the Opaque protocol to massively used free software, programming languages and two very popular webservers. Implementing and proving the worth of these password protection measures can help make password managers more attractive and trustworthy to novel users, which will ultimately better protect their privacy and security online and offline.

Technical description

Passwords are probably the most common way to remotely use private services, which makes them a major liability - humans on average find it very hard to memorize strong passwords. Luckily, passwords - or more particularly tools to work with passwords more safely - are evolving as well. SPHINX is a novel approach to password storage that is information theoretically secure. And unlike most online password managers, the user does not even have to trust the server. OPAQUE is a novel protocol that can be used to eliminate phishing as an attack vector when authenticating to servers. The combination of SPHINX and OPAQUE provides some very strong guarantees while still allowing users to only need to remember one or just a few passwords. This project will develop a SPHINX server in a safe, compiled language, with ample tests. It will also further develop and refine a protocol above SPHINX, handling creation, deletion, backup and changing of data. In addition it will add the OPAQUE protocol to various free software ecosystems such as PHP, java, nodejs, ruby, golang, erlang and rust, as well as to the two most used webservers: nginx and apache2.

Visit https://NLnet.nl/project/OpaqueSphinxServer

NGI0 PET Authentication Cryptography E2EE Password Passwords

Robur privacy-enhanced DNS resolver and DHCP server

robur

How can you understand and trust a complex system, like the operating system managing the hardware and software on your computer? You can make the complexity (as well as the security) of a system more transparent by cutting it up into parts, compartmentalizing what does what, where information is stored, which processes talk to each other. This way users can be sure their system only does what it is supposed to do and know precisely what goes in and what comes out. This can be done through virtual machines, which are isolated simulations of operating systems or programs on a computer. Simply put, you create virtual rooms where only one thing happens and only you have the keys to each door. This can give users complete control over what happens on their computer and ensures that if some malicious software finds a way in, it cannot get to the other rooms. This can be very important if your device contains sensitive information, if some ill-meaning third party tries to listen in, or when the device is part of some crucial infrastructure and is targeted for attacks.

Security by isolation can be important, for example, to keep a server or host device safe that provides crucial network services. To get anywhere on the internet, you need to have or be assigned an Internet Protocol (IP) address (which is handled by the Dynamic Host Protocol), and find out what IP address belongs to the website name you type into your browser bar (what the Domain Name System protocol helps to do, among other things through resolvers). A DHCP server and DNS resolver should be well-protected to keep your web traffic safe. This project wants to make a DHCP server and DNS resolver in an isolated virtual machine. MirageOS is an operating system that can create unikernels, isolated virtual machines that run operating systems with a single purpose. Making sure that the system running your DHCP server and DNS resolver can only do those two things limits the possibilities for an attacker to get in. Simply put, you can protect or close a back door in your system, or you can make sure that there is no back door all together. And to make extra sure that every client can rely on the server to protect its personal data, the DHCP server and DNS resolver will minimize the data it stores and encrypt all communication as much as possible. This project can show in practice how a unikernel can make your DHCP server and DNS resolver more secure and protected against anyone trying to listen in on where you go online and who you communicate with.

Technical description

DHCP and DNS are fundamental Internet protocols, DHCP is used for dynamic IP address configuration in a local network, DNS for resolving hostnames to IP addresses. In this project, we develop a robust DHCP server and DNS resolver as a MirageOS unikernel. MirageOS unikernels are self-contained virtual machine images which are composed of the required OCaml libraries, leading to a binary with a minimal trusted code base, and thus minimized attack surface. The choice of the memory-safe, functional, and statically typed language OCaml avoids common attack vectors, such as buffer overflows and double frees. MirageOS unikernels can be deployed on various hypervisors (Xen, KVM, BHyve), microkernels (Genode, Muen), or as Unix binary (also with seccomp rules that allow only 10 system calls) on x86-64 and arm64. Several DHCP and DNS privacy extensions, extensive testing, and documentation is worked on to allow everyone to use it on their home router or in the data center. Migration of existing configuration (e.g. dnsmasq) to Robur DNS resolver and DHCP server will be provided as well.

robur project — **Visit https://NLnet.nl/project/Robur**

NGI0 PET DHCP DNS DNS-over-TLS TypedLanguage Unikernel

P G P 4 c i v i C R M

Email was designed without privacy or security in mind, which is amazing for such a popular service. When you send an email, anyone that can gain access to your mail server or the mail server of the recipient can read your mail, from top to bottom. And copy it, for later usage. Computer specialists have been protecting their email with encryption for decades. This is the equivalent of putting your message very carefully in the blender, pressing the button before anyone else has read your mesage, shredding it up and sending a packet of shreds over to the other end. The amazing thing about cryptography is that you can magically (or rather mathematically) make it possible for your secret love - and not anyone else - to recreate the message from the shreds, and know it was you - and not anyone else - that sent it. For the rest of the world, the message would be meaningless garble pretty much forever. However, the solution they came up with is not easy for normal people to work with. This means that most people are probably not even aware that it is possible to protect the contents of their email with cryptography.

The encryption issue is especially important for sensitive emails, like the messages and documents we get from all sorts of institutions and civic services. Whether you apply for a new drivers license, buy a house, or need healthcare services, a lot of personal and sensitive information needs to be sent back and forth. That communication should be well-protected, which encryption can do. This project adds plug & play email encryption to an open-source customer relationship management (CRM) system. Civic sector organizations that use this CRM system to keep in touch with their clients can then easily encrypt every email they send their clients. This way we can be sure that the sensitive information we share with public institutions is private and secure.

Technical description

E-mail security and privacy is not just relevant inside organisations or between individuals. A lot of email traffic comes from the institutions we all have to deal with, including some of the most confidential emails we get. And yet there is no way for users to protect their privacy and confidentiality when sending and receiving messages from organisations using such systems. PGP4civiCRM enables automatic PGP encryption/decryption of e-mails on the server side. While the project will provide special integration for the Constituent Relation Management System CiviCRM, the basic functionality can be used also with regular mailservers like postfix. The PGP4civiCRM core will basically be a milter, that listens for input messages, then looks up PGP keys from configurable sources (local key rings, LDAP) and then, based on a local, configurable, policy, encrypts/decrypts messages (or leaves them untouched) before passing them on. This way system administrators can with tiny effort provide transparent encryption support for all their mail users. Especially for CiviCRM the project will create an extension that allows easy web-based configuration of the relevant pieces and displaying of encrypted, received e-mails using OpenPGP.js.

Visit https://NLnet.nl/project/PGP4civiCRM

NGI0 PET **CRM** **ClientSideEncryption** **E2EE** **Email** **GPDR** **KeyManagement** **Milter** **OpenPGP**

B i t m a s k

The internet is unfortunately not only populated just by kind and careful people. And it wasn't designed to be secure either. This is a dangerous and rather unfortunate combination of circumstances, and one you should take into account when you use the internet. When you go online outside of your house or office, chances are you use whatever wireless network you can find to get online. Mobile internet subscriptions are still expensive, so it is logical people seize every opportunity to connect for free. While this is a daily habit for many millions of people, and often nothing bad happens, it does expose you to serious risks.

This is because your computer doesn't really know a lot about the world. It depend on information it gets from the networks it connects to. If the network happens to cheat, the computer often has very little defenses. If you use an untrustworthy network to connect you to the internet, you move yourself into the middle of enemy territory. While you happily enjoy your free bits it provides as a way to buy money, it has ample opportunity to exploit all kinds of security weaknesses against you. Essentially, if you got away scot free with connecting to an unsafe network, it probably wasn't your security that held anyone back. You were just lucky that no-one serious tried to do anything - this time. So it is recommended to not connect over wifi networks you don't know.

Unless of course you've arranged for a secure tunnel that allows you to teleport your internet traffic across the unsafe local network to the real internet unscathed. The concept is surprisingly simple: individual messages sent through the Internet, called packets, are encrypted using some mechanism, and this encrypted message then substitutes the original one, making all communications sent through the tunnel unreadable to eavesdroppers and unalterable to attackers. Proven cryptography protects the integrity of the traffic flowing through the tunnel. And once the packets reach the other end of the tunnel, they can be unpacked. From that point onwards they may continue their life as normal internet traffic. Travelling the path in reverse path is of course also possible: packets sent from the internet to your computer are protected in exactly the same way.

In anticipation of better technologies that should arrive with the next generation internet, such tunnels (which can be created with a virtual private network or VPN) are a key technology to guarantee consumer safety. They play a major role in protecting users both from snooping and malicious traffic injection. Sadly, the tools to create these secure tunnels is rather cumbersome (if not plain hard) to work with. This has prevented mass adoption.

Bitmask wants to make encrypted communication accessible and easy for users to use. The open source application offers email encryption which handles all the complicated cryptography on its own and a virtual private network that takes extra security measures to make sure no personal information is leaked. This project aims to make these privacy and trust enhancing technologies more accessible and 'plug & play' by fitting BitmaskVPN into commonly available routers. Any VPN provider will be able to offer their clients a router with VPN built in: all they have to do is install it in their home and flip the switch. This way privacy and trust enhancing technology can actually become a part of the everyday devices internet users are accustomed to, making their online experience more private and secure without any complex technological setup or hassle.

Technical description

Bitmask is a Desktop and Android client designed to achieve a zero-configuration end-user experience for setting up a VPN that connects to a given set of providers - those that follow the LEAP platform specification. To do so, clients rely on providers exposing configuration files on well-known urls, according to their particular setup regarding the available VPN gateways and transports. This project aims at adding low-end routers a new extra platform that users can choose when installing BitmaskVPN. Running VPN software in a commonly available router, with hardware-based user interfaces, will greatly extend the target audience for Bitmask. To achieve this goal, a porting of the BitmaskVPN client will be done in nim, a statically typed language that generates small native and dependency-free executables, allowing the setup of the VPN with the switch of a hardware button. Finally, the resulting port will be packaged for OpenWRT, and build scripts will be made available for providers to offer to their users a ready-to-use flashing image for a selection of routers.

LEAP Encryption Access Project — **Visit https://NLnet.nl/project/bitmask**

NGI0 PET **Configuration** **Deployment** **Firmware** **Router** **ServiceDiscovery** **VPN**

How can you understand and trust a complex system, like the operating system managing the hardware and software on your computer? You can make the complexity simpler by cutting it up into parts, compartmentalizing what does what, where information is stored, which processes talk to each other. This way users can be sure their system only does what it is supposed to do and know precisely what goes in and what comes out. This can be done through virtual machines, which are isolated simulations of operating systems or programs on a computer. Simply put, you create virtual rooms where only one thing happens and only you have the keys to each door. This can give users complete control over what happens on their computer and ensures that if some malicious software finds a way in, it cannot get to the other rooms. This can be very important if your device contains sensitive information, if some ill-meaning third party tries to listen in, or when the device is part of some crucial infrastructure and is targeted for attacks.

The Qubes operating system is a pioneer in creating an isolated yet workable desktop. Users can segment programs and data into separate cubes, based on how trust. The default cubes are 'work', 'personal' and 'untrusted', that are each run in an isolated virtual machine. If you open a phishing email in your 'untrusted' cube and malware manages to make its way into this specific environment, it cannot get to 'personal' or 'work' and therefore cannot compromise that data (or the entire operating system, which is the case with popular operating systems like Windows that have a huge attack surface). Various colors (think green, yellow, red) can be used to indicate what window and program works in what qube.

Security by isolation can and should be a great way to make operating systems more secure by design. Usability is then of course an important issue: a better secured operating system should not be harder to use then a more vulnerable one. This project will pick up and implement existing efforts to make Qubes more transparent and usable. For example, to manage the qubes a user has created, this project will help to feature interfaces that make it easier to keep an overview. Also, existing work to internationalize the documentation that guides users and developers into Qubes will be updated. And to make the various qubes more accessible, users can switch from colored windows to other types of labels.

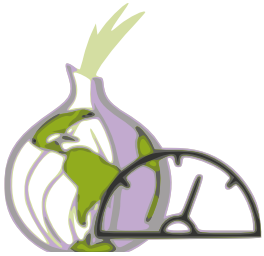
Technical description

Qubes OS is a free and open source operating system uniquely designed to protect the security and privacy of the user. Its architecture is built to enable the user to define different security environments ("qubes") on their computer and visually manage their interaction with each other and the world. This project will improve the usability of Qubes OS by: (1) reviewing and integrating already existing community-created usability improvements, (2) implementing a localization strategy for the OS and its documentation, and (3) creating a holistic approach for improved accessibility.

Qubes OS — **Visit https://NLnet.nl/project/QubesAccessible**

NGI0 PET **UX** **VPN** **i18n**

Accessibility **DesktopOS** **LiveDistro** **Localisation** **TaskIsolation**



On the internet, every computer by design gets a unique number - a so called internet protocol address (or for short IP address). This address is used to send information from your computer to the other computer you want to communicate with, and of course back. Unlike a traditional radio, you often need to send messages to receive messages on the internet. Computers are a great engineering achievement but they are certainly not magic, and thus they need to be able to somehow find each other. The IP address makes this possible. Unfortunately, the fact that every computer has a unique number opens up the possibility of abuse by dishonest actors. Because even though it is none of their business, breaking privacy is a profitable business. If they link what you do on the left side of the internet to what you do on the right side of the internet, they can create a profile and sell this to the highest bidder - with any bad luck to people that want to use it for nefarious purposes.

While work is under way to replace the design of the internet within the Next Generation Internet initiative, there are multiple ways to avoid your IP address being tracked on the current internet. A popular method to attempt to anonymize ones internet presence is to use the Tor network. Tor is a network of millions of computers and users that send messages among each other to confuse someone watching internet traffic. The network that helps to protect the privacy of journalists, activists, whistleblowers and other users is run by volunteer nodes that bounce communication details around to prevent tracking and snooping.

Like any network, Tor needs to protect itself from attacks and keep up performance so its users can safely and quickly get around online. On the Tor network this can be done with a bandwidth scanner. These scanners are run by special nodes called directory authorities which maintain a list of currently-running nodes on the Tor network. This project will improve the new bandwidth scanner so directory authorities can better use it and diagnose issues on the Tor network more quickly. Keeping closer watch on network performance and vulnerabilities ultimately will improve the performance of Tor and the quality of experience for its users, making it a more attractive alternative to the 'common' internet.

Technical description

The Tor network is comprised of thousands of volunteer-run relays around the world, and millions of people rely on it for privacy and freedom online everyday. To monitor the Tor network's performance, detect attacks on it, and better distribute load across the network, we employ what we call Tor bandwidth scanners. The bandwidth scanners are run by the directory authorities, which are special relays that maintains a list of currently-running relays. This project will make a number of improvements to the new bandwidth scanner call sbws, to make it easier for directory authorities to deploy it, for relay operators to better diagnose issues and for end users to benefit from increased quality of experience.

Visit https://NLnet.nl/project/OnBaSca

ZSipOs

Consumers and businesses overpay for computer hardware, because the market is not working well. When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component, the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

The issue of insecure hardware becomes even more important when you think of fast and widespread the use of smartphones has grown. The device that we carry with us every single day and use to call each other, do our personal banking, maintain our social life and manage a host of other online services with is frustratingly opaque and riddled with security vulnerabilities and backdoors. And because most smartphones are produced by a select number of massive companies, the entry to market for more secure and private alternative smartphone hardware is practically impossible.

One way to circumvent the status quo of smartphones is to go around the phone itself. Instead of designing a secure and private smartphone from scratch, the ZsipOS adds a plugin device to your phone that handles encoded internet telephony completely on its own. A user only has to connect the gadget to their phone and call someone. The program on the device will establish a cryptographic tunnel, basically a secure channel, that ensures no one can listen in or in anyway modify the call. Users also do not need to trust an external service provider to handle the call. Because the device is designed to only establish encrypted calls and because it handles everything instead of the smartphone it is connected to, there is little to no risk of an attacker getting in through some forgotten backdoor. The design of the device and the program it runs is completely transparent so security experts can test and verify everything that ZsipOS does and promises to do. Ultimately ZsipOS is an accessible, surefire and fully transparent solution for encrypted internet telephony that fits with any smartphone out there.

Technical description

ZSIPOs is a fully open source based encryption solution for internet telephony. It takes the shape of a little dedicated gadget you connect with a desktop phone. At its core the device does not have a normal chip capable of running regular software (including malware) but a so called FPGA (Field Programmable Gate Array). This means the device cannot be remotely updated (secure by design): the functionality is locked down into the chip, and the system is technically incapable of executing anything else. This means no risk of remote takeover by an attacker like with a normal computer or mobile phone connected to a network like the internet. The whole system is open hardware, and the full design is available for introspection. Normal users and security specialists get transparent access to the whole system and can easily check, what functionality is realized by the FPGA. This means anyone can verify the absence of both backdoors and bugs. ZSIPOs is designed to be fully compatible with the standard internet telephony system (SIP) which is the one used with traditional telephony numbers. The handling is done in principal by a regular internet phone (Dial, Confirm once – done). The cryptographic system is based on the standard RFC 6189 - ZRTP (with “Z” like Phil Zimmermann, the father of PGP), meaning it can also be used when using internet telephony on a laptop or mobile phone - of course without the additional guarantee of hardware isolation. There is no need to trust in an external service provider to establish the absolute privacy of speech communication. The exchange and verification of a secure key between the parties ensures end-to-end encryption, meaning that no third party can listen into the call. To that extent the device has a display to exchange security codes. The same approach can also also used for secure VPN Bridgeheads, secure storage devices and secure IoT applications and platforms. The ZSipOS approach is an appropriate answer on today security risks: it is completely decentralized, and has no dependency on central instances. It has a fully transparent design from encryption hardware to software. And it is easy to use with hundreds of millions of existing phones.

VIPcom GmbH — **Visit https://NLnet.nl/project/ZSipOs**

NGI0 PET Encryption HardwareIsolation OpenHardware Telephony

Wireguard Windows client

The internet is unfortunately not only populated just by kind and careful people. And it wasn't designed to be secure either. This is a dangerous and rather unfortunate combination of circumstances, and one you should take into account when you use the internet. When you go online outside of your house or office, chances are you use whatever wireless network you can find to get online. Mobile internet subscriptions are still expensive, so it is logical people seize every opportunity to connect for free. While this is a daily habit for many millions of people, and often

nothing bad happens, it does expose you to serious risks.

This is because your computer doesn't really know a lot about the world. It depend on information it gets from the networks it connects to. If the network happens to cheat, the computer often has very little defenses. If you use an untrustworthy network to connect you to the internet, you move yourself into the middle of enemy territory. While you happily enjoy your free bits it provides as a way to buy money, it has ample opportunity to exploit all kinds of security weaknesses against you. Essentially, if you got away scot free with connecting to an unsafe network, it probably wasn't your security that held anyone back. You were just lucky that no-one serious tried to do anything - this time. So it is recommended to not connect over wifi networks you don't know.

Unless of course you've arranged for a secure tunnel that allows you to teleport your internet traffic across the unsafe local network to the real internet unscathed. The concept is surprisingly simple: individual messages sent through the Internet, called packets, are encrypted using some mechanism, and this encrypted message then substitutes the original one, making all communications sent through the tunnel unreadable to eavesdroppers and unalterable to attackers. Proven cryptography protects the integrity of the traffic flowing through the tunnel. And once the packets reach the other end of the tunnel, they can be unpacked. From that point onwards they may continue their life as normal internet traffic. Travelling the path in reverse path is of course also possible: packets sent from the internet to your computer are protected in exactly the same way.

In anticipation of better technologies that should arrive with the next generation internet, such tunnels are a key technology to guarantee consumer safety. They play a major role in protecting users both from snooping and malicious traffic injection. Sadly, the tools to create these secure tunnels is rather cumbersome (if not plain hard) to work with. This has prevented mass adoption.

WireGuard is a completely new entrant to the field, and it is praised widely by technologists for its very high quality. Its goal is to be the most secure and easiest to use VPN solution available. Wireguard has many attractive traits: it is fast, simple and lean. It can run on embedded interfaces and super computers alike, and is fit for many different circumstances. Wireguard makes it very easy to set up a secure tunnel with modern technologies. It employs formally verified cryptographic constructions and has best in class performance. So you can more safely browse the web without annoying delay, even from potentially unsafe networks.

WireGuard starts from scratch with modern cryptography and best-practice defense-in-depth implementation strategies. It is suitable and easily deployable for both end users and in data centers across the world, and provides an essential core building block for making the Internet safer. Within the project the team will develop a fast and secure WireGuard client for the still widely used Microsoft Windows operating system, for which support is still immature and experimental.

Technical description

WireGuard is a next generation VPN protocol that uses state of the art cryptography. WireGuard allows to safely tunnel traffic across the internet. WireGuard presents a new abuse-resistant and high-performance alternative based on modern cryptography, with a focus on implementation and usability simplicity. It intends to be considerably more performant than OpenVPN. WireGuard is designed as a general purpose VPN for running on embedded interfaces and super computers alike, fit for many different circumstances. While still under heavy development, it is regarded by many as the most secure, easiest to use, and simplest VPN solution in the industry. Initially released for the Linux kernel, it is now cross-platform and the open source technology is ready for wide deployment. Unfortunately, WireGuard

support on the widely used Microsoft Windows operating system is still immature and experimental. This makes the technology unavailable to many desktop and notebook users. This project will deliver the first stable Windows version.

Amebis — **Visit https://NLnet.nl/project/WireGuardonWindows**

NGI0 PET DesktopClient Encryption SystemService VPN

Katzenpost

One of the things people enjoy the most about the internet, is that it enables them to talk to others remotely almost without limit. Internet allows anyone connected to keep closely connected with friends and family, and help their kids solve a math problem while they are at work. People collaborate with their colleagues from the couch of their living room, the cafe where they enjoy lunch or on their cell phone on the bus to the gym. Businesses can easily service their customers where this is most convenient to them, without having to travel themselves. This is so convenient, that some businesses have already moved entirely online. Internet communication has become the nerve center of whole neighbourhoods, where people watch over the possessions of their neighbours while these are away for work or leisure.

However, users have a hard time to understand how privacy is impacted if they use the wrong technology. Because internet works almost everywhere, the natural privacy protection of the walls of a house, a school or an office is gone. Unlike the traditional phone companies, many of the large technology providers run their business not on delivering an honest service but on secretly eavesdropping on their users and selling information to others. It is mostly not about what you say, so it is relatively easy for providers to allow some form of privacy by encrypting messages. The more interesting parts are who talks to whom, when, and where they are in the real world while they meet on the internet. if you want to be reachable across the internet, you have to constantly let the communication provider follow you wherever you go. This makes the private and professional lives of citizens an open book to companies that with the help of AI and other technologies make billions from selling 'hidden data' normal people are completely unaware of even exists. And of course in societies that are not so democratic, this type of information is critical to bring down opposition and stifle human rights.

End-to-end encryption has become more commonplace with major online messaging and communication tools, but encoding what you say to your friends online does not mean that the service provider cannot see who you contacted, when, from where. This metadata might be even more important than the content of an online conversation. If you want to profile or track someone, you can get a lot of information from the people they talk to, where they come from, who their friends are, etcetera. Katzenpost is a free software project that creates a decentralized and anonymous communication system and with this proposal, will add a security layer that prevents traffic analysis. Through traffic analysis third parties can intercept and examine messages to find out certain patterns, for example who is speaking to who, even though everything is encrypted. Resisting traffic analysis is an

important effort to ensure users actually private communication. This project aims to advance the state of art and provide a concrete building block for other applications to use and make their application more secure and private.

Technical description

Secure messaging is among the most fundamental privacy challenges of today. While there are meanwhile several widely used offerings that can encrypt instant messages you send to others, there are very few reliable options that are able to keep others from finding out who you were communicating with - and when. The most popular end-to-end messaging application do not adequately protect the identities of who-is-talking-to-who from the infrastructure operators. Katzenpost aims to offer a traffic analysis resistant messaging layer that allows all the participants in the network to have significantly more privacy than other mechanisms. It offers a decentralized mixnet architecture that works similarly to onion routing, where message routing information is encrypted, and differs in that each message is a fixed size, has random forwarding delays, and is accompanied by cover traffic messages to frustrate passive traffic analysis. The project aims to be a building block for other to build applications on, lowering the threshold for existing applications to benefit from increased privacy and confidentiality.

Visit https://NLnet.nl/project/katzenpost

NGI0 PET **Anonymity** **E2EE** **InstantMessaging** **Mixnet** **Observability** **P2P**

P o l i s c o o p s

POLISCOOPS
find local political news

Politics and government can be very confusing and exhausting. People with strong opinions and clashing world views debating at length about all kinds of niche topics and very broad and complex issues. And on the other side, endless formal announcements published in a bureaucratic way exhaust ones attention. It can be hard to stay connected, and in fact once you lose track of a thread you are kind of lost - because from that moment on you lack essential background knowledge while the discussion rushes on. This ´information debt' piles up, and at some point much of what is said may not mean a whole lot to the majority of people. Continuously repeating some of it may be good to allow people to reconnect, but it punishes the people that did pay attention and makes them disconnect. And lets not forget there are dishonest people too, that have no problem to exploit the fundamental inability for everyone to know it all, and try to take manipulate and use misinformation to enriching themselves.

To make things worse: there are different, partially overlapping levels of politics and government, from global politics to the village or city council you live in. There are many individuals and different political parties, meetings and official publication channels to track. People come and go all the time, new parties are founded, new coalitions cemented, new people appointed doing things differently. And the internet has enabled politics to go on 24/7, even further straining ones time budget and fragmenting the discussion. Politicians probe ideas and make subsequent decisions for you based on social media

feedback, while you are doing the dishes or standing next to the football field cheering the neighbour kid at her important match. How are you supposed to do your daily work, pay attention to the ones dear to you and keep up with all what is happening? Especially if the relevant messages are spread across the whole political spectrum. Tracking everything that could somehow be relevant is more than a full time job, or even more than enough to fill several lives in parallel. And yet: hidden in there are extremely important events and decisions that may directly influence your life. What if the topic of discussion is your street, budget cuts on the home for the elderly where your grandmother lives, or whether the toxicity levels in the playground where your kids play are reasonable and nothing has to be done? And if you notice something, how do you make sure others do?

The internet may have contributed to some of the information overload to citizens, but it also can help. This is where PoliFLW comes in. It can use the power of the crowd to "crowdsource" and curate relevant information in a very user friendly and non-obtrusive way. PoliFLW is an effort to index and organize political news that affects where you live and work. The platform scrapes social media accounts and collects news that (local) political parties share on their websites and and makes it searchable through tagging and indexing. Users can for example filter on date, location, source or political entity, or scrape the articles and sources for a specific topic. Why hasn't a decision been made about that dangerous crosswalk down the road? How has the municipality handled social care over the last few years? What do local parties think about addressing sustainability in your neighborhood (rather than just the political figure you voted for)?

PoliFLW is not just important to inform private individuals like yourself, but it is also a tool that greatly assists professionals like journalists. As watchdogs of democracy, journalists work hard to keep politicians honest and inform their audience about the latest political news. This is not an easy feat. They know how political parties are organized, where they get their money from, what their stances are on important topics and with the help of PoliFLW can show how the currently elected government came to decisions. Because of budget cuts local channels rarely have the time and money anymore to check up on local politics. At the same time, national and European politics becomes more connected and complex, making the political process obtuse rather than transparent. Making political data open or dumping information online does not solve this problem. Instead, such public data must be searchable, discoverable and accessible for journalists and curious citizens alike. The goal of PoliFLW is to include and organize news at all levels of politics, from hyperlocal to national parties to the European Parliament, in all member states.

PoliFLW gives everyone a clearer view of local political decisions and discussions that news media now tend to overlook. It can aid journalists to counter this negative trend and keep local politicians honest. And it can help normal citizens to maintain an overview across the whole political spectrum, and break out of political silo's and social media bubbles.

Technical description

PoliFLW is an interactive online platform that allows journalists and citizens to stay informed, and keep up to date with the growing group of political parties and politicians relevant to them - even those whose opinions they don't directly share. The prize-winning polical crowdsourcing platform makes finding hyperlocal, national and European political news relevant to the individual far easier. By aggregating the news political parties share on their websites and social media accounts, PoliFLW is a time-saving and citizen-engagement enhancing tool that brings the internet one step closer to being human-centric. In this project the platform will add the news shared by parties in the European Parliament and national parties in all EU member states. , showcasing what it can mean for access to information in Europe. There

will be a built-in translation function, making it easier to read news across country borders. PoliFLW is a collaborative environment that helps to create more societal dialogue and better informed citizens, breaking down political barriers.

Open State Foundation — **Visit https://NLnet.nl/project/PoliFLW**

NGI0 Discovery **Discoverability** **Discovery** **Indexing** **Metadata** **NGO** **Opendata** **Scraping**

---

s e a r x

Search and discovery is one of the most important and essential use cases of the internet. When you are in school and need to give a presentation, when you are looking for a job, trying to promote your business or finding relevant commercial or public services you need, most of the time you will turn to the internet and more importantly the search bar in your browser to find answers. Searching information and making sure your name, company or idea can be discovered is crucial for users, but they actually have little control over this. Search engines set the terms for what results you see, how your website can be discovered and what information is logged about your searches. What terms are set remains obscure for users and they can only follow the rules laid out for them, instead of deciding on their own what, where and how to find the information they are looking for.

More transparent, customizable and privacy-friendly search puts the user in the driver seat and can provide them with meaningful results. Searx does this by aggregating results from more than 70 search services while avoiding any user tracking or profiling. With every search users can decide what engines they want to use and which they don't, what search language must be used and other options that are saved on the device and can therefore not be tracked. Users are also free to run their own instance of searx, giving them complete control over the source code that makes that version of searx tick (and alter it however they like) and ensure additional privacy protection.

This project builds on the open and customizable setup of searx to provide users with extra privacy protective measures. Interested third parties can potentially use the IP address of that instance and the specific queries of its users to uniquely identify and follow them. This project will solve this issue while also making it easier for users to switch and maintain searx instances. Ultimately this new tooling can help to make searx more user-centric, stable and privacy-friendly.

Technical description

Searx is a popular meta-search engine, with the aim of protecting the privacy of its users. In the typical use case, few users trust one instance. However, a third-party services can easily fingerprint the users

using the IP address of the searx instance and the user's queries. The project aims to create a searx federation to solve this issue. First, a protocol needs to be defined to allow the instances to discover themselves. Then, each instance will be able to proxy the HTTPS requests through other instances, so the user only has to trust one instance. Also, each instance will spread the requests to other instance according to their response time, and make that IP addresses are evenly used, or at least in the best possible way. To ensure the latter, the statistics page will be enhanced and available through an API that other instances will use. The federation will make sure that bots can't abuse this pool of IP address.

- — **Visit https://NLnet.nl/project/FedSearx**

NGI0 Discovery **Deployability** **Federation** **Metasearch**

A Distributed Software Stack For Co-operation

The way our information is organised, has a huge impact on how society is organised. There is a lot of human activity that falls outside of existing commercial services. Society consists of families, unions, clubs, public offices, schools, public transport, sports, art, culture - a rich blend of individuals, formal organisations and ad hoc organic structures of all sizes. This complex fabric of society of people has been categorised by Kate Raworth in "The Doughnut Economy" into four sectors: the households, the commons, the state and the market. The latter two in particular are know to reach huge sizes (a relatively small amount of nation states and large multinationals), while the other two (almost by design) are millions of times larger in numbers but each of them remains small in size.

People perform and produce in households and the commons in all sorts of ways that are not visible from the other perpectives. People co-operate everywhere. They communicate, co-ordinate their actions and jointly achieve more then anyone could on their own. Given the superpowers of the internet, it is logical to support and improve that co-operation (including and perhaps especially ad hoc cooperation) with IT infrastructure. Due to the hyperscaling that is happening in the market and state, the tooling we use for both households and the commons is often not optimal. A family is not an office, and in fact behaves very differently. We are creative enough to support ourselves with what we have to our avail - how many people repurpose spreadsheets as a membership database, address book or an archiving system. There is a huge and ever changing variety of collaboration models and contexts, and the great variety of different tools needed to make optimal use of the technological possibilities could never be economically viable as products. Luckily, there are many similarities that allow for a vast amount of reuse - just like we have a limited set of slightly over 120 chemical elements that are the building blocks for the almost infinite amount of complex molecules that make up the universe, the amount of technology primitives we need to combine to enable a rich diversity of human collaboration is in fact limited too.

The Perspectives project aims to create the infrastructure that can empower an explosion of collaboration, not just limited to households and the commons but extending its use to all of human

activity. The goal is empowering people to come together to offer and exchange information, products or services with whomever they want. Perspectives is a project to build the necessary infrastructure for such online cooperation. Instead of focusing on search and keeping users in the dark about how searching actually works, Perspectives focuses on discovery and how actors in a specific domain (think of local transportation, accommodation, second hand goods, or matchmaking for that matter) organize supply and demand just the way they want to. With a flexible and federated (no central managing authority) foundation, Perspectives can accommodate diversity while maintaining a universal user experience. And because it is open technology, it can be reused, expanded and shared to accommodate any type of human activities users need support for.

Technical description

Perspectives aims to be to co-operation, what ActivityPub is to social networks. It provides the conceptual building blocks for co-operation, laying the groundwork for a federated, fully distributed infrastructure that supports endless varieties of co-operation. The declarative Perspectives Language allows a model to translate instantly in an application that supports multiple users to contribute to a shared process, each with her own unique perspective. The project builds a reference implementation of the distributed stack that executes these models of co-operation, and makes the information concerned searchable.

Real life is an endless affair of interlocking activities. Likewise, Perspectives models of services can overlap and build on common concepts, thus forming a federated conceptual space that allows users to move from one service to another as the need arises in a most natural way. Such an infrastructure functions as a map, promoting discovery, decreasing dependency on explicit search. However, rather than being an on-line information source to be searched, such the traditional Yellow Pages, Perspectives models allow their users (individuals and organisations alike) to interact and deal with each other on-line. Supply-demand matching in specific domains (e.g. local transport) integrates readily with such an infrastructure. Other patterns of integrating search with co-operation support form a promising area for further research.

Visit https://NLnet.nl/project/Perspectives

NGI0 Discovery **Collaboration** **Federation** **Modeling** **P2P** **Runtime** **Search**

---

variation graph (vgteam)

Worries over our health and safety will in many cases take precedence over the perceived value of our privacy. When it comes to our physical health and well-being, we are often in a strongly dependent position. Especially in times of great mental stress (like when a medical doctor breaks bad news to us) or fear (my daughter is late from school, a deadly virus is going round) we often

lack the time and knowledge to really consider what data we actually want to make available and under which conditions. Many people in such situations reach a point of detachment and panic, where they hand out whatever data requested from them by whomever promises to resolve the stress. And once data is out there, it is hard to trace back.

But what if we do not have to give up our privacy for the sake of better, and more personalized health care, fighting the spread of a pandemic or other safety measures? What if we can have both? The classic example is genetic research, which can be extremely effective in identifying hereditary diseases and ultimately creating a type of personal health care that perfectly fits your unique needs. It also involves extremely personal and uniquely identifying data (literally the DNA that made us the individuals we are), the wider availability of which has a potential impact on the privacy and physical security of your children and their children and their children's children etcetera. Who know what future generations will have to endure, in good times and in bad times? With the technology easily available to them, would insurance companies, employers or governments be tempted to test for yet undiscovered heart conditions or expensive and rare diseases - or worse? And yet we make important decisions about this in times of stress.

The same caution should go for a pandemic situation like SARS-CoV2. We all want a solution to help those most vulnerable, but as a society we are not prepared at all for the large security implications of exposing geolocation trajectory data seized from for instance telecom networks. And we cannot assume that lack of preparation will not be abused. And despite that we want to have a deep understanding how a virus actually spreads in a fine-grained way from person to person, meaning we need to gain insight in how people move around with actual data. For that purpose epidemiologists really could use access to a public database of personal movement trajectories, so they can do so called geophylogenetic modeling. SARS-CoV2 is not the first virus to cause a pandemic, and it will not be the last - and policy measures like a lockdown have an immense cost in terms of our economy and societal disruption. So we had better test our assumptions and create data sets with privacy preserving variation graphs that allow a wide community of researchers access without risk of security fallout afterwards.

Things do not have to be black and white. Doctors do not need to have access to all of our DNA in order to help us, so we don't have to share everything. Epidemiologists do not need to know Gabriel visited Mary, and how many times Mary met Elisabeth and when and where. As it turns out, there are clever ways to aggregate data in a privacy preserving way, preserve the characteristics needed and removing the rest. This project will build on these so called "variation graphs" to further explore and develop these technologies. There are applications throughout many other use cases as well - variation graphs can be used to produce privacy-preserving representations of collections of other sensitive data, including collections of personal writing, web browsing histories, or even quantified self. The general tenet is always to only share the relevant information, while preventing the identification of individuals.

Variation graphs have huge potential. The project is contributing to various very ambitious goals, such as assisting with the SARS-CoV2 situation and enabling the creation of searchable DNA databases that protect the individuals contributing in a provable way. Input data from healthy and non-healthy people, from sinners and saints, can be transformed in such a way that the privacy of all involved is protected while intensive study of DNA data or human movement patters remains possible. This will greatly help to convince people that they can contribute to the associated research. Of course success in such a critical application breaks the ice for all other use cases, where we see the benefit of big data but also the threats. Such a solution, if it becomes widely available, might be nothing short of revolutionary.

Technical description

Vgteam is pioneering privacy-preserving variation graphs, that allow to capture complex models and aggregate data resources with formal guarantees about the privacy of the individual data sources from which they were constructed. Variation graphs relate collections of sequences together as walks through a graph. They are traditionally applied to genomic data, where they support the compression and query of very large collections of genomes.

But there are many types of sensitive data that can be represented in a variation graph form, including geolocation trajectory data - the trajectories of individuals and vehicles through transportation networks. Epidemiologists can use a public database of personal movement trajectories to for instance do geophylogenetic modeling of a pandemic like SARS-CoV2. The idea is that one cannot see individual movements, but rather large scale flows of people across space that would be essential for understanding the likely places where a outbreak might spread. This is essential information to understand at scientific and political level how to best act in case of a pandemic, now and in the future.

The project will apply formal models of differential privacy to build variation graphs which do not leak information about the individuals whose data was used to construct them. For genomes, the techniques allow us to extend the traditional models to include phenotype and health information, maximizing their utility for biological research and clinical practice without risking the privacy of participants who shared their data to build them. For geolocation trajectory data, people can share data in the knowledge that their social graph is not exposed. The tools themselves are not limited to the above use cases, and open the doors to many other types of applications both online (web browsing histories, social media usage) and offline. .

Visit https://NLnet.nl/project/VariationGraph

NGI0 Discovery DNA PersonalData Search e-Health

Extending PeerTube

In the same year when the ARPAnet (the predecessor of the internet) was invented, people tuned into their tube televisions to watch a global live broadcast of astronauts first landing on the moon. If they missed that historical moment, that would be it. There was no ability for normal people to record television broadcasts, no ability to rewind or look back programmes from the online guide. At the turn of the millennium, three decades later, everyone was still watching traditional television: quite a few people may have had a video recorder, but this needed to be programmed in

advance or you would still miss your favourite tv programme. And there had better not be two programmes you would want to record at the same time.

That has all changed in recent years. On demand video via the internet has meanwhile assumed an important, but also somewhat controversial role. A tiny set of dominant online video hosting platforms (most people would have trouble naming more than two) has emerged, these control how hundreds of millions of users spend many billions of hours of human lives every year. The platform's features and algorithms determine what you see, who can be discovered (whether this is called "trending", "recommended" or "autoplay"), who is banned and deleted, and who is just left out of the spotlight. Users can only follow the patterns laid out for them on screen. The platforms also determine what information is logged about your searches and binge viewing behaviour, and privately decide who they sell your interests and location to. That is a far cry from the privacy granted by traditional television and radio broadcasting, where literally noone outside of the room could know which programme you would pick from the aether. What data is tracked, and what filters and algorithms are used by these online video platforms, remains opaque for users. Contrary to traditional media, the platforms feel no responsibility for checking facts: they focus on commercial value to them, not social value.

Relying on third party platforms is especially awkward for public services and organizations, as they have moral responsibilities to their citizens and constituencies to protect their privacy and promote democratic and social values. There is no reason for publicly funded and private content (possibly about you and me) or material in the public domain to be exclusively available through a foreign commercial service that may change their terms of data ownership and usage on the fly.

As a society, we want a diversity of independent platforms and search tools to facilitate a wide cultural arena. We should keep content open and available in a sustainable way, where we as a society can interact with it in a way that no-one feels exploited by or uncomfortable with. PeerTube is such an alternative to closed-off and commercial video platforms like YouTube. PeerTube is open source and free (free as in freedom) software that uses peer-to-peer technology to easily and quickly provide and share uploaded video material. Or put differently: a turnkey video platform in a box. Anyone that owns a computer connected to the internet can in principle create their own video platform, and set their own rules for users and content. Videos are stored by each instance independently, and so there is no censorship or systemic bias.

PeerTube in its current state already delivers the basic technology for federated public video hosting. But we are still a while away from industry strength deployments, needed to get public institutions, archives and other organizations to get large corpora of content online. This ambitious project will make a huge difference. It will increase the capabilities of PeerTube in terms of search technology, making it possible to even search inside the content of video. In addition, it will add to the accessibility features of PeerTube by signficantly improving subtitling support. It will make discovery of reusable content more easy, by implementing support for open licensing metadata, that communicate the legal conditions of specific content to search engines and users. This project will thus help pave the way for the massive caches of public media collections archived around the world to become first class citizens of the next generation internet. People will be handed the necessary tools to host and share any size of media collection, on a technology that is transparent from top to bottom.

Technical description

This project aims to extend PeerTube to support the availability, accessibility, and discoverability of large-scale public media collections on the next generation internet. Although PeerTube is technically capable to support the distribution of large public media collections, the platform currently lacks

practical examples and extensive documentation to achieve this in a timely and cost-efficient way. This project will function as a proof-of-concept that will showcase several compelling improvements to the PeerTube software by [1] developing and demonstrating the means needed for this end by migrating a large corpus of open video content, [2] implementing trustworthy open licensing metadata standards for video publication through the PeerTube platform, [3] and emphasizing the importance of accompanying subtitle files by recommending ways to generate them.

Beeld en Geluid — **Visit https://NLnet.nl/project/PeerTubeSearch**

NGI0 Discovery **CreativeCommons** **Decentralised** **Discoverability** **Discovery** **Education** **Foundation** **OpenData**

Explain

NGI ZERO

Every search starts with curiosity. You want to know the meaning of a word you had never heard before, find the news story behind that titillating headline you just caught on the television, or read up on that book your colleague kept raving about. You type in a few search terms, find the results you are looking for, and that's that. But what if you want to learn more about a certain topic, a complex issue, a cutting edge technology? Where do you begin, how do you get an overview of what is important and what is trivial, what should you read to stay up to date?

Users can find a wealth of open educational resources online, study material, tests and other content that can help you learn about a specific subject or topic. But just like a teacher in a classroom, you need some sort of guide to help you get started, to know where to go, to stay curious. The project Explain is that guide for anyone eager to learn from open educational resources, but is unsure how or where to begin. Explain provides users with an interface that connects and organizes resources from noteworthy educational repositories around the world to make them easily searchable. Users can also upload documents like exams, excerpts etcetera to Explain themselves to help other curious students. Now Explain aims to become a better guide through improving how material is indexed, how quickly and intuitively users can search and finally how intuitive the platform is for people to use. Ultimately this will contribute to what we believe is the core purpose of open educational resources: educate those who are curious, regardless of their background.

Technical description

The Explain project aims to bring open educational resources to the masses. Many disparate locations of learning material exist, but as of yet there isn't a single place which combines these resources to make them easily discoverable for learners. Using a broad array of deep content metadata extraction techniques developed in conjunction with the Delft University of Technology, the Explain search engine indexes content from a wide variety of sources. With this search engine, learners can then discover the learning material they need through a fine-grained topic search or through uploading their own content (eg. exams, rubrics, excerpts) for which learners require additional educational resources. The project

focuses on usability and discoverability of resources.

FeedbackFruits — **Visit https://NLnet.nl/project/Explain**

NGI0 Discovery **Extraction** **Indexing** **OER** **OpenEducationalResources** **VideoSearch**

Search and Displace

Everyone knows that once something is online, it can be hard if not impossible to take that information down again. This is especially risky when you need to share information on a document that also has particularly sensitive or even confidential data on it. Considering the amount of documents businesses, organizations and individuals share online everyday, mistakes are inevitable and potentially very harmful, possibly leading to (identity) theft, blackmail, or worse. Search and discovery in this sense is also a matter of privacy protection and granular control, the same way confidential details are sometimes redacted in government documents. This control should also be possible for documents that are already online, when 'the harm is already done' and you are desperately looking for a way to take a file down again or edit out any sensitive details.

This project can give users more control over what information they precisely want to share or publish online in their documents and what should be kept out of the public eye. A tool will be developed that can find out whether private or confidential information is leaked somewhere in the file and subsequently delete or cover up this data. The tool will cover documents in all shapes and sizes, ranging from digital forms and docs to printed files and even handwritten texts, and will be usable standalone or integrated in existing document or content management systems that organizations already use. The project aims to make a modular toolkit so the technology is relevant for all sorts of users, for example people working with government archives, court documents and legal contracts. Instead of forgoing transparency and data accessibility for privacy and confidentiality, this technology upholds both values crucial to a functioning democracy.

Technical description

The goal of this project is to establish a workflow and toolchain which can address the problem of mass search and displacement for document content where the original documents are in a range of forms, including a wide variety of digital document formats, both binary and more modern compressed XML forms, and potentially even encompassing older documents where the only surviving form is printed or even handwritten. The term "displacement" is meant to encompass actions taken on the discovered content that are beyond straight replacement, including content tagging and redaction, as well as more complex contextual and user-refined replacement on an iterative basis. It is assumed that this process will be a server application with documents uploaded as needed, on either an individual or bulk upload basis. The solution would be built in a modular fashion so that future deployments could deploy and/or

modify only the parts needed. In practical terms this involves the creation of an open source tool chain that facilitates searching for private and confidential content inside documents, for instance attachments to email messages or documents that are to be published on a website. The tool can subsequently be used for the secure and automated redaction of sensitive documents; by building this as a modular solution enables the solution to be used “standalone” with a simple GUI, or used via command line, or embedded within 3rd party systems such as document management systems, content management systems and machine learning systems. In addition a modular approach will facilitate the use of the solution both with different languages (natural and programming) and different specialities e.g. government archives, winning tenders, legal contracts, court documents etc..

Moorcrofts — **Visit https://NLnet.nl/project/searchanddisplace**

NGI0 Discovery **Anonymisation** **Redaction** **SearchReplace** **SensitiveDocuments**

WebXray Discovery

It is a scenario you probably run through several times a day on autopilot, without even noticing you are doing it: you are looking for some specific information, and submit some related key words to an online search engine. The search provider gets a large list of results, applies a set of ranking algorithms to it (pushing back potentially millions of results in favour of a handful of things it decides to push forward) , and you are given a single webpage that holds a shortlist of results together with some adds. Each of these results has a short description and a link to visit the page. A quick glance tells you that a couple of these results seem relevant to what you are looking for.

Normally, you would just click, find what you need or browse around. But what do you actually know about these sites, other than that they contain the same words that you were looking for? Once you've decided to click on a link, you have to reveal yourself to a significant extent to the website operator or its providers - but also to anyone they allow to be present on the website you visit. Your browser by default sends all kinds of fingerprintable information, some of which is unavoidable without active intervention and can be extremely telling (like ones current IP address). This can be combined with the context you are visiting: medical problem A, gossip B, professional interest C.

Many crucial internet and web standards were not designed with user privacy in mind, let alone giving users any sense of control over who can see what they do online. This opportunity for evel has been seized by all sorts of tracking and tracing schemes that make detailed profiles of people, which can then be (mis)used for commercial or even criminal gain. Another thing is to note is that trackers get to run software on your computer. So the minute you enter a web premise, you automatically start downloading all kinds of potentially risky things from around the internet, including payloads from other sites that you never actively chose to interact with. These downloads often include known attack vectors

like javascript, which (unless you actively take precautions) are even automatically executed.

Once you are on the web page you could probably search for and read the Terms of Service of the site. This may inform you that these third parties exist and are feasting away on your data, and that each has their own separate Terms of Service you could start looking for yourself. Note also that some sites contain dozens or even hundreds of trackers, which they combine with your context - and depending on what you were looking for, this can be quite telling. So at that point it is already too late.

Through regulations like the GPDR you may have the right to request what information is being captured from you, but in practical terms this is infeasible for normal people to do for every web page they come across looking for something. You just wanted to quickly search for something, remember. You can perhaps accept a company to do some analytics for its own purposes. You did not ask for an exponential exposure to a swarm of tracking companies that sell your data to the internet. In other words, you unknowingly opened a can of worms.

What you really want to know is: are the pages you are about to visit stuffed from top to bottom with hidden trackers, each of them with an unhealthy interest in as much of your online behaviour as you are unable to shield off? Who is actually behind these domain names (a single tracker company may have different web domains it hides under, and these can be changed within minutes. The data they collect remains piled up on a single mountain of observations, though. Who are the companies behind these shady business practices that index detailed information, where are the owners located and what law are they subject to?

How come that we can look through billions of pages of content in fractions of a second looking for any combination of words, but get no clue about what privacy or security we can expect there? And this despite the rather obvious nature of allowing other domains to take a peek on its visitors? A major step to taking back control of our online presence is to map out how our privacy is violated on a website by website basis, and by whom. If you ever used a tracker blocking app or browser extension, you will have seen tens or hundreds of unrecognizable titles and unfamiliar organization names. What if we can show you which of these 'parasitic' actors are where?

This is what the WebXRay Search project aims to do. It continuously runs across the web looking not for content, but for trackers. And it will make this information directly visible to you as you search, so before harm is done. If your current search engine operates trackers itself, it may not have a business interest to deploy this. This is why WebXRay Search will be made available through an extension for the privacy-friendly and customizable Searx meta search engine. This is a privacy-enhancing search proxy you can install yourself and share with others. You get the combined search results from many different sources, and for each of these results you can see what kind of tracking situation you will experience. And you will even be able to just automatically block results that feature the worst offenders. These ethical filters help inform and protect users and their privacy, because who wholly avoiding trackers is even better than using a tracking blockers.

In addition to being a very practical solution for the general audience, the results of the project will also be useful to institutions that enforce privacy legislation like the GDPR. These organisations will be able to visually check which organizations operate outside the law. Journalists or NGO's that research personal data collection can use the tool for their studies. Lets halt unsolicited and unlawful tracking and profiling, so people can just enjoy the web again without too much fear of their privacy.

Technical description

WebXray intends to build a filter extension for the popular and privacy-friendly meta-search Searx that

will show users what third party trackers are used on the sites in their results pages. Full transparency of what tracker is operated by what company is provided to users, who will be able to filter out sites that use particular trackers. This filter tool will be built on the unique ownership database WebXray maintains of tracking companies that collect personal data of website visitors.

Mapping the ownership of tracking companies which sell behavioural profiles of individuals, is critical for all privacy and trust-enhancing technologies. Considerable scrutiny is given to the large players who conduct third party tracking and advertising whilst little scrutiny is given to large numbers of smaller companies who collect and sell unknown volumes of personal data. Such collection is unsolicited, with invisible beneficiaries. The ease and speed of corporate registration provides the opportunity for data brokers to mitigate their liability when collecting data profiles. We must therefore establish a systematic database of data broker domain ownership.

The filter extension that will be the output of the project will make this ownership database visible and actionable to end users, and to curate the crowdsourced data and add it to the current database of ownership (which is already comprehensive, containing detailed information on more than 1,000 ad tech tracking domains).

Webxray — **Visit https://NLnet.nl/project/WebXRay**

NGI0 Discovery **DataBrokerage** **GDPR** **Profiling** **SearchAnnotation** **SearchFilter** **Tracking**

Openki.net

NGI ZERO

We all have things we can teach each other and conversely want to learn about. One person may know how to use the latest technology and is willing to teach others how to deal with that, another may know how to fix broken furniture or tools, or give a course on drawing or dealing with dementia. This continuous transfer of knowledge can happen through the internet, where it is relatively easy to connect with people. Or it intrinsically should happen off-line, where we can literally hold someones hand to show them how to paint or play an instrument. In the latter case, the internet just plays the role of intermediary - just like previously a note on the pin board in the local supermarket, village house or library did, just more effective.

We may have something we want to share with the world, but how to we make this information actually discoverable to others in the best possible way? And to people that are within a shared action radius, it makes no sense to offer tennis clinics to people on the other side of the planet.And people might not have decided they want to pursue learning tennis, chess, or yodeling - they are open to discovering new things. Traditional full text search engines have a limited understanding of location and facilitating serendipity, so even a popular blog may not reach the right audience. You need 'hyperlocal' understanding, combined with the right facilities for people to discover things of interest in an organic and diverse offering .

Global commercial platforms like meetup.com superfically fill part of this need. But obviously by far not all use cases are covered with services like that, and there can be significant privacy implications in using

a large commercial platform for such a delicate function. What if the course you are teaching or interested in is about dealing with a severe hereditary disease, in dealing with stalkers or something else that is sensitive? Maybe you don't want to be exposed to profiling and subsequent data transfer to an unknown amount of data brokers around the world - either as a teacher or as a student.

Users should be able to learn freely, and find each other as humans without fear of tracking or retribution. Openki is an open source project that wants to provide a platform for local education, where users can organize and find courses, learning groups and workshops in their area. The platform assigns roles like mentors (experts in their field), hosts (for example venues), organizers and participants and shows which course still needs a venue, whether a group is looking for a venue, or that everything is set for a great course on German literature at the library around the corner tomorrow. Now this project wants to move beyond hosting strictly local education organization and support broader grassroots initiatives, like bottom-up project initiation. This kind of platform fulfills the fundamental need for people to self-organize with technology that empowers instead of extracts.

Technical description

How do you discover what you can learn from the people around you? How do you search what other people in the same region have to offer, like a training course or a debating event?

Openki is an interface between technology and culture. It provides an interactive web platform developed with the goal to remove barriers for universal education for all. The platform makes it simple to organise and manage "peer-to-peer" courses. The platform can be self-hosted, and integrates with OpenStreetMap. At the moment Openki is focused on facilitating learning groups and workshops. The project will improve the tool, so it can be used not only to organise courses (with the collaboration of many different actors, in a more participatory way) but much broader,for bottom-up project initiation, for grassroot organizations and facilitating societal dialogue.

Verein Kopf — **Visit** **https://NLnet.nl/project/OpenKi**

NGI0 Discovery **Education** **EventSearch** **Grassroots** **OpenEducation** **SelfOrganisation** **Teaching**

---

eduVPN multi-protocol

Visit **https://NLnet.nl/project/eduVPN-multiprotocol**

VPN Fund **Architecture** **Multiprotocol** **VPN** **Wireguard**

Consumers and businesses overpay for computer hardware, because the market is not working well. When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component, the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

This ambitious project wants to deliver the first completely open computer processor in history - one you don' t have to merely trust, because you can verify and modify everything about it. All of the technology included, from top to bottom, will become available for inspection, and can be tuned by anyone technically capable enough. In practical terms, this will mostly be done by experts, but there is one major difference: anyone can become an expert, and noone has to ask permission to experiment. This will significantly contribute to the creation a new generation of computer technologies, as well as more energy efficient and cheaper devices.

Technical description

It is 2020 and it is not possible to buy a mass-produced laptop, tablet or smartphone and replace all of its software (with software that a user can trust) without loss of functionality. Processor boot-loaders are DRM-locked; WIFI, 3D Graphics and Video Processors are proprietary, and Intel's processors contain problematic features and intransparent elements such as the "Management" Engine. The most logical way to restore and engender trust is to literally make a new processor - one that is developed transparently and may be independently audited to the bedrock. The Libre-SOC project develops a low-power, mobile-class, 64-bit Quad-Core OpenPower SoC at a minimum 800mhz clock rate, suitable for tablet, netbook, and industrial embedded systems. Full source code files are available for the operating system and bootloader, and the actual processor, its peripherals and its 3D GPU and VPU. Details at http://libre-riscv.org/3d_gpu/

n/a — **Visit https://NLnet.nl/project/Libre-SOC-fund**

System-on-a-Chip Fund Hardware LibreSoC OpenHardware Risk-V

be trusted

As our lives get more digital every day, we use the internet to have important conversations - both personal and professionally. We also store and share more and more sensitive personal data on devices. On the internet you cannot just close the door to talk privately. So we need digital safe spaces and digital locks and vaults that are just as reliable and easy to use to store our secrets and mediate our communication.

Recently manufacturers have started to build so-called hardware enclaves or secure elements into their devices that function like a digital safe: even if someone is able to get some software installed into your computer, phone or laptop, they should not be able to immediately access what is in the safe.

But of course, creating a secure space or making a digital safe in an environment you don't really control or understand is really hard. All the technical protection no longer matters when someone can invisibly take control or peer over your shoulder. Especially since you as a user can't see yourself what is happening on the inside of your digital house. A safe and a rogue application can and will look completely identical to a user, and there is simply no way to distinguish among them based on their appearance. Users install many unknown games and applications all the time ("install our app to start getting discounts now!"), and forget that this is actually letting more or less random entities run unknown software on the phone that holds some of their most important information. And what if the operating system of your computer or phone itself has an unhealthy interest in your data or metadata, or is weakly protected to that others can just enter - similar to how unsafe it would feel if your landlord or the janitor is a peeping tom or a thief?

Betrusted is a dedicated open hardware device with the goal to create safe and more easily protected private channels for your communication. You can have a frivolous phone to play games, and do all the other things you meanwhile use your phone for. The Betrusted device is a complementary device that restricts itself to protecting the things that matter most, like your conversations and phone calls. It will also be able to hold passwords, digital versions of your passport (and other digital credentials and attributes), and whatever sensitive digital information you need to keep completely secure.

The idea is to create a portable, dedicated physical vault isolated from everything else you do, and with a deliberately limited feature set which makes it much harder to attack. The device can connect to your phone through wifi, and is ideally suited for so-called end-to-end encryption. This means you don't need a separate subscription. It does not matter if your phone is hacked or if the free wifi you use is safe or not - the internet skips your phone and betrusted can set up encrypted communication with end-to-end assurance.

The overall approach is security through isolation and simplicity: you can never leave a backdoor open if you don't build a door in the first place. As a user you can verify this, because the entire design and development of the device will be open to the public, from the software it runs down to the silicon that makes up its chips. A transparent, easy to use and secure digital safe that you can actually trust, with an configurable and easily understandable interface you want to use.

Technical description

Betrusted aims to be a secure communications device that is suitable for everyday use by non-technical

users of diverse backgrounds. We believe users shouldn't have to be experts in supply chain or cryptography to gain access to our ultimate goal: privacy and security one can count on. Today's "private key only" secure enclave chips are vulnerable to I/O manipulation. This means there is no essential correlation between what a user is told, and what is actually going on. Betrusted will build a full technology stack, including silicon, device, OS, and UX that is open for inspection and verification. Betrusted is a simple, secure, and strong device that aims to advance Internet freedom.

Visit https://NLnet.nl/project/betrusted

NGI0 PET FPGA HardwareIsolation Mobile OpenHardware

SASL Works for the InternetWide Architecture

Privacy is a matter of control. When you want to protect your privacy, it does not mean you never tell anyone anything, it means you want to be in control of who you share your personal information with. On the internet a lot of control is taken away from you. The technology that lets you connect to networks all around the world and find information anywhere it is stored is built around identification, both of its users and the virtual places they visit. Unfortunately, many crucial networking standards and protocols were not designed with user privacy in mind, let alone giving them any sense of control over how they can safely identify and authenticate themselves and whoever they want to communicate with on the internet. Secure identification and authentication should be the starting point for your online journey, instead of relying on workarounds and patches that may not cover all the exits.

Remaking the internet to be secure and private by design is what the ARPA2 project has been doing for several years by using and extending existing security standards and developing flexible, simple and reliable identity management solutions for users. Technology developed in the ARPA2 project simplifies and centralizes encryption and integrates state of art authentication and security standards, among other things. These tools help build an internet that "treats its end users as full-blown citizens, and not as milking cows", as the ARPA2 developers explain on their website. Just like other ARPA2 initiatives this project makes existing and proven internet technology more interoperable and usable to better protect users online data and identity on the internet. Users can identify and authenticate themselves more easily (instead of handling different passwords for every site) and securely (instead of relying on possibly broken identity management tools of third parties) regardless of the service they use, which puts the user back in the driver seat where they belong.

Technical description

The SASL Works allow clients to use authentication mechanism that meet their requirements, and use it

in virtually all protocols, which includes but is not limited to the web. Servers on the other hand, can flexibly adapt to clients from any domain, by backporting authentication inquiries to the client's own realm for the desired level of approval. Once configured, this process frees service providers from the need to manage user accounts and secure storage of credentials. Clients finally get a choice to use strong cryptographic authentication mechanisms instead of being forced to use a site programmer's poor approach to security. This in turn is helpful for setting higher levels of security policies in formal bodies such as organisations and governments, while generally simplifying the user interaction.

OpenFortress.nl / InternetWide.org — **Visit https://NLnet.nl/project/SASLworks**

NGI0 PET **Diameter** **IETF** **IdentityManagement** **RealmCrossover** **SASL** **SingleSignOn**

---

o ff e n

As you fire up your computer, laptop or smartphone and click your browser icon to connect to your favorite site, do you know what happens behind the scenes? Many websites actually have dozens of different trackers, and some of these have such a global presence that they can form a pretty clear picture of ones online behaviour. Some argue that privacy is and has been dead for quite some time. As long as users have a quick internet connection and can access the web, email, games and messages without a hitch, they won't complain. But if you question people about the importance of online privacy, usually the answer is that it is indeed important and should be better protected. What is happening here? Perhaps we misunderstand carelessness with unfamiliarity. The technology behind most of our devices, our connection to the internet and the virtual spaces we inhabit is complex, yes, but the solutions we use to access them have also kept actual control away from us under the guise of 'intuitiveness' and 'pick up and play'. Playing here means playing by the rules of the developer, not by your own. What users instead should have are tools that give them actual access to what their devices do, what choices are made, and decide for themselves whether they agree with them or not.

Privacy isn't dead, we just lack the tools to actually protect it. On the internet this would mean users need tools that first give them behind the scenes access and show how they are tracked and profiled. Then they should be able to flip a switch and decide, no, I don't want some unknown company to follow me around and record everything I do.

This is what offen will develop: a tool that gives users just as much insight and control as a website owner has over data gathering and analysis - putting both on an equal footing. And the user remains in full control: before any data is actually collected, users can see precisely what would be collected about them, and who that data would be shared with. Since website owners needs to convince the visitor that

they will respect her or his privacy in order to get their explicit consent, the rather than brutally grabbing any data she or he can get how that affects their privacy. Then they can either opt-in. This way, users have the tools and the access they need to make informed choices online and web site operators, who usually just want to know how many views their site gets and where their visitors are coming from, can respect the choices of their viewers.

Technical description

Transparently handling data in the open creates mutual trust: Offen is a web analytics software that gives users insights into the data they are generating by giving them access to the same suite of analytics tools site operators themselves are using. Usage metrics come with explanations about their meaning, relevance, usage and possible privacy implications, and also details which kind of data is not being collected. Offen treats both users and operators as parties of equal importance. Users can expect full transparency and are encouraged to make autonomous and informed decisions regarding the use of their data, and operators are being enabled to collect needed usage statistics while fully respecting their users' privacy and data. No user data is being collected until the user has explicitly opted-in. All data can be deleted either selectively or in its entirety by the users.

Visit https://NLnet.nl/project/offen

NGI0 PET **Analytics** **GPDR** **Opt-in**

G N U T a l e r

⟨Taler⟩

What website do you trust the most? Is it the search engine that gives you the results you are looking for, that news site that keeps you updated? What about the site or app that lets you access your money? Online banking over the last few years has become so user-friendly and commonplace that people pull up their phone to pay for their groceries or send their friends a payment request for the drinks they had last night. They trust their bank, which usually handles the site or app they use, to keep their money safe and make sure transactions go where they are supposed to. If that bank wants to keep its customers happy, it will make sure that banking site or app is safe and stays safe. But that does not mean it will not be interested in your payment data, or decide for you how the site or app works that lets you access your money.

Maybe there is a way to pay online just as you can in the real world: with money that you put in a wallet of your choice (one that only you have access to) and use to pay for whatever you want, anonymously. The open-source and free software (free as in freedom) collaborative GNU project is developing GNU Taler, a transparent electronic payment system that lets users do their finances privately and safely. Users exchange their existing money into digital currency that goes in their electronic wallet. Just like paying with cash, you only need a legal proof of payment when you buy something. GNU Taler lets you pas just as anonymously, while ensuring safety, promising immunity against fraud by phishing or creditcard information. Because the open-source banking environment does keep check of the incomes of selling parties, it also prevents money laundering or tax evasion.

Funding can help take GNU Taler the next step and be independently audited, so users can finally have a transparent, trustworthy and truly private online banking environment they get to choose themselves.

Technical description

GNU Taler is an advanced electronic payment system for privacy-preserving payments, also in traditional ("fiat") currencies like the Euro and the dollar. Unusually, the entire Taler system is free/libre software. Unique to the GNU Taler system is that it provides anonymity for customers, while delivering various anti-fraud measures. Payments can in principle be made in any existing currency, or a bank can be launched to support new currencies. Taler does fall under the usual regulations for electronic money issuers. One of the regulatory requirements that needs to be satisfied before payment providers can switch to this new system is that the payment service operator (the exchange) will need to be subjected to an independent auditor (which naturally would be paid for the service). In this project, a third party security audit of the GNU Taler codebase will be performed, creating a technology commons for fintech.

Taler Systems SA — **Visit https://NLnet.nl/project/GNUTaler**

NGI0 PET **Audit** **Cryptography** **ElectronicPayment** **FiatCurrencies** **Notary**

Privacy Enhancements for PowerDNS and DNSdist

NGI ZERO

If you want to look something up online, send an email to a friend or read the morning news, your computer panics and starts asking for help. How does it know where to retrieve or send anything? Luckily, it is connected to the domain name system. This naming system has been translating names users can remember (like ngi.eu or NLnet.nl) into numbers (or with a fancy word: addresses). Your computer has such a unique number itself, but it needs the numbers of the other computers you want to interact with to connect. You probably use domain names every day, whether you type in the address of a website, listen to a podcast or send an email.

It is called a domain name system for a reason, because it comprises more than just a naming convention. Getting a domain name involves talking to a lot of different computers. Your computer or phone basically doesn't know much about the world. One thing it does know, is how to ask that question to other, specialised computers. These computers actually also probably don't know themselves, unless they have recently answered the same question for another user. Names can change really fast for good reasons, so you would need to refresh this data a lot - otherwise users could end up on the wrong computer. The computers you sent your question to, thus pass the question on to other computers - and so forth. After just a few steps, some of the computers that were consulted get parts of the answer we were looking for. And at some point in time, the domain name system will have the entire answer. The magic happens so fast, most people are not even aware how complex this is. For them it "just works". One disadvantage: many other computers have learned something about us, about who we interact with

and about our interests - in an neatly labeled way. Someone is connecting to derspiegel.de or globaleaks.com. The more unique your question, the deeper the digging inside the DNS - and the more it stands out.

Domain names are at present an critical component for users, and so also a critical point of failure and a choke point. Without functioning DNS, most people will have a hard time finding basically anything on the network of networks. There have been cases where for instance a Spanish company got their domain name taken away, even though what they did inside Europe for European citizens was legitimate here. But not in the USA. And since the organisations that handle the .org, .com and .net domain names are based in the USA, these could be forced to remove these names from the DNS.

When DNS was designed, neither security nor resilience was that much of a concern for most users. The internet in its early days was not yet 'open to the public'. This of course has changed dramatically. The massive use of the internet and thereby our dependency on DNS has highlighted very important privacy and security issues with the design of DNS. At present, it is is not always capable of preventing misleading users nor can it prevent some leakage of what users do, who they talk to and where they go.

To make sure users can freely and privately search the web, over the years there have been numerous privacy protective additions made to DNS. Progress has definitely been made, but to actually keep users safe such technologies must be readily available to DNS providers, operators and generally everyone on the internet. This project will develop and contribute to open and trustworthy tools that can encrypt your DNS request as it leaves your computer, goes halfway around the world and comes back with the website you were looking for.

Technical description

DNS over TLS (DoT) and DNS over HTTPS (DoH) are two recent developments in the DNS field, and currently these are dominated by US based providers. The project will enhance the availability of open, trustworthy, privacy respecting DNS Resolvers in such a way that it allows any DNS provider, operator, or user to provide encrypted DNS service. This project aims to speed up implementation, improvement and standardisation of the most important Privacy enhancing features of DNSdist and PowerDNS resolvers to allow for the entire DNS-chain (from client, to caching-resolver, to authoritative nameserver) to be encrypted. The project will add support to the (open source) PowerDNS components (dnsdist, recursor and Authoritative server) for the privacy features necessary.

PowerDNS.com (part of Open Xchange) — **Visit https://NLnet.nl/project/PowerDNS-DNSdist**

NGI0 PET DNS DataMinimisation DoH DoT IETF

OpenPGP Certificate Authority

NGI ZERO

Email was designed without privacy or security in mind, which is amazing for such a popular service. When you send an email, anyone that can gain access to your mail server or the mail server of the recipient can read your mail, from top to bottom. And copy it, for later usage. It is often

compared to sending a post card, and of course in many cases there may be little harm in others reading what the weather is like in Paris. But what if you want to use email to send something confidential, something you do not want to share with others? Like a love letter, a political rant or an important contract? And what if you can't actually trust the mail man, for instance because the other party is using a free email service known to search through everything? Or what if you don' t like the fact that your writings are stored in a country you have never been, with different laws that may not be compatible with your thoughts about the world? Or what if you live in a country that has an unhealthy interest in bringing down certain political voices, or are part of a cultural minority that is at risk?

Computer specialists have been protecting their email with encryption for decades. This is the equivalent of putting your message very carefully in the blender, pressing the button before anyone else has read your mesage, shredding it up and sending a packet of shreds over to the other end. The amazing thing about cryptography is that you can magically (or rather mathematically) make it possible for your secret love - and not anyone else - to recreate the message from the shreds, and know it was you - and not anyone else - that sent it. For the rest of the world, the message would be meaningless garble pretty much forever.

However, the solution they came up with is not easy for normal people to work with. You need a lot of patience and technical skill to make use of it. Many people have tried, and could not get it to work or gave up because it hindered them. It was in fact too hard to turn it on by default. This means that most people are probably not even aware that it is possible to protect the contents of their email with cryptography. And so, unfortunately, normal citizens and business have been left behind - exposed to people reading their email messages, and (in the absence of other security measures) potentially also receiving fake or manipulated messages.

Such proven privacy-friendly technology should not go to waste. This project aims to fix encryption's usability problem for organizations and their members through using something called a certificate authority. This can be an organizations administrator for example, which already handles and manages most of the communication going on inside and out. The same goes for encryption: technically complex tasks like cryptographic key creation and handling is automated and authentication of each others encryption keys and signatures is handled by the certificate authority. Just like they rely on central management for other administrative tasks, organization members can be sure encryption is handled as they securely and privately email each other.

Technical description

OpenPGP CA is a tool for managing OpenPGP keys within an organization. Its primary goal is to make it trivial for end users to authenticate the OpenPGP keys of users in their organization, and in adjacent organizations. In an OpenPGP CA-using organization, users delegate authentication to an in-house CA. This allows users to securely and seamlessly communicate via PGP-encrypted email without having to manually compare fingerprints, without having to understand OpenPGP keys or signatures, and without having to trust a third-party with potentially conflicting interests. This goal is achieved by shifting the authentication burden from individual users to an organization's administrator, and providing a tool that largely automates key creation, and signing as well as key dissemination. Importantly, because OpenPGP CA works within the existing OpenPGP framework, users do not need any new software to take advantage of OpenPGP CA's benefits; they can continue to use existing email clients and encryption plugins. Further, OpenPGP CA can co-exist with other authentication approaches, like traditional key signing workflows.

Visit https://NLnet.nl/project/OpenPGP-CA

NGI0 PET Deployability OpenPGP TrustDelegation UX

Tracking the Trackers

Everyone knows there is no such thing as a free lunch. But when you are browsing through an app store, it sure looks like there is a wealth of free apps. Software companies have learned that users can put up with in-app advertising and data tracking as long as the app works relatively well (and the ads don't last too long). But what if a user doesn't want to be followed and profiled? Or what if the user is underage? In a user-centric internet, you should be able to choose. F-Droid is a necessary alternative to proprietary app stores that instead focuses on free, open source software that respect user privacy. So-called 'anti-features' like advertising, tracking and software dependencies on nonfree technology are clearly flagged and marked in app descriptions. This way users can make informed choices what software they put on their phone and be sure they are not tracked without their knowing.

Any privacy-friendly and transparent alternative to proprietary digital spaces is fundamentally built on trust. Users rely on the community behind F-Droid to rigorously audit the software in the app store and make sure no hidden tracking and profiling features are sneaked in. As the offer on F-Droid grows, auditing becomes more taxing and time-consuming for the community and new apps may end up in the app store that do not meet F-Droid's ethical requirements which can turn off users. This project helps speed up the review process, without fully automating it. New machine learning tools can quickly identify tracking and advertising technology in Android apps so an auditer can focus on making informed decisions, instead of having to do a lot of manual searching. Ultimately this will help keep F-Droid free of (hidden) tracking and advertising in apps and remain a trustworthy, user-centric and privacy-friendly app store.

Technical description

F-Droid is a free software, community app store that has been working since 2010 to make all forms of tracking and advertising visible to users. It is the trusted name for privacy in Android, and app developers who sell based on privacy make the extra effort to get their apps included in the F-Droid.org collection. These include Nextcloud, Tor Browser, TAZ.de, and Tutanota. Auditing apps for tracking is labor intensive and error prone, yet ever more in demand. Our tools already aide F-Droid contributors in this process. This project creates new tools using machine learning to drastically speed up this process by augmenting the human review process. Since the prime motivation of the F-Droid community is ethical software distribution, algorithms will never replace humans in making ethical decisions. We will also

explore using machine learning to detect tracking in a more generic way, without requiring manually compiled lists of key information. The resulting tools will be generally available for any use case needing to reliably detect trackers in Android apps. This builds upon our collaboration with Exodus Privacy and LibScout.

Guardian Project — **Visit https://NLnet.nl/project/F-Droid-Trackers**

NGI0 PET **Appstore** **MachineLearning** **MobileApps** **Packaging** **SecurityAudit** **Tracking**

Zerocat Chipflasher Flashrom Interface

Most users rely on antivirus programs to keep their system and important data safe and private. Visited sites, downloaded files, email coming in and out, everything should pass through a digital border control that keeps malware and spyware out. Perform a complete system scan every other month and most users will be reassured: I am safe. The truth is that there is more than one way into your system and not every backdoor is properly protected. Attackers can also target the BIOS (Basic Input/Output System) program that every computer has to boot up and load the operating system. The BIOS is the first process to run when you power on your computer and is usually not scanned by any antivirus or security software you have installed. Accessing the BIOS and installing malicious software on such a fundamental level gives attackers far-reaching control over a system (which is why it is used for ransomware) and the user usually does not even realize it. And updating their BIOS probably is not something they do (if they are even aware of it at all).

Fortunately, there are plenty of open-source tools developed over the years that can completely secure your system, down from the hardware and the BIOS up to the software you use. Unfortunately, the barrier to entry of many tools is probably too high for most users, who will not now where to begin and get lost in a maze of technical details. Which program is better than the rest, how can I make tool A work with framework B? And how will all of this affect my system, can I still use my computer the way I am used to? The Zerocat Chipflasher Project wants to empower users to create their own trustworthy devices, remove any proprietary BIOS firmware and instead run and install verifiable open source, free software.

Technical description

The Zerocat Chipflasher Project aims to provide a fully user controlled electronic device, that helps users to remove the proprietary BIOS firmware from their laptops. The tool allows them to instead run verifiable and Free Firmware, produced by the Coreboot and Libreboot project. Proprietary BIOS is opaque with regards to functionality, and may contain known and unknown security issues. Also

controversial elements like the Intel Management Engine can be deactivated. The project helps to empower everyone to create trustworthy digital hardware on her or his own and has been successfully certified by the Respects-Your-Freedom (RYF) Certification Program, set up by the Free Software Foundation in Boston, USA. The device combines the Do-it-Yourself concept with free-design hardware development, even down to chip level. This is achieved by skipping convenient functionalities which would require chips of a proprietary design and by instead using a free-design microcontroller, only. The flasher's integration into the grid of related existing free software projects yet is to be improved by an additional interface and an in depth firmware review.

Zerocat.org — **Visit https://NLnet.nl/project/Chipflasher**

NGI0 PET BIOS DIY Firmware Flashrom OpenHardware

Pixelfed

After you take a picture of your brand new car, your smiling baby or the food you were just served, what do you do? You want to show it to everyone you know of course. But do you really know who you are actually sharing your private snapshots with when you post them online? With high grade cameras in nearly every mobile phone and numerous instant messaging apps and social media platforms available, sharing photos is just as easy (and perhaps more popular) than typing out what you want your friends and family to know about your life.

Social platforms and apps make us feel like we are only sharing our images with our own social circle and maybe some faraway friends we met online. But because many so-called 'free' social sharing tools like Instagram actually monetize your data and online activity to sell you personalized ads, your online picture book may not be so private at all. And where do those snapshots, that sometimes contain very personal information about where you live, what you are doing and who you know, actually end up after you clicked that upload button?

When you want to show someone your holiday pictures, you simply want to share those pictures, instead of also handing over a copy to the postal service to check where you went to and possibly send you a cheap flight deal for the coming holidays. Pixelfed is a platform that makes this possible on the internet. Users can choose to run and host the service themselves or choose someone they trust to store their pictures and private data with. No one will track what photos you share and which people you follow. The pictures your friends and family share pop up in your timeline one after the other, without ads or algorithms that decide what you can and cannot see. This project aims to give users tools and features they can use to search, find and share photos on the platform, making Pixelfed a more attractive (and ethical) alternative to for example Instagram.

Technical description

Pixelfed is an open source and decentralised photo sharing platform, in the same vein as services like Instagram. The twist is that you can yourself run the service, or pick a reliable party to run it for you. Who better to trust with your privacy and the privacy of the people that follow you? The magic behind this is the ActivityPub protocol - which means you can comment, follow, like and share from other Pixelfed servers around the world as if you were all on the same website. Timelines are in chronological order, and there is no need to track users or sell their data. The project has many features including Discover, Hashtags, Geotagging, Photo Albums, Photo Filters and a few still in development like Ephemeral Stories. The goal of the project is among others to solidify the technical base, add new features and design and build a mobile app that is compatible with Mastodon apps like Fedilab and Tusky.

Visit https://NLnet.nl/project/Pixelfed

NGI0 Discovery ActivityPub Federation PhotoSharing SocialMedia

---

N y x t

Many of the hours we spend online, we spend within the windows and tabs of a web browser. We may not always realize it, but these applications are essentially gatekeepers to the modern web. Not many businesses can say that their technology has millions, even billions of users that fire up their program every day and use it to find most of the information they will consume that day. How these organizations setup their browser, and what kind of control they give the user, actually shapes how we access many public services, social spaces and business environments online.

To a great degree, web browsers define how users see the internet and more specifically, use online search. When you want to find something, why should you only see a list of search results on the left and advertisements on the right? There are countless ways we can find our way on the web, but as long as the vehicles we use look and work basically the same and do not really let users check what is under the hood, we cannot move forward to find new ways of discovery.

This project will give users a browser that is like a toolkit: to organize and look through information online, they can select precisely the tools they need, or even build new ones, while the browser is actually running. Ultimately, they can build a browser that is uniquely their own and that filters the insane amounts of online data they traverse exactly how they need it. Just like nobody learns in the same way, no one finds and consumes information in the same way. So why should we search and browse the same?

Technical description

Nyxt is a new type of web browser designed to empower users to find and filter information on the Internet. Web browsers today, largely compete on performance in rendering, all whilst maintaining similar UIs. The common UI they employ is easy to learn, though unfortunately it is not effective for traversing the internet due to its limited capabilities. This presents itself as a problem when a user is trying to navigate the large amounts of data on the Internet and in their open tabs. To deal with this problem, Nyxt offers a set of powerful tools to index and jump around one's open tabs, through search results and the wider Internet. For example, Nyxt offers the ability for the user to filter and process their open tabs by semantic content search. Because each workflow and discipline is unique, the real advantage of Nyxt is in its fully programmable and open API. The user is free to modify Nyxt in any way they wish, even whilst it is running.

Atlas Engineer LLC — **Visit https://NLnet.nl/project/NyxtBrowser**

NGI0 Discovery **Browser** **Integration**

---

o p e n E n g i a d i n a

openEngiadina

At your local supermarket or community center, everyone in the neighborhood can put up a note to announce a local event, sell something they do not longer need, or organize a fun party. Search and discovery in this way is organized completely equally: as long as no one messes with the notes or tries to keep someone away from the note board, everyone is free to search for what they need or have their services and products be discovered.

On the internet, search and discovery is not always equally organized. Most users are free to get together, start a website and share what they want to share, but how can they be sure they are actually discovered? That is governed by search engines and social platforms, who make up their own rules which sites, profiles and messages are displayed in their search results and how they are ranked. Users have to rely on these intermediaries to get their information out into the world, and usually have no other choice but to simply accept the terms of service of these platforms.

What if we could make online search work just as simple, democratic and transparent as a note board? Or even better? This project helps to make search and discovery on the internet more democratic by giving users the tools and technology they need to put their information out there on their own. If you want to organize a block party and announce the date to people in your community, you do not need to blindly trust some company to hopefully include your data in a list of search results users see when they are looking for local festivities. Instead, you can simply announce your party date in a particular way, and the technology this project will develop makes sure that this information can be easily found. This way, online search becomes a democratic community effort, instead of a commercial popularity contest.

Technical description

OpenEngiadina is developing a platform for open local knowledge - a mashup between a semantic

knowledge base (like Wikipedia) and a social network using the ActivityPub protocol. openEngiadina is being developed with small municipalities and local organizations in mind, and wants to explore the intersection of Linked Data and social networks - a 'semantic social network'.

openEngiadina started off as a platform for creating, publishing and using open local knowledge. The structured data allows for semantic queries and intelligent discovery of information. The ActivityPub protocol enables decentralized creation and federation of such structured data, so that local knowledge can be created by indepent actors in a certain area (e.g. a music association publishes concert location and timing). The project aims to develop a backend allowing such a platform, research ideas into user interfaces and strengthen the ties between the Linked Data and decentralized social networking communities.

Visit https://NLnet.nl/project/openEngiadina

NGI0 Discovery **ActivityPub** **Hyperlocal** **LinkedData**

G N U G u i x

Guix

Technical description

GNU Guix is a universal functional package manager and operating system which respects the freedom of computer users. It focuses on bootstrappability and reproducibility to give the users strong guarantees on the integrity of the full software stack they are running. It supports atomic upgrades and roll-backs which make for an effectively unbreakable system. This project aims to enhance multiple facets; the main three goals are: (1) distributed package distribution (e.g. over IPFS), (2) composable and programmable user configurations / services (a way to replace "dotfiles" by modules that can be distributed and serve a wide audience), (3) broaden accessibility via, among others, a graphical user interface for installation / package management.

Visit https://NLnet.nl/project/GUIX

NGI0 Discovery **Deployment** **Services** **Sharing** **Software**

Search and discovery are some of the most important and essential use cases of the internet. When you are in school and need to give a presentation or write a paper, when you are looking for a job, trying to promote your business or finding relevant commercial or public services you need, most of the time you will turn to the internet and more importantly the search bar in your browser to find answers. Searching information and making sure your name, company or idea can be discovered is crucial for users, but they actually have little control over this. Search engines decide what results you see, how your website can be discovered and what information is logged about your searches. And because many fundamental internet technologies were not designed with security or privacy in mind, it is quite simple to identify you online (and difficult to shield off what you do, search for and lookup). What filters and algorithms search technology apply usually remain opaque for users. They can only follow the rules laid out for them, instead of deciding on their own what, where and how to find the information they are looking for.

Most people would be quite surprised and very uncomfortable if every time they visited a library, someone walks behind them to write down their name, precisely time how long they look at a certain row of books and note what titles they take with them. All this data however is registered by most commercial search engines. This project helps users protect their online privacy when they look up information online by mixing up all kinds of very personal data (not just your search terms, but what computer you use, where you live, etcetera) in such a way that it becomes next to impossible to uniquely identify you. This prevents search engines and platforms from taking your personal data and building very personal profiles to sell you ads and unnecessarily 'personalize' what search results you get to see and what remains hidden from you. Users can simply install this technology as an extension to their browser and search the way they are used to.

Technical description

The minedive project is building several components: first, minedive is a browser extension aiming to allow users to search the web while preserving their anonymity and privacy. The second is an open source reference implementation of its rendez-vous server. minedive instances connect each-other (via WebRTC data channels) forming a two layered P2P network. The lower layer (L1) provides routing, the upper layer (L2) provides anonymous and encrypted communication among peers acting as a MIX network. This architecture guarantees that peers which know your IP address (L1) do not know search data for (L2) and vice-versa. A central (websocket) rendez-vous server is needed to find and connect with L1 peers, and to exchange keys with L2 peers, but no search goes through it. We are running a default server which can be overridden by users who want to run their own (using our reference implementation or a custom one). Users can also set the extension to pick peers from a given community (identified by an opaque tag). Currently all requests are satisfied by letting L2 peers return results from the 1st page of mainstream search engines (as they see it, in an attempt to escape the search bubble). While this will stay as a fallback, we plan to implement web crawling on peers, doing keyword extraction from URLs in local bookmarks and history and ranking with open algorithms, being transparent with users about which techniques are used and open to suggestions.

CKIN — **Visit https://NLnet.nl/project/minedive**

NGI0 Discovery **Anonymity** **Overlay** **P2Psearch**

Handling Data from IPv6 Scanning

The internet, when you put it very simply, is like a phone book. If you want to reach someone, you pickup the phone (or fire up your device) and call a specific number (type in a website or email address with a particular domain name). Now that not only our computers need addresses, but also our phones, televisions, and even our refrigerators, we have been quickly reaching the point where the phone book becomes full.

Luckily, there are ways to make the phone book of internet addresses a great deal larger, so that there are thousands of times more addresses for all the sensors and devices we are currently installing everywhere around us. Switching from the old to the new phone book is not without problems however, and the new address space is actually so massive, we can hardly keep track of it all. This project takes a smart approach to scanning new internet addresses and will help us keep tabs on how the 'new' internet is doing.

Technical description

Scanning is state of the art to discover hosts on the Internet. Today’s scanning relies on IPv4 and simply probes all possible addresses. But global IPv6 adoption will render brute-forcing useless due to the sheer size of the IPv6 address space, and demands more sophisticated ways of target generation. Our team developed such an approach that generally allows to probe all subnets in the currently deployed IPv6 Internet within reasonable time. Positive responses are however scarce in the IPv6 Internet; thus, we include error messages in our analysis as they provide meaningful insight into the current deployment status of networks. First experiments covering only parts of the Internet were promising and at least 5% of our probes trigger error messages. However, a full scan would lead to approx. 10^14 responses causing Petabytes of data, and demands an adequate solution of data handling. In this project, we will develop a data storage and analysis solution for high-speed IPv6 scanning. It will process the high amount of received data concurrently with scanning, and provide continuous results while scanning for long periods. This effort enables full scans of the IPv6 Internet.

SBA Research — **Visit https://NLnet.nl/project/ipv6scanning**

NGI0 Discovery **IPv6** **Probing**

Search and discovery are some of the most important and essential use cases of the internet. When you are in school and need to give a presentation or write a paper, when you are looking for a job, trying to promote your business or finding relevant commercial or public services you need, most of the time you will turn to the internet and more importantly the search bar in your browser to find answers. Searching information and making sure your name, company or idea can be discovered is crucial for users, but they actually have little control over this. Search engines decide what results you see, how your website can be discovered and what information is logged about your searches. What filters and algorithms are used is unclear for users. They can only follow the rules laid out for them, instead of deciding on their own what, where and how to find the information they are looking for.

Instead of relying on just a few companies for the incredibly important task of organizing online information, users can also collaborate and organize search and discovery together, providing more control over indexing and ranking, as well as better privacy protection. YaCy is a long-running project where peers index the contents of websites themselves and use decentralized search software that is not managed by a central organization or authority , preventing for example censorship or user tracking.

The peer-to-peer approach to web search not only gives users more control and privacy protection, it can also allow organizations, businesses and individuals to organize their own search portal. This project intends to use the existing indexing and search technology of YaCy to create decentralized, peer-to-peer search as a turnkey service. This can be useful for example for universities who need specific search tools to go through massive caches of scientific research, or companies that want to index and look through specific domain knowledge. This way, anyone can easily customize search how they see fit, all the while protecting user privacy.

Technical description

YaCy Grid Search-as-a-Service creates document crawling indexing functionality for everyone. Users of this new platform will be able to create their custom search portal by defining their own document corpus. Such a service is an advantage as a privacy or branding tool, but also allows scientific research and annotation of semantic content. User-group specific domain knowledge can be organized for custom applications such as fueling artificial intelligence analysis. This should be a benefit i.e. for private persons, journalists, scientists and large groups of people in communities like universities and companies. Instances of the portal should be able to self-support themselves financially: there is turn-key infrastructure to handle payments for crawling/indexing amounts as a subscription on a periodical basis while search requests are free for everyone. The portal will consist of free software, and users can download the portal software itself together with the acquired search index data - so everyone can start running a portal for themselves whenever they want.

YaCy.net — **Visit https://NLnet.nl/project/YacyGrid**

Librecast Live

If you want to share your message (or data) with the world, you send a packet that travels across a great deal of the networks that make up the internet to finally reach its destination and deliver someone the file, video or software they were looking for. There are actually quite a number of routes your data can take and ways to deliver messages over the internet, each with their own ups and downsides. The Next Generation Internet intends to create a more user-centric, private and decentralized internet. One of the ways to reach this goal is by making internet routing itself more privacy-friendly and decentralized.

An existing, but underappreciated routing scheme that can make our internet traffic less privacy invasive and more decentralized, is multicast. Instead of unicast which delivers something to one specific point in a network, multicast shares a message with a group of nodes that express their interest in the particular package. Such a node can for example be an internet service provider, where users can subscribe to particular content. Think of it like a television channel where you can subscribe to all sorts of stations broadcasting their programs, or Twitter, where you can follow people without needing to register anywhere.

Multicast has the potential to level the playing field for parties that want to transmit sizable traffic and to protect the privacy of users. Instead of having to rely on some facilitating third party to transmit large or continuous files to a thousands or millions of people, you can route your content instead more efficiently to several hundred internet service providers that your user base can then subscribe to. This has the added benefit for users that they would only need to register with their ISP for this particular content, instead of disclosing to the entire world what they want to watch. This project aims to make multicast easier to deploy and will develop a live streaming video service that users on decentralized, self-hosted social media can use to privately stream and chat with. Making multicast more usable can ultimately help to create an internet that is decentralized and private by default.

Technical description

The Librecast Live project contributes to decentralizing the Internet by enabling multicast. Multicast is a major network capability for a secure, decentralized and private by default Next Generation Internet. The original design goals of the Internet do not match today's privacy and security needs, and this is evident in the technologies in use today. There are many situations where multicast can already be deployed on the Internet, but also some that are not. This project will build transitional protocols and software to extend the reach of multicast and enable easy deployment by software developers. Amongst others it will produce a C library and POC code using a tunneling method to make multicast available to the entire Internet, regardless of upstream support. We will then use these multicast libraries, WebRTC and the W3C-approved ActivityPub protocol to build a live streaming video service similar to twitch.tv. This will be a complement to the existing decentralised Mastodon and Peertube projects, and will integrate with

these services using ActivePub. By doing so we can bring live video streaming services to these existing decentralised userbases and demonstrate the power of multicast at the same time. Users will be able to chat and comment in realtime during streaming (similar to YouTube live streaming). This fills an important gap in the Open Source decentralised space. All video and chat messages will be transmitted over encrypted channels.

Visit https://NLnet.nl/project/LibrecastLive

NGI0 Discovery Decentralisation Encryption Multicast Routing Tunneling

Nextcloud

Everyone knows that once something is online, it can be hard if not impossible to take that information down again. This is especially risky when you need to share information on a document that also has particularly sensitive or even confidential data on it. Today, countless individuals, businesses and governments use and rely on cloud services to share and manage data online in a controlled space. But how much control do these users really have over their data? Ask yourself this: when you 'rent' cloud space from a provider, where and how does this organization actually store your data? How is it protected, how can you be sure that the personal information you have in there, is kept away from prying eyes? And what's more, where does this provider actually store your data, in what country and under which laws?

One of the ways you can actually answer these questions is by hosting cloud space yourself or choosing who will host it for you and where, using technology you can check and modify as you want. Nextcloud offers this functionality with an open source solution for file hosting where users keep complete control over their data. This can be crucial for organizations and businesses who manage personal or sensitive information. Of course these users also need to go through their data just as easy as a competitor would that uses public cloud services. This project aims to develop a user-friendly interface for search across privately hosted cloud spaces, where you can select and search for data based on date, type, owner, size, keywords and other useful criteria. It is even possible to also go through data outside of the cloud space you host yourselves and search your offline database. Self-hosting online files should be just as useful as any proprietary cloud service, and this project will help to level that playing field.

Technical description

The internet helps people to work, manage, share and access information and documents. Proprietary cloud services from large vendors like Microsoft, Google, Dropbox and others cannot offer the privacy and security guarantees users need. Nextcloud is a 100% open source solution where all information can stay on premise, with the protected users choose themselves. The Nextcloud Search project will solve the last remaining open issue which is unified, convenient and intelligent search and discoverability of

data. The goal is to build a powerful but user friendly user interface for search across the entire private cloud. It will be possible to select data date, type, owner, size, keywords, tags and other metadata. The backend will offers indexing and searching of file based content, as well as integrated search for other contents like text chats, calendar entries, contacts, comments and other data. It will integrate with the private search capabilities of Searx. As a result the users will have the same powerful search functionalities they know and like elsewhere, but respecting the privacy of users and strict regulations like the GDPR.

Nextcloud — **Visit https://NLnet.nl/project/NextCloudSearch**

NGI0 Discovery **Indexing** **PrivateSearch**

GNU Mes on ARM

When you start up your computer, you will probably think twice before you download some random piece of software from the internet and run it. You know that doing so could allow unwelcome guests to your computer and your data. Your computer might even end up in a bot net. So when you see some nice piece of software, you will ask yourself the question: can I really trust the software? Perhaps you will check the origin it comes from. Better safe than sorry.

Did you miss checking something, though? What about the software that is already on your computer before you started? A computer is not much use without an operating system. Have you thought about the trustworthiness of that fundamental piece of software - your most fundamental travel companion on the wild west of the internet? While most computers are sold with an operating system, actually you have the choice to remove that and install something different. Trustworthiness is essential. When an operating system has a so called 'back door' (either intentionally or not), someone could extract whatever user data - like personal pictures or home movies - from your computer. And the worse thing: without you ever finding out. The operating system guards all the other software, and warns you when you install software from the internet. But itself, it doesn't have to ask for permission. Ever. It doesn't just have "access all areas": in fact, it pretty much runs the whole show.

With commercial software like Microsoft Windows or Mac OS X that you get delivered when you buy a computer, trust in what their closed operating system does will of course always be a serious leap of faith: as a user you essentially are given no choice. In proprietary systems you do not have the freedom to study the source code, or to control what really happens. So you either trust the vendor, or you'd better not use it. For an increasing amount of people, after the revelations from whistleblowers like Edward Snowden, that "leap of faith" is not so obvious anymore. They prefer to use free and open source operating systems like GNU Linux, FreeBSD and OpenBSD. These are technology commons: the people that wrote the software allow you to inspect the source code. Even more so, they give you the source code to do anything with it that you like. So you don't just blindly have to take their word for it and trust

them, you can take matters into your own hands.

But until now, there were some parts that would escape introspection. You would have to trust them, not because the people involved didn't want to share everything with you - but because they couldn't. When an operating system is loaded, you need to get the computer into a state from where it can manage itself. The necessary software is poetically called a "binary seed", because it is, well, a very very long string of bits from which everything grows. In fact, a few hundreds of millions of bits. And of course, that amount of information without any hints or cues as to how they interact are rather hard to grasp - and thus a potential point of risk.

What if we could get the computer into the right state through a different path? The GNU Mes project aims to replace the traditional "binary seed" by something orders of magnitude smaller. The really clever and innovative part is that they will add the more complex parts to a "second stage", which is being created from scratch by the project, in a human understandable programming language. This two stage approach allows to make all of computing more trustworthy, in a very controlled way - and will grant our future selves the ability to use computers without taking a leap of faith. If the project succeeds, it will make a very fundamental contribution to the security of the next generation internet. By making it work not just on desktop computer and servers, but also on mobile phones, tablets and lots of other devices the impact of the project increases.

Technical description

GNU Mes was created to address the security concerns that arise from bootstrapping an operating system using large, unauditable binary blobs, which is common practice for all software distributions. Mes is a Scheme interpreter written in a simple subset of C and a C compiler written in Scheme that comes with a small, bootstrappable C library. The final goal is to help create a full source bootstrap for any interested UNIX-like operating system. This funding will enable GNU Mes to work on the ARM platform.

Friendly Machines e.U. — **Visit https://NLnet.nl/project/GNUMes-arm**

NGI0 PET **BinarySeed** **Bootstrap** **C6** **Compiler** **EmbeddedSystems** **Minimisation** **OperatingSystem**

Thunderbird - native EteSync integration using TbSync

People and organisations use both free and paid online services to manage their private address books, calendars and tasks. These services allow them to back up their data and share the same

information across different devices - so they can add an appointment or new contact while they are on the mobile phone at the train station, or on the couch at home, and it magically emerges on their desktop calendar. Other tools allow our loved ones to know where we are at any given moment in time. Given how personal and confidential such information is, use of these convenient services can make users vulnerable to all kinds of abuse.

That risk is not necessary. Service providers can perform the core services (sharing and backup) just as well without any knowledge about user data. Given how normal encryption has become elsewhere on the internet, for instance in instant messaging, it is high time that we start applying it to the information we store about the people we meet, the places we go and the things we do. The overarching goal of the open source EteSync project is to enable users to end-to-end encrypt all of their information, and the expected outcome of this project is to make EteSync available for users of the popular open source email client Thunderbird - thus making it possible for many more users to switch off the unprotected storage and regain their privacy. The new extension will support integrated sync for contacts, calendars and tasks. This allows users to safely store their private data without having to be a computer science wizard.

Technical description

EteSync is a secure, end-to-end encrypted and privacy respecting sync solution for contacts, calendars and tasks. It protects user data by encrypting it and decrypting it on the end user device, meaning that the user does not have to trust the service provider. Etesync is being developed with support of NGI Zero. This project is adding native sync support for EteSync to the popular Thunderbird mail client (via the existing TbSync which is about to be integrated into Thunderbird) in order to drastically lower the entry threshold. This will allow even non skilled users to fully protect their data with end-to-end encryption. Setup will just involve (auto-)installing an add-on and entering credentials, and selecting which resources should be synchronized.

Visit https://NLnet.nl/project/EteSync-Thunderbird

NGI0 PET | Calendaring | ClientSideEncryption | Cryptography | DAV | E2EE | MobileApp | Sync

---

Wireguard Rust Implementation

VPNs (Virtual Private Networks) are common every-day tools, used by businesses, governments and private citizens alike to create secure overlay networks protected against adversaries controlling the underlying network architecture. Private citizens primarily use VPNs to enhance their privacy, by routing their traffic through a trusted intermediary they can hide their origin from any service they access on the internet and hide the contents of their traffic from any eavesdropper between them and their provider; whether it be the shady hotel wifi or an oppressive government. Businesses primarily use VPNs to connect remote sites as if they were situated on the same LAN (Local Area Network), enabling secure remote sharing of internal resources (e.g. printers) without exposing these directly to the internet. Additionally large internet service providers often emulate a secure local network between a number of physically

decentralized "cloud nodes" by connecting them using a VPN.

WireGuard is a new VPN protocol, which aims for security and speed by dramatically simplifying its design and configuration. WireGuard has traditionally been implemented as a Linux kernel module, however a userspace implementation in the Go programming language also brings WireGuard to Windows, Android, MacOS, iOS, and BSD variants. While working with the Go implementation we identified a number of points for improvement: improved control of memory consumption, control of sensitive data in memory, easier integration into other applications, as well as speed. All of these problems stems from language design of Go, notably the garbage collected nature of the language and the extensive runtime. This also prohibits any future effort to run the same code in userspace and Linux kernel space.

The users should expect improved speed, memory consumption, security (better control of secrets for "forward secrecy") and stability, wherever the userspace implementation is used. We also expect that the switch from Go to Rust might bring improved battery life on mobile platforms. For developers and potential contributes to the WireGuard project, the Rust implementation is also intended to ease integration into other software (notably the iOS and Android applications), as well as provide better compartmentalization of the different WireGuard components.

Technical description

WireGuard is an emerging open VPN protocol, WireGuard stands out from similar solutions, notably OpenVPN and IPSec, by being significantly simpler and hence easier to analyze and implement. WireGuard is currently available on Linux, Windows, MacOS,iOS, Android and BSD variants. WireGuard-rs will be an implementation of WireGuard in the Rust systems programming language. The WireGuard projects desire for a Rust userspace implementation, stems from the improved speed, memory consumption and safety guarantees offered by the Rust language, all of which are essential to the nature of the WireGuard project: a high performance, high security VPN. This implementation will be targeting userspace for Linux, Windows, MacOS and BSD variants.

WireGuard — **Visit https://NLnet.nl/project/Wireguard-Rust**

NGI0 PET **Cryptography** **Protocol** **Rust** **Security** **TypedLanguage** **Userland** **VPN**

Betrusted OS

be trusted [io]

As our lives get more digital every day, we use the internet to have important conversations - both personal and professionally. We also store and share more and more sensitive personal data on devices. On the internet you cannot just close the door to talk privately. So we need digital safe spaces and digital locks and vaults that are just as reliable and easy to use to store our secrets and mediate our communication.

Recently manufacturers have started to build so-called hardware enclaves or secure elements into their devices that function like a digital safe: even if someone is able to get some software installed into your

computer, phone or laptop, they should not be able to immediately access what is in the safe.

But of course, creating a secure space or making a digital safe in an environment you don't really control or understand is practically impossible. All the technical protection no longer matters when someone can invisibly take control or peer over your shoulder. Especially since you as a user can't see yourself what is happening on the inside of your digital house. A safe and a rogue application can and will look completely identical to a normal consumer, and there is simply no way to distinguish among them based on their outside appearance. Users install many unknown games and applications all the time ("install our app to start getting amazing discounts now!"), and forget that this is actually letting more or less random entities run unknown software on the phone that holds some of their most important information. And what if the operating system of your computer or phone itself has an unhealthy interest in your data or metadata, or is weakly protected to that others can just enter - similar to how unsafe it would feel if your landlord or the janitor is a peeping tom or a thief?

Betrusted is a dedicated open hardware device with the goal to create safe and more easily protected private channels for your communication. The Betrusted device is a complementary device that restricts itself to protecting the things that matter most, like your conversations and phone calls. It will also be able to hold passwords, digital versions of your passport (and other digital credentials and attributes), and whatever sensitive digital information you need to keep completely secure. In this project a custom, minimalist operating system will be written to run on the Betrusted open hardware. The overall approach is security through isolation and simplicity: you can never leave a backdoor open if you don't build a door in the first place. The end result will be a portable, dedicated physical vault isolated from everything else you do, and with a deliberately limited feature set which makes it so much harder to attack.

As a user you can verify everything from top to bottom. The entire design and development of the device is open to the public, from the software it runs down to the silicon that makes up its chips. A transparent, easy to use and secure digital safe that you can actually trust, with an configurable and easily understandable interface you want to use.

Technical description

Betrusted OS will underpin the Betrusted ecosystem, and will enable secure process isolation. It will be written a safe systems language - namely Rust - to ensure various components are free from common programming pitfalls and undefined behavior. Unlike modern operating systems that trade security for speed, the Betrusted OS will prioritize security and isolation over performance. For example, it will be a microkernel that utilizes message passing and services rather than a monolithic kernel with modules. Unlike other deeply-embedded operating systems, it will require an MMU, and support multiple threads per process. This will let us add features such as service integrity and signature verification at an application level.

Visit https://NLnet.nl/project/BetrustedOS

NGI0 PET **EmbeddedSystems** **Firmware** **MemorySafety**

Consumers that go shopping for a new cell phone or tablet these days, at the surface have quite a choice. Even the cheapest of mobile phones sold today, is surprisingly powerful compared to that of a couple of years ago. All that seems left for consumers to do is to match their own sense of style and of course budget. If they are really eager, they might compare a limited set of technical specifications: How long does the battery last? How big and bright is the screen? And do games and movies run smoothly? Most users tend to not even bother about that, eager to jump straight to the app stores filled with more applications than a human could feasibly install in their life. What more could a mere user want?

Somewhere in the back of our minds there may be lingering some larger, less happy thoughts. What about security and privacy? Who really is in control of our devices? It is not easy to connect the joyous occasion of our (often much anticipated) purchase of a really cool new gadget with societal resilience, our collective future well-being or any other of the larger economic effects of our individual choices...

In the early GSM era, there wasn't a single dominant operating system from a single vendor. The market was competitive and rather straightforward from todays perspective. Major efforts like Symbian (which ran on the very popular phones of erstwhile market leader Nokia, but also on those of Siemens, Alcatel, Bosch, Sharp, Sony Ericsson etc) were the result of a pragmatic collaboration on more or less equal footing of many manufacturers. These had a shared development responsibility, and equal opportunities. None of them knew how their users actually used the phones they created: that was the business of the customer.

The subsequent rise of the smartphone resulted in market disarray, because the dynamics of the new situation were so different. It wasn't so much a difference in technical quality that set the new masters of the universe apart, it was a complete change of the underlying business model and value proposition few people properly understood - if any.

The real-world cost of developing and maintaining the first generation of mobile platforms was non-trivial, and price competition in the devices was heavy. And then suddenly a no-visible-cost and feature-rich smartphone operating system appeared on the market. It wasn't produced by any of the current competitors or by an open consortium. The source was a single company that had heavily invested into this for strategic reasons. In parallel Apple was able to launch its own effort, take its slick iPod music player and its strong media presence and market visibility in the desktop space. Their premium iPhone line addressed the most luxurious part of the market - also with the help of Google. The CEO's of both companies even sat on each others boards, so the strategy was certainly aligned.

It was a perfect coup. Among the two of them they effectively levered the possibilities of the mobile smartphone platforms, media stores and restricted-access platform-owned app stores to take ownership and control of large parts of the software and content ecosystems at global scale. Traditional phone

manufacturers (many of which were European due to the success of the pioneering GSM standard) had historically been just selling a phone at competitive margins (with "no strings attached"). The whole economy of their operations and ecosystem of collaboration was effectively pushed aside by this audacious new strategy. The new Android operating system was funded not by the sale of the product itself, but by the promise of future user data gathering without real limits or much oversight - which had elsewhere proven to be able to create giant revenues. And unlike a desktop computer, a phone is nearly always on. It moves wherever the user goes, and thus it is always near. It has a camera, a microphone and lots of sensors. When users search for something, they use the default search bar which you control.

So effectively the new "smart" phone was primarily a vehicle for extensive data gathering about users, which could be resold and monetized later on. The manufacturers could get the operating system for free. The small margins that could be made on selling the software to they were negligible compared to the advantages later on. And of course at the time there was still a generation adoration of these "tech darlings" - press wrote lovingly about the "reality distortion field" around Apple's CEO Steve Jobs.

Right from the start this conceiled play was extremely profitable for both of them, allowing lots of subsequent investment - into their platforms, into the developer tools, into marketing and into legislative lobby. The "mobile first" strategy actually worked out better than anyone would have imagined, especially because the mobile phone operating system produced by Google turned out to be more than just a "loss leader". The market funnel of the free option it provided only became visible at the end. Technically advanced and more fair platforms appeared, but were unable to counter the "winner takes all" development in time. At present the vast majority of the phones are sold using one of only two operating systems: Android and iOS. In the absence of effective policy and legislative efforts to curb this unfortunate situation, that market dominance is a hard problem to solve at a technical level.

In our consumer bubble, we actively contributed and still contribute to this. The software stores of both platforms may offer consumers plenty of options at the application level. This seems quite healthy at first. But when you analyse the situation, it is far from how society should want this to be. This all starts with the fact that users do not have to manually install all applications. Apple has full control and puts its own software in pole position. Google is able to make the manufacturers do the same through contractual obligations. The result is the same: a strategic choice of end user applications is preinstalled alongside the platform, and effortlessly available to all users.

Many of us have meanwhile become used to these omnipresent "free" but closed "blockbuster" applications that ship alongside the dominant platforms. As we know from history, for instance through the famous European anticompetition cases against dominant technology companies taking control over web browsers, media players and portable runtimes (Java/C#), preinstalled applications have a huge competitive advantage. Not all users are as technically competent, and this creates enough inertia with consumers to keep manufacturers on a leash. The huge market share of platform 'defaults' like Android's default browser have a deep impact on the market, leaving little room for web developers to follow pretty much all what Google implements - even if they disagree or would actually like to follow proper web standards as produced by W3C. Who can afford for their website or web application to look worse on an operating system with the majority of market share?

Apple holds all the cards closely to its chest, and keeps full control. As long as it has Google as competitor, it feels secure of anti-competition measures. Their main strategy to even increase control is to buy suppliers, or make them sign exclusive contracts keeping others at bay. The defense strategy of Google is publishing most of Android source code. Manufacturers can and have tried to build alternative versions based on that. But in the market real-world control remains tightly with Google through the critical applications which need the "blockbuster" restrictively licensed apps and the larger infrastructure - both of which remain tightly closed. A certain percentage of users will always at some

point demand these "free" applications, while others cannot withstand the social lock-in and will actively push vendors to bow down. No small time manufacturer can afford to be out.

The platforms realise this powerful position very well, and are not afraid to lever it. Either a manufacturer is all-in, or all-out: it cannot selectively allow individual users to use blockbuster applications later on. This cut-throat dilemma has left the companies that make the actual phones little choice but to accept unattractive licensing conditions that restrict their freedom to innovate. And even if they do comply with all the demands including a non-disclosure agreement to seal their lips, their license can be withdraw at any time. In fact this may even happen due to geo-political pressure, as a very large Chinese manufacturer of Android found out to its great dismay in May 2019 when it was banned from future upgrades to Android. While part of this was retracted later, the fact is that such a thing could happen to any phone vendor using Android at any time.

The rigid control over the platform and the app stores was originally meant as a way to secure access to consumer data. These days, it is actually making an awful lot of money on its own. Consumers are paying a huge and very direct cost for the 'free platform' deal of the manufacturers. The dominant mobile platforms both charge developers up to an incredible 30% of their revenues (more than any VAT rate around the world!).

If your company wants to sell enough apps to make a living, you will want to use the default sales channel with the most users. This of course is the platform app store, which comes preinstalled on the prime spot. In fact, most users would not know how to install apps any other way, or are warned against that with scary messages. Selling through the app store means you have to pay up and at the same time obey all kinds of rules. The companies behind the mobile platforms themselves can at any time see an interesting market emerging. At that point there is a clear inequality of arms: if they want, the next update will put their own applications preinstalled on hundreds of millions of devices. This giving them a clear and unfair business advantage over anyone else in the market. Meanwhile developers ironically pay for the privilege of being allowed to exclusively develop for the platform concerned, and sell the outcome in the default (and most restrictive) app store. The platform almost certainly has a higher more profit margin from the average developer, even if it is a direct competitor. But what can developers do? Their investment into the software they wrote is hard-wired to the initial choice of platform...?

Non-trivial applications that run on one mobile platform do not run on another, and require additional effort to write in a way where they can. This invisible 'cost of diversity' to the larger ecosystem of creators (which is orders of magnitude bigger) contributed significantly to the "winner takes all" scenario at platform level. When the European Commission orders some app to be developed for citizens to access its services, crowdsource data gathering or inform them of passenger rights, it does not care about creating something for the users of the innovative Finnish mobile platform Sailfish from Jolla - or in fact anyone else. If you look at the apps officially published by the European Commission on the app stores, you will not find any app for any European mobile platform ever published there. The same 'selfish' short term considerations will of course be made even more frequently by smaller actors with less deeper pockets, like independent publishers. As a result the market will make the largest platforms larger, and will completely ignore the rest.

In the new mobile world we live in now, control as a user is limited to the very surface of things. Significant privacy and security issues start directly below that surface. You don't really know what the platform actually does while executing apps, and more importantly, who sees your data - or if you are a business, looks at the data of your customers. When you use one of the hundreds of thousands of existing apps and games, you only see the service they provide. But you can't inspect or even see what more they take. What does an app do exactly when you click on the pretty icon? This is very much unlike for instance interacting with a web page, which is fully transparent. As it turns out, mobile apps do lots of

things users do not know about, and would not agree with if they did. In some cases literally hundreds of companies have been known to get access to data on the phone.

A consumer-friendly platform should empower the user to notice and take action, or even make it technically impossible. However, the companies that produce the operating systems seem to have other interests. Have you ever wondered why everyone tells you your desktop computer needs a firewall and you are allowed full control to see everything happen. Now stop and think about why your cell phone does not have the very same level of firewall capabilities, but only very much simplified and less capable?

So what can we as a society do in the face of such a complex situation of market failure, anti-competitive practices, perverse incentives and general confusion? How do we give control back to the users? How do we create equal opportunities for European phone manufacturers? How do we stop the unfair "platform tax" on app developers, stimulating employment and startups?

One reasonable direction is to try and lay the ground work for creating viable alternative platforms. Such a fundamental approach is necessary in order to end these extractive practices and the resulting lack of consumer freedom. Smart phones are really just small computers. This means we can build upon plenty of meanwhile mature building blocks and technical work done over decades. In fact, both Android and iOS followed the same path. They were not created from scratch, but based on existing open source projects for desktop and server operating systems. There is nothing magical, it is just engineering work. This is what this project contributes to: it will lever joint work by Nokia engineers and the free and open source software community on the Maemo platform to create an independent alternative mobile OS. Of course there is significant work needed to make such a project relevant to end users, but the absence of purely commercial drivers allows this project a worthy mission: to provide a secure and modern mobile operating system that consists only of free software, obeys and respects the users' privacy and digital rights.

Technical description

Maemo Leste aims to provide a free and open source Maemo experience on mobile phones and tablets. It is an effort to create a true FOSS mobile operating system for the FOSS community. Maemo Leste is based on GNU/Linux, and specifically - Devuan GNU/Linux. The goal is to provide a secure and modern mobile operating system that consists only of free software, obeys and respects the users' privacy and digital rights. The project also works closely with projects that aim to produce hardware that Maemo Leste and other community mobile operating systems could run on. The operating system itself takes much of its design and core components from the Nokia-developed Maemo Fremantle, while replacing any closed source software with open source software.

Visit https://NLnet.nl/project/MaemoLeste

NGI0 PET HardwareDrivers MobileOS Reproducibility

Modern computers are fast. They process millions if not billions of instructions per second. Whenever you are waiting for a computer to do something, it therefore likely is either of two cases: data has to be fetched from far away and it takes time to transport the volume of data needed, or you are waiting for a database looking up some (combination) of data. Often it will be actually be both, because much of the data is stored remotely and much of that data will be in a database as well.

Historically databases were huge, and required an entire mainframe to run. These days, databases are small and fit in pretty much everywhere. This allows even the smallest applications to involve a database, whether it is for storing configuration data, logs or user data. Whether it is an operating system, a browser, a router, a game or even an entire programming language like Python - embedded databases are everywhere.

That makes the security and privacy of these embedded components an interesting and important topic. LumoSQL wants to innovate in the embedded database space, by bringing together best of breed components of existing databases together in a single powerful application - and adding useful features currently missing such as encryption, and networked usage. The idea is to be a drop-in replacement for popular embedded databases such as SQLite.

Technical description

The most widely-used database (SQLite) is not as reliable as it could be, and is missing essential features like encryption and safe usage in networked environments. Billions of people unknowingly depend on SQLite in their applications for critical tasks throughout the day, and this embedded database is used in many internet applications - including in some core internet and technology infrastructure. This project wants to create a viable alternative ('rip and replace'), using the battle tested LMDB produced by the LDAP community. This effort allow to address a number of other shortcomings, and make many applications more trustworthy and by means of adding cryptography also more private. Given the wide range of use cases and heavy operational demands of this class of embedded databases, a serious effort is needed to execute this plan in a way where users can massively switch. The project will extensively test, and will validate its efforts with a number of critical applications.

Visit https://NLnet.nl/project/LumoSQL

NGI0 PET CopyOnWrite Cryptography Database EmbeddedSystems Multicast

be trusted [io]

As our lives get more digital every day, we use the internet to have important conversations - both personal and professionally. We also store and share more and more sensitive personal data on devices. On the internet you cannot just close the door to talk privately. So we need digital safe spaces and digital locks and vaults that are just as reliable and easy to use to store our secrets and mediate our communication.

Recently manufacturers have started to build so-called hardware enclaves or secure elements into their devices that function like a digital safe: even if someone is able to get some software installed into your computer, phone or laptop, they should not be able to immediately access what is in the safe.

But of course, creating a secure space or making a digital safe in an environment you don't really control or understand is practically impossible. All the technical protection no longer matters when someone can invisibly take control or peer over your shoulder. Especially since you as a user can't see yourself what is happening on the inside of your digital house. A safe and a rogue application can and will look completely identical to a normal consumer, and there is simply no way to distinguish among them based on their outside appearance. Users install many unknown games and applications all the time ("install our app to start getting amazing discounts now!"), and forget that this is actually letting more or less random entities run unknown software on the phone that holds some of their most important information. And what if the operating system of your computer or phone itself has an unhealthy interest in your data or metadata, or is weakly protected to that others can just enter - similar to how unsafe it would feel if your landlord or the janitor is a peeping tom or a thief?

Betrusted is a dedicated open hardware device with the goal to create safe and more easily protected private channels for your communication. The Betrusted device is a complementary device that restricts itself to protecting the things that matter most, like your conversations and phone calls. It will also be able to hold passwords, digital versions of your passport (and other digital credentials and attributes), and whatever sensitive digital information you need to keep completely secure.

In this project a virtual space will be set up to develop and test software. The overall approach is security through isolation and simplicity: you can never leave a backdoor open if you don't build a door in the first place. The end result will be a portable, dedicated physical vault isolated from everything else you do, and with a deliberately limited feature set which makes it so much harder to attack. As a user you can verify everything from top to bottom. The entire design and development of the device is open to the public, from the software it runs down to the silicon that makes up its chips.

Technical description

The Betrusted software project utilizes the strongly typed Rust programming language to build the first applications and libraries for the open hardware Betrusted.io project. Betrusted is pioneering a new class of open hardware communications device, with a grant by NGI Zero. The project will set up a virtual environment for betrusted (e.g. QEMU / RISC-V) in order to develop and test software as close to target as possible and unlock community collaboration and contributions. The second main task in the project is to write a Matrix protocol command line client in order to analyze the memory characteristics in the highly constrained betrusted environment. The additional time is to be allocated to development support for the Bestrusted OS, develop glue layers and verify necessary interfaces for applications,

provide unit/integration tests and develop (test) applications for it.

Visit https://NLnet.nl/project/BetrustedSoftware

NGI0 PET **EmbeddedSystems** **Messaging** **UI** **UX**

Fix the Pitch Black Attack in Freenet friend-to-friend routing

Privacy is a matter of control. When you want to protect your privacy, it does not mean you never tell anyone anything, it means you want to be in control of who you share your personal information with. On the internet a lot of control is taken away from you. The technology that lets you connect to networks all around the world and find information anywhere it is stored is built around identification, both of its users and the virtual places they visit. Unfortunately, many crucial networking standards and protocols were not designed with user privacy in mind, let alone giving them any sense of control over who can see what they do online. This vacuum has been filled with all sorts of tracking and tracing schemes that can make detailed profiles of people, which can then be (mis)used for commercial or even criminal gain.

Freenet is one of the oldest online platforms that protects user privacy and free speech by offering as much anonymity as possible. The users of Freenet actually make up the network where instead of central servers, each peer contributes bandwidth and some hard drive space to store content. Together this creates a decentralized data store that makes it possible to create censorship-resistant websites, chat forums and search within the network. Users can go even further to protect their privacy and only trust peers they know they can trust. This makes it even harder to block usage of Freenet, which is especially valuable for people living with governmental censorship.

Because a peer-to-peer-network like Freenet relies on its users, attacking peers can potentially affect the whole platform. For example, someone could create a large number of fake users that together try to influence the network in a certain way. This project aims to implement a defense measure against these kinds of attacks and ensure that the network remains stable and trustworthy. The Next Generation Internet needs places like Freenet that protect free speech, resist censorship and allow users unobstructed communication, which this project will make sure of.

Technical description

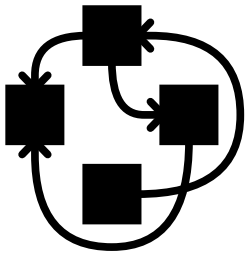
Freenet is a peer-to-peer platform with academic roots, offering censorship-resistant publication and privacy by design. It uses a decentralized distributed data store to store and forward information of its users, and is one of the oldest privacy related infrastructures - having been in continuous development for two decades, and predating the alpha version of TOR with several years. This project solves a published theoretical denial-of-service attack on the friend-to-friend structure of its routing, which has

been a looming threat since it was discovered a number of years ago.

Visit https://NLnet.nl/project/Freenet-Routing

NGI0 PET **Anonymity** **P2P** **Routing**

Implement sound support in the Hurd



When you start up your computer, you will probably think twice before you download some random piece of software from the internet and run it. You know that doing so could allow unwelcome guests to your computer and your data. Your computer might even end up in a bot net. So when you see some nice piece of software, you will ask yourself the question: can I really trust the software? Perhaps you will check the origin it comes from. Better safe than sorry.

Did you miss checking something, though? What about the software that is already on your computer before you started? A computer is not much use without an operating system. While most computers are sold with an operating system, actually you have the choice to remove that and install something different. Have you thought about the trustworthiness of that fundamental piece of software - your most fundamental travel companion on the wild west of the internet? Trustworthiness is essential. When an operating system has a so called 'back door' (either intentionally or not), someone could extract whatever user data - like personal pictures or home movies - from your computer. And the worse thing: without you ever finding out. The operating system guards all the other software, and warns you when you install software from the internet. But itself, it doesn't have to ask for permission. Ever. It doesn't just have "access all areas": in fact, it runs the whole show.

With commercial software like Microsoft Windows or Mac OS X that you get delivered when you buy a computer, trust in what their closed operating system does will of course always be a leap of faith: as a user you essentially are given no choice. In proprietary systems you do not have the freedom to study the source code, or to control what really happens. So you either trust the vendor, or you'd better not use it. For an increasing amount of people, after the revelations from whistleblowers like Edward Snowden, that "leap of faith" is not so obvious anymore. They prefer to use free and open source operating systems like GNU Linux, FreeBSD and OpenBSD. These are technology commons: the people that wrote the software allow you to inspect the source code. Even more so, they give you the source code to do anything with it that you like. So you don't just blindly have to take their word for it and trust them, you can take matters into your own hands.

The GNU Hurd is an initiative to take transparency over your operating system and device even further. The project replaces the widely used Unix kernel, which is the fundamental program that controls everything in the operating system. The Hurd aims to separate tasks and responsibilities as much as possible and give users complete control over for example security. This particular project aims to develop an audio system that is as modular as possible: users can easily remove and add access to audio-

hardware in their operating system, instead of blindly trusting some audio application. These functionalities bring the GNU Hurd one step closer to realizing the last important software component for a complete free and open operating system.

Technical description

The GNU Hurd is a light weight kernel (the central part of an operating system) on top of the Mach microkernel, with full POSIX compatibility. The mission of the Hurd project is: to create a general-purpose kernel suitable for the GNU operating system, which is viable for everyday use, and gives users and programs as much control over their computing environment as possible. Hurd provides security capabilities like adding access to services for programs at runtime when and only while they need it, and to enable easy low-level development - like replacing a file system during runtime and real-time kernel debugging as if it were a normal program. This project adds an important feature to GNU Hurd: an audio-system with fine-grained access management to physical hardware.

Visit https://NLnet.nl/project/Hurd-Audio

NGI0 PET **GNU** **Kernel** **Microkernel**

Video chat privacy

One of the things people enjoy the most about the internet, is that it enables them to talk to others remotely almost without limit. Internet allows anyone to keep closely connected with friends and family, and help their kids solve a math problem while they are at work. People collaborate with their colleagues from the couch of their living room, the cafe where they enjoy lunch or on their cell phone on the bus to the gym. Businesses can easily service their customers where this is most convenient to them, without having to travel themselves. This is so convenient, that some businesses have already moved entirely online. Internet communication has become the nerve center of whole neighbourhoods, where people watch over the possessions of their neighbours while these are away for work or leisure.

However, users have a hard time to understand how privacy is impacted if they use the wrong technology. Because internet works almost everywhere, the natural privacy protection of the walls of a house, a school or an office is gone. For example, when you make a video call, your webcam or phone camera captures a lot more than just you talking, for example the people around you, the books on your shelf or the street outside. A lot of this information can be used to uniquely identify you or to find your location, which you may not always be aware of. Because high definition cameras are embedded in more and more devices everywhere around us, we need more control over what these digital eyes actually record about us.

Instead of only being able to switch your webcam on or off, this project will develop technology that lets you remove or anonymize the background while you are actually making a video call. This will give users more tools to protect their visual privacy and fight back against all sorts of sophisticated tracking schemes.

Technical description

Making video calls can be very invasive to privacy: the camera does not only capture the face and posture of the person talking, but will in fact capture the entire environment in glorious high definition - from the books in your bookshelf to family members or laundry rack behind you. This information is of no interest to the other end, but with a camera you have little choice: once you slide open the camera cover, it takes everything within the field of view and broadcasts it to the other side. This project aims to use advanced AI technology to edit the video feed in real-time, and apply various privacy enhancements such as removal of backgrounds.

Visit https://NLnet.nl/project/VideoChatPrivacy

NGI0 PET **AI** **Videoconferencing** **VisualPrivacy**

B a l t h a z a r

BALTHAZAR

Hundreds of millions of people depend on a laptop computer to use the internet wherever they go. You can easily take a laptop to a library or coffee shop, or over to a friend. Despite ergonomic limitations many people use their laptop at home or in the office as well - having multiple devices around is expensive and remains a bit cumbersome. And compared to a desktop computer a portable one takes up considerably less space, which is certainly important if you share a household or office with others.

Unlike a desktop computer, exchanging broken parts of a laptop or upgrading is something that normal end users are typically not equiped to do. This means that when a laptop computer breaks or needs an upgrade, we face significant cost - and also run a non-trivial risk of abuse of our privacy. After all, some of our most precious data is held on our computers. While in general one should really be able to trust our technology suppliers, it is also well known that there has been at least one major case where repair staff was actually instructed and paid to nose around broken laptops for customer data that could be resold. In other cases depraved individuals took compromising private pictures from devices (often from unknowing victims) and shared these with others. This kind of liability pushes people to just buy a new device, instead of risking unknown cost and loss of privacy.

But why is this still the case when laptops are technically quite a mature product, no longer pushing the boundaries of physics and electronics (compared to a smart phone a laptop has plenty of space). And we live in an era where we know we have to be more careful about using natural resources than ever? Why can't we make laptops that you can just easily fix or upgrade yourself? Where one component that is broken, does not mean the end of a device. So that your wish to add a snazzier camera or some extra power does not mean that you need to buy a new one.

There is even more to it. How trustworthy and transparent are devices currently anyway? Data protection authorities have warned against privacy invasive behaviour of software that has been reported to run on over a billion devices worldwide - including (still) many governments. And there is something deeply uncomfortable about your laptop vendor mentioning in the manual that should you want to reinstall a fresh copy of the operating system, that you don't need to give in any activation code

anymore because this information is already burned into your motherboard.

The Balthazar laptop wants to give the world a new type of personal computing device, one that is extensible and environment friendly. A trustworthy laptop that allows its users (including those with low income) to feel secure, safe and comfortable using it. For the children of all ages.

Technical description

Project's ambition is to design and deliver an innovative and technically advanced open hardware (RISC-V/ISA) based, European made, inexpensive, FOSS laptop as a personal computing device, containing on board all desirable (FOSS compliant) hardware and software features and functionalities needed to prevent any 3rd party intrusion into the system. It adds physical safety features currently not available in the market such as hot-swappable CPU, hardwired switches for e.g. camera and audio devices, and a quickly removable encrypted hard drive and peripherals. A goal of Balthazar is to enable and educate end users to be private, safe and careful with their own data, and that of others. Another goal is to make computing more sustainable and reach eco-friendly footprint, by empowering users to take up their 'right to repair', through a modular laptop that allows components to be easily exchanged and upgraded - up to the CPU itself. The goal is to lead by example and gently lead other hardware manufacturers to become fully open and transparent. And create an educational platform, as well as an advanced computing device where its users (including those with low income) to feel secure, safe and comfortable using it. For the children of all ages.

Visit https://NLnet.nl/project/Balthazar

NGI0 PET **Laptop** **OpenHardware**

F u n k w h a l e

Funkwhale

How do you discover new music? Do you rely on the good taste of a friend, are you part of a fan community, do you simply tab through your favorite genre at your local music shop? On the internet, you can learn about exciting new artists in a number of ways, but today most users rely on their audio service of choice to discover new tunes. Unfortunately only a few major music services control most of the music streaming market, leaving users as well as artists with little control over how their music gets discovered. When you want to discover exciting new bands and artists using these services, usually you will be served the best-selling name of the ay, burying more independent artists under top-ten hype lists. What's more, these major streaming providers do not just mistreat artists but also their users, tracking and profiling their activity to coerce them even further into listening to only a sliver of what is actually available on the platform.

Major streaming services force users and artists into a commercial straightjacket that is not privacy-friendly and that prohibits free cultural exploration. We deserve alternatives, like Pixelfed and Mastodon are decentralized alternatives to Instagram and Twitter, platforms that commercialize and centralize our online social interaction.

Funkwhale is an effort to create such a decentralized alternative to services like Spotify, Deezer, Google

Music and others. Like Mastodon and Pixelfed, Funkwhale is free and open source software, can be self-hosted and lets users decide for themselves what they want to share privately or publicly. Users can either create a Funkwhale-server (called a pod, like a pod of whales) and upload their own music library to share with others, discover new content without restrictions or coercion, and stream for their device of choice. And because Funkwhale is also powered by the ActivityPub-technology, users can connect to instances of Mastodon, Pixelfed and many other decentralized social networks. This project will improve how artists can publish their music using Funkwhale, make music discovery richer for users and advance search mechanisms. Together, these new features will help to make music discovery the way it should be: free, fun and unpredictable.

Technical description

Funkwhale is a free, decentralized and open-source audio streaming and sharing platform, built on top of the ActivityPub protocol. It enables users to create communities of interest around music and audio content in general, listen to their private music library or distribute their own productions on the network. Each Funkwhale pod, or server, can communicate with other pods to exchange audio content, metadata or for user interactions. In this project, Funkwhale will improve the publication experience for creators, release its first stable version, improve content discovery inside the platform through better sharing and search mechanisms. We will also continue research and development for Retribute, a community wealth sharing platform meant to support creators on Funkwhale or any other platform.

The Funkwhale Collective — **Visit https://NLnet.nl/project/Funkwhale**

NGI0 Discovery ActivityPub Audio Creators Discovery Federation ServerApp Sharing

---

X W i k i

Online search and discovery reaches further than the search bar in your browser. There are all sorts of places where people can come together to share knowledge and store information for others to sift through, looking for a particular name, email address or useful snippet. One of these places is a wiki, of which the free community-backed encyclopedia Wikipedia is the most prominent example. On a wiki, people can effectively organize their own knowledge base, decide how their information is organized and linked, making it easily findable. Wiki's are used by organizations, governments and businesses everywhere, sometimes storing data essential for everyday operations, or with sensitive credentials. Some cities have their own wiki's, containing rich localized content useful for inhabitants, shop owners and tourists.

To make a wiki work, you need active and involved users. Xwiki is a platform offering free and open source wiki software for organizations to create their own knowledge base, extending and modifying how the wiki works as they please. Extensibility is essential, which is why Xwiki in this project wants to connect itself to the larger federation of decentralized social networks, also known as the federated universe or fediverse. Connecting to content and interacting with users of for example Mastodon,

Nextcloud and PeerTube makes Xwiki an even richer wiki platform, allowing all sorts of useful extensions of your knowledge base, website, or collaborative intranet using Xwiki. And because the project is built on open source software and protocols, other communities can learn from these efforts to tie all sorts of public and hidden treasure troves of knowledge together precisely how they want to, while staying in control over their social data and the information they want to share online.

Technical description

XWiki is a modern and extensible open source wiki platform. Up until now, XWiki had been focusing on providing the best collaboration experience and features to its users. We're now taking this to the next level by having XWiki be part of the larger federation of collaboration and social software (a.k.a. fediverse), thus allowing users to collaborate externally. XWiki is embracing the W3C ActivityPub specification. Specifically we're implementing the server part of the specification, to be able to both view activity and content happening in external services inside XWiki itself and to make XWiki's activity and content available from these other services too. A specific but crucial use case, is to allow content collaboration between different XWiki servers, sharing content and activity.

XWiki SAS — **Visit https://NLnet.nl/project/WikiActivityPub**

NGI0 Discovery ActivityPub Collaboration Federation NGI0 Wiki

---

Web Annotation

Undoubtedly you have come across some article, wiki entry, video or comment that made you want to react immediately. Maybe someone was looking for some obscure artist you know everything about, a question was answered incorrectly, or an article left out some essential bit of information. Or you simply did not have the time to go through everything and wanted to leave a reminder for yourself, 'must watch this when home'. But to leave a comment or pin a note to a page, you need to login to some service you don't know, fill in your name and email address, and instead of all the hassle you simply leave it alone and close the page.

What if creating and exchanging annotations on the web would be as simple as writing down a note on a piece of paper? Instead of using all sorts of different tools and apps that do not interoperate, there is an organized effort of standardization communities and open source developers to make open, flexible and extendable annotation technology.

This project will use a new standard format for annotations for new tools and solutions that can recognize and enrich each other, ultimately creating a fluid and friction-less environment for web annotation. Users will be able to easily add comments or notes to a web page, read other people's annotations of their choice and search through useful items, all from the comfort of their browser. Web annotation can make search and discovery richer and more intuitive, but only when the technology used is sufficiently interoperable and open that for a user, it is as simple as browsing, or as scribbling their thoughts on a page.

Technical description

The idea of web annotation is to support the creation and exchange of annotations on any visited page; thereby enabling people to make, share, and discover corrections, rebuttals, side-notes, or other contextually relevant resources. Using the W3C's Web Annotation standard, and contributing to the incubating Apache Annotator project, this project works on modules and tools that facilitate a diverse ecosystem of interoperable annotation systems.

Visit https://NLnet.nl/project/WebAnnotation

NGI0 Discovery **Annotation** **W3C**

ForgeFed

To make new software, developers need to know what the latest version of a program is, what issues are still unsolved who will do what with that exciting new upgrade coming up next week. These environments are essential to software developing companies worldwide, basically containing all the work and priorities of every developer in the room. This is even more important for open source communities, where people contribute from all over the world and need to come together somewhere to make a plan, set some goals and then: do the work.

Developers can choose from all sorts of collaborative environments, also known as code forges. This project wants to extend the version control, issue tracking and project management of these code forges and create a more social, interconnected place for collaboration. This will be done by connecting these environments to the rich universe of decentralized social networking and publishing, also known as the fediverse (federated universe). Instances of for example the self-hosted social network service Mastodon and the federated photo sharing platform Pixelfed will be able to connect to the collaborative environment of you and your community, enriching the work flow with all sorts of content and comments.

Mind you, tracking issues, maintaining version control and managing a project is not only valuable for software developers: how about collaboratively publishing a book, school material, a manual? This project will break through the walls of traditional code forges and deliver technology you can use to create your own, self-hosted creative space. You can manage your project and connect to inspiring content, discussions and projects coming from other gathering places. No tracking of users, no rules set for how to collaborate, free search and discovery across a wealth of independent and connected communities.

Technical description

When you are searching for new software to use, you will have to visit many different software forges - like Gitlab, Codeberg or Sourcehut. There isn't really a tool to search for anything across the boundaries of these different software forges.

ForgeFed aims to define a vocabulary and a protocol for decentralized communication and federation of websites used for hosting and collaboration on version control repositories, issue tracking and project management. Typical such websites are code forges such as GitLab and Gitea instances (and centralized services like github), but the idea also applies to applications like collaborative civic planning, publishing of creative writing (such as prose and poetry) and more. ForgeFed is to be designed as an extension of ActivityPub, and web apps implementing it would be joining the Fediverse. The world of repo and project hosting would switch from the centralized model of github (and the lonely disconnected websites running GitLab or Gitea etc.) into a network of federating websites, creating a global decentralized community. The project will publish a set of specifications and guides for implementing the federation protocol, and to work with existing projects and communities to refine and finalize the specifications and implement ForgeFed federation.

Visit https://NLnet.nl/project/ForgeFed

NGI0 Discovery **Collaboration** **DistributedVersioning** **Federation** **SoftwareDevelopment**

FairSync

Climate change is the most crucial (and divisive) issue of our time. The world needs to switch from fossil fuels to renewable sources of energy as soon as possible, not to avoid the consequences of a rising worldwide temperature, but to limit the irrevocable damage it will cause to our environment, food and vital infrastructures as much as we can.

While the science is crystal clear on the consequences of climate change and the necessity for progress, actually changing politics, business and our own behavior is easier said than done. Around the world activists are fighting for progress, all for the same goals, but not always together. This project aims to organize and connect the actions of all these independent groups. Using the same technology that enables decentralized social networking like Mastodon, the activity hubs, events and actions can be synchronized, bringing these independent climate activists closer together into a truly global, formidable cry for sustainable change.

Technical description

How can we make it possible to search across different maps and lists of events maintained by different organisations? By connecting them, of course! FairSync develops and collects best practices to

synchronize maps and events and to federate messengers and identities active in the global movement for sustainability. System integrators are faced with fast evolving APIs and protocols when they try to discover and connect systems and make search more easy.

We will work on master-master replication frameworks of metadata enriched data sets and test with platform providers for sustainability affairs. One approach is the "lazy master scheme": a common update propagation strategy where changes on a primary copy are first committed at the master node, afterwards the secondary copy is updated in a separate transaction at slave nodes.

We will try to advance such immediate update propagation in this project using protocols such as ActivityPub or the InCommon API. Federation of identities will be managed with SAML or oAuth2 protocols with fairlogin as a common identity provider.

Fairkom — **Visit https://NLnet.nl/project/FairSync**

NGI0 Discovery **Geo-tagging** **Maps** **News** **Synchronisation**

Interpeer

Collaboratively writing a document together in real-time with others is still a bit magic. Someone else, perhaps on the other side of the planet, is typing something. And within a fraction of a second, the text magically appears on your screen. This amazing technology is the ideal companion for say an online meeting - everyone can contribute, and correct any flawed minutes without much effort. For this kind of collaboration in real-time, there is a limited set of options in the market you can use. Most available services in the market like Google Docs, Microsoft Office or LibreOffice Online.

Most online collaborative services share one very undesirable characteristic: you need to fully trust the company running the service you use. Whomever has access to the servers used to connect everyone together, can read everything you have written - and deleted. That means that if you need to work on something confidential like an important contract, you may want to reconsider using the service. Especially if you write about sensitive topics like corruption, money laundering or state surveillance this open backend you cannot control is a really significant problem.

Peer-to-peer collaboration is a way for internet users to connect and work together directly, without the need for a central authority or in-between layer. Search and discovery in this way can be crowd-sourced, instead of organized by one central party (a search engine) that is more vulnerable to attack and misuse. Together, peers can publish data, subscribe to other people's messages and documents, recommend and disseminate information and news and tag correct and informed articles and stories, that can then be searched by others. The group filters what data and information should be spread wide and far and what should be forgotten, not a third party (i.e. the search engine provider) that will not give access to its search algorithm to protect their commercial interests.

This project wants to prepare peer-to-peer collaboration for current day user needs and future challenges, tackling a very tough use case: high-resolution video broadcasting using everyday consumer connections. This is a tough use case because to stream this type of video, you need to have very little

delay (or latency) and a high rate of data transfer (or bandwidth) in your connection, which is hard to pull off in a peer-to-peer-network. Consider this a test of strength that only a truly advanced network solution can achieve, which would make this technology incredibly useful for real-time peer-to-peer online collaboration.

Technical description

The Interpeer Project's purpose is to research and develop novel peer-to-peer technologies for open and distributed software architectures. The goal is to enable serverless modes of operation for collaborative software with rich feature sets equal to or surpassing centralized client-server architectures. For that reason, the initial focus lies on facilitating the extreme end of the use case spectrum with very low latency and high bandwidth requirements, as exemplified by peer-to-peer video communications in quality as close to 4k resolution as possible. When that initial goal is reached, the project focus will shift to other collaboriative applications of the technology.

Visit https://NLnet.nl/project/Interpeer

NGI0 Discovery Application IETF Library MultiSource P2P Protocol Streaming Video

---

Software Heritage

How do you preserve a piece of software for prosperity? You might have a box of floppy disks in your attic somewhere, a treasure trove of games and programs you fired up daily in your childhood. Physical memory can be a great way to store digital data, but do you know anyone that still has a computer with a floppy drive? Better yet, does your laptop or computer even have a CD drive? The internet can provide a better data archive, but this still requires maintenance: everything that is online needs to be physically stored somewhere and once data is lost, it is lost forever. So how do you organize archiving data and software?

Software Heritage is an organized effort to preserve all the software ever written. The programs we use everyday say something about how we interact with our devices and connect with each other through technology. What can our software do, how can we use it, understand it, make it work for us? Preserving these programs is a constant and challenging mission, as software hosting never is a given.

For this project the software preserving community will focus on making sure certain open source software is saved from the digital black hole, as a particular code version control system will be discontinued soon. Software Heritage will make sure these programs are preserved so we can learn from them, to make even better and more human software in the future.

Technical description

Software Heritage is a non profit, multi-stakeholder initiative with the stated goal to collect, preserve and share the source code of all software ever written, ensuring that current and future generations may discover its precious embedded knowledge. This ambitious mission requires to proactively harvest from a myriad source code hosting platforms over the internet, each one having its own protocol, and coping with a variety of version control systems, each one having its own data model. This project will amongst other help ingest the content of over 250000 open source software projects that use the Mercurial version control system that will be removed from the Bitbucket code hosting platform in June 2020.

Software Heritage — **Visit https://NLnet.nl/project/SoftwareHeritage**

NGI0 Discovery **Archiving** **Crawling** **CulturalHeritage** **Foundation** **Indexing**

Personal Food Facts

When you go do your groceries, how do you decide what food you will buy? Most people rely on a mix of familiarity, habit and package texts to find out what products fit in their diet, what is missing in their cupboard or simply what they want to try out. But what about the millions of people with allergies or strict health-related diets? Doing your groceries becomes a more difficult when you need to be sure you are not buying anything that might trigger potentially very dangerous allergic reactions.

People turn to the internet for information about health and food, but are often confronted with either conflicting opinions or commercial apps and databases that are after their personal information. What you eat tells a lot about who you are, like a kosher or halal-diet clearly indicates your religion affiliation. Finding out what products you can and cannot buy, should not mean you have to disclose very personal information with all sorts of untrusted third parties. Unfortunately, this can happen when you log in to a website of a supermarket chain and filter their offer based on your personal diet.

Because our diet and our health is our own information, we deserve open and public information about food we can search through freely. Open Food Facts is an effort to collaboratively build such a database and currently contains open data on 1 million food products from around the world, independent from the food industry and commercial interests. This project will use this information to give users personalized search results, for example filtering out products that do not fit within a specific diet. Users are free to search this information on their own and do not have to worry about anyone snooping in or somehow messing with their search results for commercial gain.

Technical description

Open Food Facts is a collaborative database containing data on 1 million food products from around the world, in open data. This project will allow users of our website, mobile app and our 100+ mobile apps ecosystem, to get personalized search results (food products that match their personal preferences and diet restrictions based on ingredients, allergens, nutritional quality, vegan and vegetarian products, kosher and halal foods etc.) without sacrificing their privacy and having to send those preferences to us.

Open Food Facts — **Visit https://NLnet.nl/project/OpenFoodFacts**

NGI0 Discovery **Client-side** **Crowdsourcing** **EthicalFilter** **FoodIngredients** **Foundation** **MobileApp** **OpenData** **ServerApp**

Tantum Search

Search and discovery are some of the most important and essential use cases of the internet. When you are in school and need to give a presentation or write a paper, when you are looking for a job, trying to promote your business or finding relevant commercial or public services you need, most of the time you will turn to the internet and more importantly the search bar in your browser to find answers. Searching information and making sure your name, company or idea can be discovered is crucial for users, but they actually have little control over this. Search engines decide what results you see, how your website can be discovered and what information is logged about your searches. What filters and algorithms are used is unclear for users. They can only follow the rules laid out for them, instead of deciding on their own what, where and how to find the information they are looking for.

One of the ways most commercial search engines decide what results you see, is something called link popularity. This is ma metric that indicates how many other other links point toward a particular website. Sites and domains that everyone refers to, usually end up at the top of your search results. Of course this does not mean that this website best answers your question, has the most informative content, or is even correct at all. And because this process of higher link popularity, higher search results ranking reinforces itself, all this mechanism does is narrow your search results and give you less useful or insightful information over time.

This project gives users and information providers back the control they deserve over online search and discovery, putting quality over popularity. Instead of counting the number of links, search is focused on the actual question of a user, querying on the words themselves, their context and location-based relevance. Extra ranking options allow you to search for things like eco-friendliness, giving you a broader range of search tools and perhaps a whole new look on the services and products you were looking for.

Technical description

Tantum Search's goal is to present information in a fair and transparent context for the users. The platform lets users make an inventory of any information using schema.org schemas (like video, audio, paintings, ebooks, events, goods, services) and allows users to search through these entries on three

axes: word, contextual and geo reference resolution. Providers of information can easily and without great effort add their information to the platform and make it available online – the platform automatically creates an interactive page which will be search engine optimized and users get free and unbiased access to search for goods and services. The ranking focuses on the search query and less on link popularity. Thus, 'internet giants' are not necessarily listed at the top due to their popularity and in addition, the ranking algorithm will be transparently released as open source so the community can optimize it.

Tantum — **Visit https://NLnet.nl/project/TantumSearch**

NGI0 Discovery **Context** **EthicalFilter** **Ranking** **Search**

---

eduVPN app

Visit https://NLnet.nl/project/eduVPN-app

VPN Fund **Architecture** **Multiprotocol** **VPN** **Wireguard**

---

eduVPN mobile testing

CommerceTest Limited — **Visit https://NLnet.nl/project/eduVPN-testing**

VPN Fund

Verifpal

Secure communication over the internet is critical. Humans however are not infallible, and the same holds for the humans that design the protocols that should make our internet traffic safe. Internet engineers and software developers need to handle a lot of complexity, and even a small

oversight or a very improbable scenario or combination of factors can mean breaking part or whole of the protection required The secure technologies we depend on to keep internet communications secure are frequently found to suffer from fundamental design vulnerabilities as well as implementation errors. Truth is, while trust is a fundamental human trait, we should not just trust human intuition to get everything right.

This is where computers can come to help us out, to see if we can underpin that trust in a systematic way. Computers have no problem to exhaustively try out all options, even if it takes them millions and millions of tries. When instructed in the right way, that means their endless combinatorial capabilities can be used to simulate even the most unlikely of events. Again and again, if necessary. A lot of awesome computer science brain power has gone into so called formal proofs. Formal proofs use very strict mathematical modelling to take everything that could possibly happen into account, and prove that the software or protocol at hand does what it is assumed to do. However, as you may imagine, this modelling can get pretty complex and as such is an art in itself - restricting the usage to a very limited set of experts. However, once you have the models right you can actually go a lot further than just prove the protocol: from the model you can automatically generate secure software libraries that you can be sure implement the protocols involved exactly right. This is a guarantee that no human programmer can give.

VerifPal is new software that makes formal verification of cryptographic protocols more accessible and intuitive. It is a breakthrough that regular users and software engineers can easily write out (or model) protocols to verify whether they are secure, and then immediately them against all sorts of possible attacks.

VerifPal is one of the first technology projects funded by the NGI and has already been used to verify the security of widely used protocols that for example protect Whatsapp and Signal-messages. Through this new project VerifPal will further prove that its verification of these protocols is sound, create extra implementations of the modeling language for a larger user base and integrate other protocol verification software to add additional layers of security checks. This way software developers, engineers and students around the world can benefit from formal verification software that is both secure and accessible, which can ultimately help make the internet a safer and more trustworthy place.

Technical description

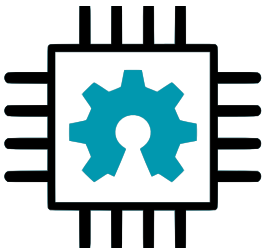
Verifpal is new software for verifying the security of cryptographic protocols. Building upon contemporary research in symbolic formal verification, Verifpal's main aim is to appeal more to real-world practitioners, students and engineers without sacrificing comprehensive formal verification features.

In order to achieve this, Verifpal introduces a new, intuitive language for modeling protocols that is much easier to write and understand than the languages employed by existing tools. At the same time, Verifpal is able to model protocols under an active attacker with unbounded sessions and fresh values, and supports queries for advanced security properties such as forward secrecy or key compromise impersonation.

Verifpal has already been used to verify security properties for Signal, Scuttlebutt, TLS 1.3, Telegram and other protocols. It is a community-focused project, and available under a GPLv3 license.

Symbolic Software — **Visit https://NLnet.nl/project/VerifPal-Proven**

NGI0 PET Analysis CodeGeneration Cryptography NGI0 SymbolicProof



When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component, the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

To break through this standstill, developer communities are working hard to deliver open, trustworthy and accessible alternative computer hardware that anyone can use, study, modify and distribute, just like they can with open source software. This project will deliver such open hardware development tools and bring the production of custom chips for particular computing purposes a lot closer to for example individual designers, start-ups and creatives. Setting free the knowledge to make such chips yourself pushes technological innovation forward, and opens up the computer market from top to bottom for creativity and for new classes of devices we need to power the Next Generation Internet.

Technical description

Current scaling of micro-electronics is focused on improving power, performance and cost per device but with an exponentially increasing start-up cost related to the increased process complexity. For the design of custom chips currently expensive proprietary electronic design automation (EDA) tools need to be used and hefty license fees are due for blocks implementing specific functions like the CPU, USB etc. All this together makes custom chip development only accessible for high-volume production and proprietary designs. In this project a development version of the libre licensed Libre-SOC system-on-a-chip will be manufactured in a 0.18um process combined with development on the open source tools and open source chip building blocks to make this possible. Development on the free and open source tools will be focused on making them compatible with the selected process and the building block development will be focused on the so-called standard cell library, the IO library and the SRAM compiler. This project fits in the longer term goal of the Chips4Makers project to make low-volume custom chip production possible using mature process technologies and free and open source tool chains and building blocks. Purpose is to get innovation using custom chips within reach of small start-ups, makers

and even hobbyists.

FibraServi BVBA — **Visit https://NLnet.nl/project/Chips4Makers**

NGI0 PET **ASIC** **CellLibrary** **EDA** **OpenHardware** **SRAMCompiler** **Semiconductor**

Adopt improvements in Email Encryption in KMail

Email was designed without privacy or security in mind, which is amazing for such a popular service. When you send an email, anyone that can gain access to your mail server or the mail server of the recipient can read your mail, from top to bottom. And copy it, for later usage. It is often compared to sending a post card, and of course in many cases there may be little harm in others reading what the weather is like in Paris. But what if you want to use email to send something confidential, something you do not want to share with others? Like a love letter, a political rant or an important contract? And what if you can't actually trust the mail man, for instance because the other party is using a free email service known to search through everything? Or what if you don' t like the fact that your writings are stored in a country you have never been, with different laws that may not be compatible with your thoughts about the world? Or what if you live in a country that has an unhealthy interest in bringing down certain political voices, or are part of a cultural minority that is at risk?

Computer specialists have been protecting their email with encryption for decades. This is the equivalent of putting your message very carefully in the blender, pressing the button before anyone else has read your mesage, shredding it up and sending a packet of shreds over to the other end. The amazing thing about cryptography is that you can magically (or rather mathematically) make it possible for your secret love - and not anyone else - to recreate the message from the shreds, and know it was you - and not anyone else - that sent it. For the rest of the world, the message would be meaningless garble pretty much forever.

However, the solution they came up with is not easy for normal people to work with. You need a lot of patience and technical skill to make use of it. Many people have tried, and could not get it to work or gave up because it hindered them. It was in fact too hard to turn it on by default. This means that most people are probably not even aware that it is possible to protect the contents of their email with cryptography. And so, unfortunately, normal citizens and business have been left behind - exposed to people reading their email messages, and (in the absence of other security measures) potentially also receiving fake or manipulated messages.

Autocrypt is a major contribution to make it far more convenient for people to use cryptography with email. It provides a specification for software to do most of the hard work (hence the portmanteau

Autocrypt, which comes from Automatically Encrypt), and thus help also normal users protect the privacy and security of their mail. This project will implement Autocrypt into the open-source and widely used email client Kmail, along with the MemoryHole-software which will further encrypt your email by hiding your email headers. This way every user of Kmail automatically uses secure and trustworthy email, without leaking any personal or sensitive information.

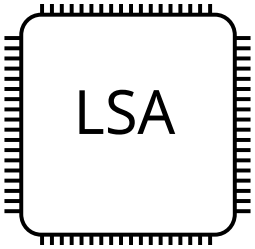
Technical description

The goal of this project is to make it more simple for inexperienced users to just use encrypted mails, at the click of a button. Autocrypt is a new method for email encryption, that needs nearly no user interaction. It performs the needed key exchange transparently in the background, and does key management automatically. Encrypted Headers is a protocol to send mail headers in the encrypted mail part. Traditional encryption methods leaked meta-data, which could be used for mass surveillance purposes. The result will be part of the KDEPIM codebase, so you don't have to install anything else than KMail to use these improvements.

Visit https://NLnet.nl/project/Kmail-Encryption

NGI0 PET **Autocrypt** **Email** **OpenPGP**

Libre Silicon compiler



Behind the screens of every mobile phone, laptop or tablet you will find essentially the same components that are produced by a small number of companies. Using patents and closed-off work methods these monopolists hold a firm grip on how essential technical building blocks of consumer electronics are actually made. Not only does this prevent innovation in the market, it also makes the devices that users, companies and governments across the world rely on for vital services and infrastructures essentially untrustworthy. If you cannot verify that the parts that make your device work are secure, can you really trust the device at all?

One of the ways to break through this standstill, is to construct computer parts from the ground up and make your designs open for everyone to check and verify. Combine this open hardware with open source software and you have a device that, with the right knowledge and skills, is completely transparent and customizable. This project aims to develop an open source production process for custom computer chips, making manufacturing of these chips quick, easy, inexpensive and auditable. NGI Zero funds various parts of this project, like this effort to create open and transparent design plans for computer chips.

Technical description

LibreSilicon Compiler (LSC) is a place + route suite for silicon. The main focus of this project is to produce legal and efficient silicon layouts from digital netlists (e. g. BLIF, EDIF). Traditionally the placement and routing problem are handled separately and in sequence and the final layout is given by the routing step. In this setup the routing step gains information from placement but not the other way around. LSC attempts to shift this paradigm to create a feedback loop between the two main problems to improve the solution. Furthermore we are incorporating formal methods to produce the compiler software and to verify resulting layouts. While the latter is standard practice, proving properties of the compiler software itself is only widespread in the domain of software compilers. This exercise will be favored by the use of the programming language Haskell and advanced theorem provers. Finally this software aims to profit from explicit module hierarchies given by the developers of digital logic in register-transfer level (e. g. Verilog, Chisel). Greedy solutions can be found for highly modularised chips: when logic is not inlined in the conventional software compiler sense, the size of problem instances is kept small. This also gives parallelism for free, as the dependency tree is resolved from the bottom up.

LibreSilicon — **Visit https://NLnet.nl/project/LibreSilicon-compiler**

NGI0 PET **HardwareDesign** **HardwareSynthesis** **Semiconductor**

Libre-SOC, Coriolis2 ASIC Layout Collaboration

libreSOC

When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component, the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

This ambitious project wants to deliver the first completely open computer processor in history - one you don' t have to merely trust, because you can verify and modify everything about it. All of the technology included, from top to bottom, will become available for inspection, and can be tuned by anyone technically capable enough. This will significantly contribute to the creation a new generation of computer technologies, as well as more energy efficient and cheaper devices. NGI Zero funds several important building blocks of this project, like this effort to develop open and auditable chip designs.

Ultimately these building blocks will come together in a transparent computer processor that can make our computing devices more trustworthy.

Technical description

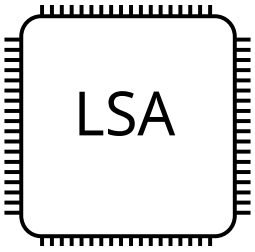
One of the key issues in a trusted, trustable ASIC is for the toolchain to be libre-licensed, so that there is no possibility for hardware-level spying or backdoor compromises. The Alliance / Coriolis2 ASIC layout toolchain by LIP6.fr is one of the leading tools in this area. The Libre-SoC is another project being funded through NGI Zero, and at this moment that project needs to get beyond FPGA-proven status. The challenging next phase is to do an actual ASIC layout. With the System-on-Chip being developed in nmigen (a python-based HDL), Alliance / Coriolis2 also makes sense as it is written in Python as well. The funding will go towards doing an ASIC layout in 180nm.

Visit https://NLnet.nl/project/Coriolis2

NGI0 PET HardwareDesign HardwareSynthesis LibreSoC Semiconductor

---

LibreSilicon



When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component, the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

One of the ways to break through this standstill, is to construct computer parts from the ground up and make your designs open for everyone to check and verify. Combine this open hardware with open source software and you have a device that, with the right knowledge and skills, is completely transparent and customizable. This project aims to develop an open source production process for custom computer chips, making manufacturing of these chips quick, easy, inexpensive and auditable.

Technical description

LibreSilicon aims to reduce the steep entry barriers to full custom application-specific integrated circuit (ASIC) design and help people to regain trust in their computing devices, right at the bedrock: When they are manufactured. LibreSilicon provides a standard for manufacturing semiconductors which allows platform independent process design kits (PDKs) and design rules that allow manufacturing the same chip layout in any factory that has calibrated their process according to the LibreSilicon specs with the PearlRiver test wafer. By introducing this process standard, full custom ASIC design should become available to private persons without corporate or academic access to IC foundries. After democratizing software development with tools like Arduino, and PCB design with tools like KiCAD, LibreSilicon will democratize ASIC design, and GDS2 intends to become the new Gerber file format for semiconductor manufacturing.

Lanceville Technology Group Co. Ltd. — **Visit https://NLnet.nl/project/LibreSilicon**

NGI0 PET HardwareDesign Manufacturing PDK Semiconductor

Libre-SOC Video Acceleration

libreSOC

When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component, the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

This ambitious project wants to deliver the first completely open computer processor in history - one you don' t have to merely trust, because you can verify and modify everything about it. All of the technology included, from top to bottom, will become available for inspection, and can be tuned by anyone technically capable enough. This will significantly contribute to the creation a new generation of computer technologies, as well as more energy efficient and cheaper devices. NGI Zero funds several important building blocks of this project, like this effort to create open and trustworthy hardware acceleration, so no closed-off hardware or software to run specific components is needed in the world's first open computer processor. Ultimately these building blocks will come together in a transparent computer processor that can make our computing devices more trustworthy.

Technical description

The Libre-SoC Project, has been funded by NLnet to get to FPGA-proven status. This was for the "core" (the main processor). One of the next, specialist, phases, is to ensure that its capabilities are useable to perform Video Acceleration. To do so, Video Software such as ffmpeg, gstreamer and their low-level libraries need to actually use the hardware-accelerated capability. A "normal" commercial processor usually has a separate proprietary VPU, along with proprietary software: both unfortunately are vectors for attack against users, undermining trust and privacy. Without access to Video Acceleration, users are left with the stark choice: be compromised, or don't watch any video, period. This project therefore provides a commercial-grade Video Decoder (minimum 720p) and helps restore trust in the software *and* hardware.

Libre RISC-V SoC — **Visit https://NLnet.nl/project/LibreSoC-Video**

NGI0 PET **OpenHardware** **Semiconductor** **VideoEncoding**

Libre-SOC Formal Correctness Proofs

libreSOC

When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component, the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

This ambitious project wants to deliver the first completely open computer processor in history - one you don' t have to merely trust, because you can verify and modify everything about it. All of the technology included, from top to bottom, will become available for inspection, and can be tuned by anyone technically capable enough. This will significantly contribute to the creation a new generation of computer technologies, as well as more energy efficient and cheaper devices. NGI Zero funds several important building blocks of this project, like this effort to provide so-called mathematical correctness proofs, which test every little part of a hardware design for possible logic flaws a human would not be able to spot. Ultimately these building blocks will come together in a transparent computer processor that can make our computing devices more trustworthy.

Technical description

Hardware projects like the Libre-SOC Project involve writing an inordinate amount of comprehensive unit tests to make sure everything functions the way it should. This is a critical and expensive part of the overall design process. Formal Mathematical Proofs (already quite popular in secure software development) provide an interesting alternative for several reasons: they're mathematically inviolate, which we believe makes them more trustworthy. And they are simpler to read and much more comprehensive (100% coverage), saving hugely on development and maintenance. From a security and trust perspective, both aspects are extremely important. Security mistakes are often accidental due to complexity: a reduction in complexity helps avoid mistakes. Secondly: independent auditing of the processor is a matter of running the formal proofs. The project aims to provide proofs for every module of the Libre RISC-V SoC, and therefore contributes significantly with the larger goal of developing a privacy-respecting processor in a way that is independently verifiable.

Visit https://NLnet.nl/project/LibreSoC-Proofs

NGI0 PET **FormalProof** **HardwareDesign** **Semiconductor**

Port of AMDVLK/RADV 3D Driver to the Libre-SOC

libreSOC

When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component, the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

This ambitious project wants to deliver the first completely open computer processor in history - one you don' t have to merely trust, because you can verify and modify everything about it. All of the technology included, from top to bottom, will become available for inspection, and can be tuned by anyone technically capable enough. This will significantly contribute to the creation a new generation of computer technologies, as well as more energy efficient and cheaper devices. NGI Zero funds several important building blocks of this project, like this effort to develop completely transparent 3D driver software. Ultimately these building blocks will come together in a transparent computer processor that can make our computing devices more trustworthy.

Technical description

The Libre SoC is being developed to provide a privacy-respecting modern processor, developed transparently and as libre to the bedrock as possible. As a hybrid processor, it is intended to be both a CPU and a GPU. GPUs are typically proprietary (and thus not fully transparent), as is the 3D driver software. The SoC design requires a Vulkan compliant hybrid hardware-software API. The development of the Kazan 3D Driver (developed from scratch inside the Libre SoC) that aims to provide such an API is therefore on the critical path to final release. Given the complex nature of 3D driver development, and because Kazan is a novel approach (written in rust, for security reasons) that dependency is considered a liability. This project develops a second, more traditional Mesa3D driver in c++. This reduces the pressure on the Kazan development, and allows for benchmarking and increased transparency and collaboration on this ambitious project.

The Libre RISC-V SoC — **Visit https://NLnet.nl/project/LibreSoC-3Ddriver**

NGI0 PET **Driver** **GPU** **OpenHardware** **RISCV**

Wishbone Streaming

When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component, the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

This ambitious project wants to deliver the first completely open computer processor in history - one you don' t have to merely trust, because you can verify and modify everything about it. All of the technology included, from top to bottom, will become available for inspection, and can be tuned by anyone technically capable enough. This will significantly contribute to the creation a new generation of computer technologies, as well as more energy efficient and cheaper devices. NGI Zero funds several important building blocks of this project, like this effort to improve an existing open source hardware component that will let parts of the open processor share data with each other. Ultimately these building blocks will come together in a transparent computer processor that can make our computing devices

more trustworthy.

Technical description

On System-on-Chips (SoC) the commercial grade bus infrastructure is covered by patents and at best available "royalty-free" (but with no ability to change). A serious alternative with significant adoption is the Wishbone SoC Bus, which is an Open Standard but does not yet have a "streaming" capability. That capability is needed for high-throughput data paths and interfaces. This project will provide an enhancement to the current Wishbone SoC Bus specification, provide Reference Implementations and Bus Function Models (BFM) to easily allows unit tests for all Wishbone BFM users. For demonstration purposes the project will implement an example peripheral to prove the overall concept.

Visit https://NLnet.nl/project/WishboneStreaming

NGI0 PET BusInfrastructure FormalProof HPC OpenHardware Streaming System-on-Chip

---

Libre-SOC Formal Standards Development

When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component, the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

This ambitious project wants to deliver the first completely open computer processor in history - one you don' t have to merely trust, because you can verify and modify everything about it. All of the technology included, from top to bottom, will become available for inspection, and can be tuned by anyone technically capable enough. This will significantly contribute to the creation a new generation of computer technologies, as well as more energy efficient and cheaper devices. NGI Zero funds several important building blocks of this project, like this effort to develop standards for video and 3D graphics acceleration, which are essential parts that the open processor will rely on. Ultimately these building blocks will come together in a transparent computer processor that can make our computing devices more trustworthy.

Technical description

Libre-SOC was first funded from NLnet in 2018. This was for the core of the project, based on an informally-developed Hybrid CPU-GPU 3D instruction set that had been written (and implemented in a simulator) in the 18 months prior to contacting NLnet. During the implementation it became clear that a lot more work is needed, and, further, that to meet proper transparency criteria, the proposed instruction set enhancements would need to be properly written up. In addition, negotiations and communications with the Standards Body responsible for POWER ISA (the OpenPower Foundation) also needed to be taken into consideration. The goal of this project is to deliver on those requirements, and achieve full transparency and understanding of the Libre-SoC.

The Libre RISC-V SoC — **Visit https://NLnet.nl/project/LibreSoC-Standards**

NGI0 PET **3D** **Acceleration** **Conformance** **Graphics** **LibreSoC** **OpenHardware** **RISCV** **TestSuite** **Video**

CryptPad for communities

Collaboratively writing a document together in real-time with others is still a bit magic. Someone else, perhaps on the other side of the planet, is typing something. And within a fraction of a second, the text magically appears on your screen. If you insert some text in the text just typed, this travels to all people you are in the session with. This amazing technology is the ideal companion for say an online meeting - everyone can contribute, and correct any flawed minutes without much effort.

For this kind of collaboration in real-time, there is a limited set of options in the market you can use. Most available services in the market like Google Docs, Microsoft Office or LibreOffice Online share one very undesirable characteristic: you need to fully trust the company running the service you use. Whomever has access to the servers used to connect everyone together, can read everything you have written - and deleted. That means that if you need to work on something confidential like an important contract, you may want to reconsider using the service. If you by accident cut and paste a password in the wrong window, you probably need to change it.

Especially if you write about sensitive topics like corruption, money laundering or state surveillance this open backend you cannot control is a really significant problem. If the server is located in another jurisdiction, you probably want to watch carefully what you write - you may inadvertently violate some laws you are literally unaware of.

Cryptpad is different: it is free and open source software you can run anywhere you want yourself. This means you can choose someone you really trust, rather than being forced to trust. But even better,

CryptPad will make everything you do undecipherable to the outside world before anything is sent to the service to be distributed among all the participants. From a user perspective it works as any other application. That means CryptPad puts you square back in control.

In this project, Cryptpad will become more accessible and usable for groups of users, introducing highly requested features like adding comments to documents and more access control for Cryptpad-hosts of group members. Ultimately this will give user communities more control over how they work together and share data efficiently and safely.

Technical description

CryptPad is a secure and encrypted open-source collaboration platform, that allows people to work together online on documents, spreadsheets and other types of documents. The amazing thing is that while the participants can work with these web applications as they would with any normal tool, the server has no way of telling what it is they are working on. Everything is encrypted on the device of the user, before it is sent to the server. The "CryptPad for communities" project will improve the experience of users adopting the platform for community management tasks. We'll spend time solving the issues most commonly reported by our users as obstacles to their broader adoption of the platform as an alternative to proprietary services. Document review is as important to many as collaborative editing, so we'll implement comment workflows that integrate our recently introduced social features into our text editors. Our Kanban and spreadsheet apps will both receive some crucial updates to better facilitate project management tasks without compromising on privacy. We'll develop extra access control features based on users' public keys for documents that require stricter protection than is currently offered. Those hosting their own CryptPad instance will benefit from new functionality for their admin panel as well as detailed documentation to make server management more accessible. Finally, we'll implement extra controls permitting admins to limit access to their instance by requiring invites for registration. Altogether we hope these tools will allow communities more determination when it comes to their data, their processes, and their ability to work together productively.

XWiki SAS — **Visit https://NLnet.nl/project/Cryptpad-Communities**

NGI0 PET **ServerApp**

Client-side **Collaboration** **Encryption** **Groupware** **NGI0**

Off-the-Record messaging version 4

One of the things people enjoy the most about the internet, is that it enables them to talk to others remotely almost without limit. Internet allows anyone to keep closely connected with friends and family, and help their kids solve a math problem while they are at work. People collaborate with their colleagues from the couch of their living room, the cafe where they enjoy lunch or on their cell phone on the bus to the gym. Businesses can easily service their customers where this is most

convenient to them, without having to travel themselves. This is so convenient, that some businesses have already moved entirely online. Internet communication has become the nerve center of whole neighbourhoods, where people watch over the possessions of their neighbours while these are away for work or leisure.

Tools like Whatsapp, Signal and Telegram has become a mainstay for individuals, businesses and even local governments as a low-threshold channel to reach out to people, be it for a friendly chat or customer support. The services promise their users that everything they share and discuss is shielded off from spying eyes. Nothing is said about the metadata that shows who talks to who, and where they are. And these still suffer from issues of centralized services maintained by one party, like censorship and country-wide bans. A Signal user cannot communicate with a Telegram user through either service. And all of them can be blocked easily.

Actually secure, private and decentralized chat is important to offer users but also businesses and governmental organizations a transparent and trustworthy communication channel. This is especially the case when sensitive and personal data is shared and even more so for people living in less democratic societies who run the risk of being arrested or harassed for who they talk to or what they say. Everyone has the right to confide in someone, be it a friend or a professional, and be sure what is said does not leave the (virtual) room. For journalists, activists, whistle-blowers and vulnerable minorities, this right can be a matter of life or death.

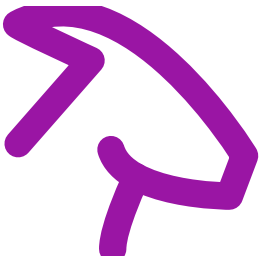
The off-the-record-protocol offers a way to both encrypt the content of a conversation as well as its participants. Uising this protocol to chat is in a lot of ways like talking to someone in real life, with no cameras, devices, or spying eyes around: no-one needs to know a thing who you talked to and when. This right to offline privacy should be respected digitally as well, which is what this project aims to do using the tried and proven off-the-record messaging protocol to develop a proper privacy-friendly chat app.

Technical description

OTRv4 is the newest version of the Off-The-Record messaging protocol. It is a protocol where the newest academic research intertwines with real-world implementations. It's aim is to give end-to-end encryption, deniability, authentication, forward secrecy and post-compromise security for any kind of messaging (online or offline). The goal of this new version is to give the most secure privacy and security properties that have a real impact on the world. This new version aims to be available in different desktop clients (that use XMPP or other messaging protocol) and in mobile clients.

Dyne — **Visit https://NLnet.nl/project/OTR4**

NGI0 PET **Cryptography** **Deniability** **MobileApp** **OTR** **Protocol**



As you fire up your computer, laptop or smartphone and click your browser icon to connect to your favorite site, do you know what happens behind the scenes? Many websites actually have dozens of different trackers, and some of these have such a global presence that they can form a pretty clear picture of ones online behaviour. Some argue that privacy is and has been dead for quite some time. As long as users have a quick internet connection and can access the web, email, games and messages without a hitch, they won't complain. But if you question people about the importance of online privacy, usually the answer is that it is indeed important and should be better protected. What is happening here? Perhaps we misunderstand carelessness with unfamiliarity. The technology behind most of our devices, our connection to the internet and the virtual spaces we inhabit is complex, yes, but the solutions we use to access them have also kept actual control away from us under the guise of 'intuitiveness' and 'pick up and play'. Playing here means playing by the rules of the developer, not by your own. What users instead should have are tools that give them actual access to what their devices do, what choices are made, and decide for themselves whether they agree with them or not.

Privacy isn't dead, we just lack the tools to actually protect it. This is true for both the user as for the website owner that wants to know who their visitors are. Web analytics software is usually invasive by default and give a website owner little control over what data is logged and who else gets to access it. This is not only unfriendly to your website visitors, it can also be bad for business when profiling data is leaked or misused by a third party and you are held responsible. Instead of giving away so much control over the website you own and the visitors you want to attract, why not do it yourself?

GoatCounter is an effort to develop simple web statistics technology that does not track users, is more accessible and easy to use than proprietary analytic services and at all times lets you own your analytical data. This is one of the tools we need to make an internet that revolves around its users: privacy-friendly by default, transparent and only owned by yourself.

Technical description

GoatCounter aims to provide meaningful privacy-friendly analytics for businesspurposes, while still staying usable for non-technical users to use onpersonal websites. The choices that currently exist are between hosted online services that have serious privacy issues, running your own complex software, or extremely simplistic "vanity statistics". GoatCounter attempts to strike a good balance between various interests. Major features include an easy to run self-hosted option, an intuitive user interface that is also accessible to website maintainers with accessibility needs, and meaningful statistics that go beyond "vanity stats" but still respect user privacy.

Visit https://NLnet.nl/project/GoatCounter

NGI0 PET Analytics GPDR Self-Hosted Webservice

Secure User Interfaces (Spritely)

Online deception and social engineering, better known as phishing, is becoming a bigger threat everyday as we store and share more of our (sensitive) data online. Because the risk of getting caught is low and the payoff potentially high, fraud and theft on the internet is running rampant. Through fake emails, websites and instant messages, users and businesses are tricked into sharing sensitive data like passwords and credit card details. People can end up with all of their money stolen, their lives ruined or their personal sensitive data spread all across the internet.

Social media are one of the channels used by cybercriminals to extort and pressure users into handing over their credentials. Because phishing attempts become more believable and pervasive everyday, social media networks need to protect their users. Commercial networks like Twitter and Facebook can organize protective measures on their own, while decentralized networks like Mastodon and Pleroma rely on the hosts of individual instances to protect users against spam, trolls and phishing attempts.

While decentralized social media offer users more privacy, less ads and data governance, they are more vulnerable to all sorts of cybercrime that can turn users away. This project will improve privacy and security in decentralized social networks by showing users how they can best protect themselves against phishing, using the Mastodon web interface. Decentralized networks by design give users more governance over their personal data and anonymity, but to win users over should provide the same security as centralized, commercial social media. This project will help create decentralized networks that are just as privacy-friendly as they are secure, putting the user first.

Technical description

Spritely is a project to advance the federated social network by adding richer communication and privacy/security features to the network. This particular sub-project aims to demonstrate how user interfaces can and should play an important role in user security. The core elements necessary for secure interaction are shown through a simple chat interface which integrates a contact list as an easy-to-use implementation of a "petname interface". Information from this contact list is integrated throughout the implementation in such a way that helps reduce phishing risk, aids discovery of meeting other users, and requires no centralized naming authority. As an additional benefit, this project will demonstrate some of the asynchronous network programming features of the Spritely development stack.

Libre Labs — **Visit https://NLnet.nl/project/Spritely**

NGI0 PET ActivityPub WebOfTrust

Rust Threadpool

Software security today can be a matter of life and death, because we rely on all sorts of applications and programs to keep our lights on, our vehicles moving and our money available. Backdoors in software leave room for attackers to steal data or disrupt important processes. To make this less abstract: hackers have already managed to get inside the control rooms of power plants to potentially cause blackouts.

One of the ways to make software more secure is by addressing security at the most basic level: the code that makes up the software itself. More precisely, the programming language software developers use to create their programs and that runs important parts of a process. Rust is a programming language focused on security that protects against software bugs and vulnerabilities attackers may try to exploit. This project improves a specific Rust tool that gives developers more control over how they design technology, ultimately making the software we use and trust more stable and secure.

Technical description

ThreadPool is a free and open-source library that provides a simple and intuitive interface for programmers to multi-threaded programming. ThreadPool aims to make parallel programming accessible to the general public. Running tasks in parallel is a vital building block for building efficient solutions on modern hardware. Combined with Rust's type-system this library allows programmers to parallelize their applications without introducing unsafe behaviour while managing the administrative tasks of interacting with the operating system.

Visit https://NLnet.nl/project/RustThreadpool

NGI0 PET Library Rust Threading

Dat Private Network

It has been several decades since the first internet connection was made and we still have not solved the issue of free, safe and controlled file sharing. Common channels like email set strict file size limits and leave possibly sensitive data strewn about inboxes and servers. File hosting and

sharing services keep users in the dark about what happens to their uploads and do not keep files up for long. Torrent environments are fraught with illegally uploaded or malicious content that may be harmful for users, who have no tools to verify or authenticate anything or anyone.

Users want to share their data with their friends, colleagues, partners or clients, not with some unknown third parties or shady service providers. Instead of relying on someone else's infrastructure, users can build a network themselves to store and share data as peers. Dat is a longstanding community effort to make peer-to-peer (p2p) data sharing accessible, easy and safe. Originally intended for academics, Dat is now used to share websites, music, art, chat programs and other content hosted and provided only by users and data owners themselves. Because the dat protocol distributes data on every peer's computer, it can even work offline or with limited connectivity. And because all distributed data is uniquely identified, peers can verify that files have not been tampered with.

This project aims to make the dat network and technology more accessible for archiving communities. Putting archives on the internet can be a foolproof method to preserve unique material, but only when the archives are securely distributed and data owners keep complete control over the data. The foundation responsible for the dat-technology will develop easy-to-use tools to first encrypt and store archives offline and then distribute the data on a peer-to-peer-network controlled by the archiving community. This way unique data is securely backed-up, always available and trustworthy, because no unauthorized third parties have access. These tools and infrastructure can of course benefit communities everywhere to store and share their unique information on their own terms.

Technical description

The dat private network is a self-hosted server that is easy to deploy on cloud or home infrastructure. Key features include a web-based control panel for administration by non-developers, as well as on-disk encryption. These no-knowledge storage services will ensure backup and high availability of distributed datasets, while also providing trust that unauthorized third-parties won't have access to content.

By creating a turnkey backup solution, we'll be able to address two of our users' most pressing questions about dat: who serves my data when I'm offline, and how do I archive and secure important files? The idea for this module came from the community, and reflects a dire need in the storage space -- no-knowledge backup and sync across devices. A properly-designed backup service will provide solutions to both of these questions, and will do so in a privacy-preserving way.

This deliverable will put resources into bringing this work to a production-ready state, primarily through development towards updates that make use of the latest performance and security updates from the dat ecosystem, such as NOISE support. We plan to maintain the socio-technical infrastructure through an open working group that creates updates for the network as it matures.

Dat Protocol Foundation — **Visit https://NLnet.nl/project/Dat-Private**

NGI0 PET **Dat** **Decentralisation** **Encryption** **FileSharing** **Foundation** **Middleware** **P2P** **Protocol** **Server** **Storage**

v i d e o b o x

FOSDEM

One of the things people enjoy the most about the internet, is that it enables them to talk to others remotely almost without limit. Internet allows anyone to keep closely connected with friends and family, and help their kids solve a math problem while they are at work. People collaborate with their colleagues from the couch of their living room, the cafe where they enjoy lunch or on their cell phone on the bus to the gym. Businesses can easily service their customers where this is most convenient to them, without having to travel themselves. This is so convenient, that some businesses have already moved entirely online. Internet communication has become the nerve center of whole neighbourhoods, where people watch over the possessions of their neighbours while these are away for work or leisure.

However, users have a hard time to understand how privacy is impacted if they use the wrong technology. Because internet works almost everywhere, the natural privacy protection of the walls of a house, a school or an office is gone. Unlike the traditional phone companies, many of the large technology providers run their business not on delivering an honest service but on secretly eavesdropping on their users and selling information to others. It is mostly not about what you say, so it is relatively easy for providers to allow some form of privacy by encrypting messages. The more interesting parts are who talks to whom, when, and where they are in the real world while they meet on the internet. if you want to be reachable across the internet, you have to constantly let the communication provider follow you wherever you go. This makes the private and professional lives of citizens an open book to companies that with the help of AI and other technologies make billions from selling 'hidden data' normal people are completely unaware of even exists. And of course in societies that are not so democratic, this type of information is critical to bring down opposition and stifle human rights.

These risks of surveillance and profiling also exist for videoconferencing, which can be very useful to reach a worldwide audience, but is not so democratic when that audience is continuously tracked. Especially public institutions like schools, universities and local governments should not rely on proprietary hardware and software for videoconferencing, as they risk having their viewers logged and their videos stored and monetized. Instead public conferencing can be done with publicly developed and transparent devices and tools.

This project aims to develop open videoconferencing hardware and software for FOSDEM, the largest European gathering on free and open source software. In the main campus of the Belgian ULB-university, thousands of developers gather from all over the world to discuss and promote free and open source software. The conference is non-commercial and entrance is free for all. At times over 50 parallel sessions are organized and every talk is recorded and streamed, currently using hardware that is not entirely transparent. This project aims to develop open videoconferencing devices that are not only free, but also less costly and sizable. FOSDEM will then be available for everyone using the same open hardware and open source software promoted at the conference. And other organizations using the FOSDEM-setup will benefit just the same from trustworthy, transparent videoconferencing, built solely to spread ideas and knowledge.

Technical description

The goal of the FOSDEM video box project is to develop a cheap, compact, open hardware & free software video-to-network solution. Initial motivation came from scratching our own itch: replacing 60 bulky, costly, not entirely free boxes currently used at the https://fosdem.org conference. Several other conferences have already used the current setup successfully. We expect this number to grow in the future. The solution being free software and open hardware should make it flexible to adapt to different

environments, like education. Being cheap and compact encourages experimental use in areas difficult to foresee. On the hardware side, we use the open hardware Olimex Lime2 board (EU built!) as a base. We plan an open hardware hdmi input daughterboard, iterating on a simplified prototype that helped us verify feasibility. On the software side, the core Allwinner A20 chip has attracted a lot of free and open source development already. That enables us to focus our efforts on optimising video encoding on this platform from a hdmi signal to a compact network stream.

FOSDEM vzw — **Visit https://NLnet.nl/project/VideoBox**

NGI0 PET **Hardware** **OpenHardware** **Videostreaming**

Langsec in Pectore

Do you completely understand how your computer, laptop or smartphone works? Do you know what happens behind the browser, the text editor, the operating system? Probably not, and that is not a surprise nor is it something to be ashamed of. The development of consumer electronics is like a web that becomes increasingly intricate, where new technologies added continuously without anyone checking how the wires are connected or if there is a risk for short-circuiting. All sorts of vulnerabilities and back doors have crept in software and hardware over the years that even the developers themselves are sometimes unaware of.

You want technology you can trust. This becomes a matter of life and death when that technology helps keep you alive. To ask you one more question, do you know how an implanted medical device like a pacemaker works? And who can access and control it? Devices like pacemakers but also insulin pumps today are often connected devices, which essentially makes your body a part of the internet. And given the poor security of most so-called 'smart' devices, this puts patients using medical implants in a lot of danger.

This project aims to give people back control over their connected medical implants, starting with a secure and transparent pacemaker. Instead of creating a device with too much capabilities and a large attack surface, we can make single-purpose machines that only do what they need to do, using so-called langsec or language security design principles. Simply put, you do not have to shut a backdoor if there is no backdoor to begin with. Medical implants designed this way have a minimal attack surface and as such are more secure by design. Extending this approach to other connected devices can potentially solve the rampant security and privacy problem of 'smart' devices we are currently facing.

Technical description

Design and build a Proof-of-Concept (PoC) cardiac pacemaker circuit with an analog/mixed-signal CMOS ASIC based on a description of the device functionality as formal grammar/automaton based on language security (langsec) design principles. Internet-of-things (IoT) devices are usually designed around a general purpose microcontroller with a much larger state space than needed for their purpose. Only after the initial design, interface capabilities of the IoT device are artificially restricted for privacy and security. An implanted pacemaker is a safety-critical IoT device that fits into a very small state space, as proven by early pacemaker designs that did not use high performance microcontrollers. Langsec methods use formal grammars to specify minimal interface parsers to reduce the attack surface, but not

the attack volume behind the attack surface. As PoC, formal langsec methods are adapted to reduce the attack volume of a pacemaker: A domain-specific language (DSL) translates requirements of a cardiac pacemaker patient and an information security researcher (ideally one and the same person) into an implantable minimum state space analog/mixed signal pacemaker application specific integrated circuit (ASIC). Such a minimum automaton methodology can be transfered to less life-critical IoT devices. ASICs for minimum automaton IoT designs are a use case for completely free CMOS IC fabrication processes, e.g., LibreSilicon. Non-essential state space that isn't implemented can't be hacked.

Visit https://NLnet.nl/project/LangsecInPectore

NGI0 PET **EmbeddedSystems** **FormalGrammar** **OpenHardware**

R e d w a x

One of the oldest questions on the internet is: how do you adequately prove you are you? Or perhaps the reverse formulation offers a better mental model: how do you prevent others from succeeding in pretending they are you? Now lets flip this question around once more: how would you like to see this managed yourself, if you could? How heavy-weight or convenient do you want to be proven that you are you, to allow you to get into your own environment or have something done on your behalf? And what is it worth to you in terms of effort? Would you be willing to spend a minute to have some clever secure device you have in your pocket involved? Authenticate via your mobile phone? And what if you are in a rush, or on the go? Are you happy with some company like your email provider or a large social network having the ability to make that judgement, based on a user login a few hours ago? And what if that company is based in some other jurisdiction, and could be forced to let others in as well? Or would you rather choose your own identity, and formulate direct rules to have complete control at any given point?

As could be guessed, individual people have a need for different levels of confidence and security in different contexts. A security breach matters perhaps less if you just want to login to a music service to change a playlist. After all, the worst that can happen is that someone messes things up and you have to create a new one. It matters a great deal more if you want to do a significant financial transaction at work, or open the door of your house remotely to let the babysitter in while you are delayed in traffic. Perhaps you can think of scenarios where you want even more control.

So what proof to use as the basis of your trust, and the subsequent actions taken? Historically people rely on some authority they collectively trust. Such an authority has typically taken high tech countermeasures to make the channel through which that trust is conveyed hard to fraud. A passport or banknote are quite tricky to fabricate due to the use of special techniques. Online we have only a very limited amount of trust "anchors" of varying quality. The domain name system is such an anchor, digital certificates or customer relationships are another. Today, having access to a certain mail account or

phone which is known to be yours is the most common proof used. Email is often called the "poor man's solution" to identity management, and it is what most organisations and businesses fall back on. Can't log in? We will send you an email to reset your login. Just click on the link. And of course, email was never designed to be safe. It kind of works, but really we can do better.

Perhaps your use cases require more strict proof than that of normal consumers, or less strict proof. Even for a single large service provider, it would be hard to figure this out satisfactorily for all users. For the same reason people write their own testament to document what should happen with things they own or control after they die, you want to document what should happen with things you own or control what happens when you are physically absent. There is no universal will that is acceptable to all, nor is there a universal policy that satisfies all use cases.

So what if you yourself would be able to create and control your own identity, and determine your own proofs and methods? In order to function in a global internet, you would need to be able to convey your requirements and demands in a portable way. There would be no central authority dictating you what to do here. That would mean you you yourself would have to make things explicit upfront in a foolproof way - so that elsewhere on the internet people and services would know what you expect them to do to distinguish the real you from fraudsters.

This project will push decentralized trust management forward and make it instantly usable for all sorts of online services. Right now there is a lack of suitable standards for decentralized trust management in browsers. Using existing software that embody best practices in the field, new standards will be developed that make decentralized trust management accessible and easy to implement. This way secure and decentralized identity management can become the default, making for a more trustworthy and less centralized internet.

Technical description

The internet was not designed as a public infrastructure and most of the engineering trade-offs of the lower-layer technologies have generally erred on the side of accommodating fast growth and ease rather than values such as security, confidentiality and privacy. Yet today the internet is everywhere from providing a place for democratic discourse to healthcare to finance and personal communication. Redwax aims to decentralise trust management so that the values security, confidentiality and privacy can be upheld in public infrastructure and private interactions. The overarching goal of Redwax is to strengthen the existing technologies and infrastructure by providing a modular and practical set of tools to manage public key based trust infrastructures as currently used. These tools capture and hard code a lot of industry best practice and specialist PKI knowledge so that they can be put into the hands of a much wider community than currently served by a few specialist industries. With this project the Redwax team hopes to help re-establish (and/or strengthen) the support for these non-centralized trust management technologies inside web browsers and other relevant applications by working with standards organizations and industry coordination groups, and to create the initial reference implementations for their standardisation.

Red Wax — **Visit https://NLnet.nl/project/Redwax-PKI**

NGI0 PET **Authentication** **IETF** **PKI**

Improve usability of Linux firewall userspace tools

Most users rely on antivirus programs to keep their system and important data safe and private. Visited sites, downloaded files, email coming in and out, everything should pass through a digital border control that keeps malware and spyware out. Perform a complete system scan every other month and most users will be reassured: I am safe.

If your antivirus program is the main filter between the wild west of the internet and your device and data, you want to be sure you can trust that program to keep you safe. What do you do? Do you check out software reviews and ratings, ask a friend, simply rely on the default antivirus software that comes with your operating system?

How about using a firewall built into an open source operating system that is governed by a worldwide community that constantly checks and tests every cog and wheel? Netfilter is software built into the most popular open source operating system Linux that lets users control how incoming internet traffic is filtered, among many other useful features. This project will make netfilter and its many options more usable, inform you in greater detail about occuring errors and provide useful hints how to improve the firewall. Ultimately this can help make Linux a more safe operating system and give users more control over their online safety.

Technical description

Netfilter is the project offering the packet classification framework for GNU/Linux operating systems. Netfilter supports for stateless and stateful packet filtering, mangling, logging and NAT. Netfilter provides a rule-based language to define the filtering policy through a linear list, sets and maps. This language is domain specific and it provides a simplified programming language to express filtering policies.

Firewall operators are usually not programmers, although they are typically knowledgeable about shell scripting. Humans currently have few means to check for mistakes when elaborating filtering policies, which as a result can interact in unpredictable ways or cause performance issues - meaning one can never be sure how much they can be trusted to protect users.

Lack of correctness and inconsistencies emerge as the rule set increases in complexity. Introducing ways to assist the operator to spot these problems and to provide hints to express the filtering policies in a better way would help to improve this situation. Error reporting is another key aspect to assist humans in troubleshooting. This project aims to extend the existing tooling to introduce infrastructure to cover this aspects.

Netfilter — **Visit https://NLnet.nl/project/LinuxFirewall**

NGI0 PET Filtering Firewall Kernel Linux

M e t a - P r e s s . e s

To keep up with the news online, you need to traverse a swamp of third-party trackers on most newspaper websites or rely on news aggregators that give you little choice how to search and select from the multitude of stories out there. And your search engine of choice just might put a few fake news articles in between actually relevant news results simply because these sensational and untrue pieces attract a lot of (unjust) attention.

Instead of evading all sorts of surveillance schemes and fake news, news readers simply want to search through relevant reporting and reliable news sources to catch up on the latest developments. This projects will develop such a press search engine in the form of a Firefox addon, where everything happens strictly between your computer and the newspapers you are interested in. No surveillance, no censorship and no fake news, but instead privacy-friendly, trustworthy reporting as it should be.

Technical description

Meta-Press.es is a press search engine, in the shape of a browser add-on. When using it, everything happens between the user's computer and the queried newspapers. Using Meta-Press.es, there is no data sent to third party (including our servers). We're not asking the users to believe that we respect their privacy, it's a matter of verifiable fact that we do. That means there is no single point of failure, of surveillance or of censorship.

Acoeuro.com — **Visit https://NLnet.nl/project/Meta-Presses**

NGI0 Discovery BrowserPlugin News SearchEngine

---

p c b - r n d

Behind the screens of every mobile phone, laptop or tablet you will find essentially the same components that are produced by a small number of companies. Using patents and closed-off work methods these monopolists hold a firm grip on how essential technical building blocks of

consumer electronics are actually made. Not only does this prevent innovation in the market, it also makes the devices that users, companies and governments across the world rely on for vital services and infrastructures essentially untrustworthy. If you cannot verify that the parts that make your device work are secure, can you really trust the device at all?

One of the ways to break through this standstill, is to construct computer parts from the ground up and make your designs open for everyone to check and verify. Combine this open hardware with open source software and you have a device that, with the right knowledge and skills, is completely transparent and customizable. To create this open hardware, we need open design tools and file formats. This project will create open source tools to design and edit the most fundamental building blocks of almost all electronic devices: printed circuit boards.

Technical description

Pcb-rnd is a modular printed circuit board editor that is designed with the UNIX mind set. It has a convenient GUI for editing the graphical data of the board but is also has a handy command line interface. Both the GUI and the CLI aspects are scriptable (in more than 10 scripting languages) and pcb-rnd can also process boards as a headless converter tool. It has support for various proprietary schematics/netlist and board formats which makes it also a good choice for converting free hardware designs coming in proprietary formats to free file formats. Among the upcoming challenges are a full rewrite of the Design Rule Checker, more file format support and making the menu system even more dynamic to match the modular nature of pcb-rnd better.

Visit https://NLnet.nl/project/pcb-rnd

NGI0 PET EDA OpenHardware PCB Workflow

Sylk Mobile

One of the things people enjoy the most about the internet, is that it enables them to talk to others remotely almost without limit. Internet allows anyone to keep closely connected with friends and family, and help their kids solve a math problem while they are at work. People collaborate with their colleagues from the couch of their living room, the cafe where they enjoy lunch or on their cell phone on the bus to the gym. Businesses can easily service their customers where this is most convenient to them, without having to travel themselves. This is so convenient, that some businesses have already moved entirely online. Internet communication has become the nerve center of whole neighbourhoods, where people watch over the possessions of their neighbours while these are away for work or leisure.

However, users have a hard time to understand how privacy is impacted if they use the wrong technology. Because internet works almost everywhere, the natural privacy protection of the walls of a house, a school or an office is gone. Unlike the traditional phone companies, many of the large technology providers run their business not on delivering an honest service but on secretly eavesdropping on their users and selling information to others. It is mostly not about what you say, so it is relatively easy for providers to allow some form of privacy by encrypting messages. The more interesting parts are who talks to whom, when, and where they are in the real world while they meet on the internet. if you want to be reachable across the internet, you have to constantly let the communication provider follow you wherever you go. This makes the private and professional lives of citizens an open book to companies that with the help of AI and other technologies make billions from selling 'hidden data' normal people are completely unaware of even exists. And of course in societies that are not so democratic, this type of information is critical to bring down opposition and stifle human rights.

Users assume the confidentiality and privacy when they communicate, and they are morally justified to do so. There is nothing natural or final about internet communication providers having access to all this very personal information - or going down the dark path of selling data about customers. The cost of this in terms of internet usage and computer power needed is actually negligible, and so all it takes it the availability of open alternatives that people can use. Sylk is clearly one part of the puzzle: it is a mature open source videoconferencing tool that anyone can install anywhere for free. Businesses like the internet provider or the IT company around the corner can run it for their customers, and individuals can run it themselves from their home. And by switching, people can regain their privacy and make communicating via the internet as secure and confidential as we all need it to be. This project extends the unique and user-centric features of Sylk to smartphones and other mobile devices, offering an important private and trustworthy alternative to other videochat and instant messaging apps.

Technical description

Internet communications privacy is important to users, and there is a limited set of encrypted multiparty audio and videoconferencing solutions available to consumers and businesses today. The market, predominantly occupied by proprietary services that often require risky plugins, lack introspection and transparency, proved to expose users to significant security and privacy issues. This trend must be counteracted by better open source equivalents. Sylk Mobile provides a multi-party video encrypted conferencing solution mean to run on an end user computer or a mobile device. It is based on the WebRTC standard, and has a focus on user privacy and easy of use.

AG Projects B.V. — **Visit https://NLnet.nl/project/SylkMobile**

NGI0 PET **Conferencing** **Encryption** **NGI0** **RealtimeCommunication**

Securing PLCs via embedded Open-Source protocol adapters

The internet is unfortunately not only populated just by kind and careful people. And it wasn't designed to be secure either. This is a dangerous and rather unfortunate combination of circumstances, and one you should take into account when you use the internet. To make matters worse, not only can the internet be a fundamentally insecure channel when used incorrectly, a lot of devices we have been hooking up to this network are unsafe as well. This is the case for both old and new devices, that are sometimes misconfigured or lack any protective barrier at all.

Connecting unprotected devices to the internet is especially problematic when these machines have an important job to do, like automate an assembly line in a factory. That is what so-called programmable logic controllers (PLCs) do: digital computers that can reliably and constantly command other machines what to do in a high-pressure, harsh environment. Unfortunately, these PLCs are not as well protected from an online attacker as they are from heat or dust: some of them do not even have a secure password. Nevertheless these unsafe devices are connected to the internet more and more in the hopes of making factories 'smarter', but also increasingly vulnerable. This project will help protect PLCs against outside attackers, ultimately making the offline and online world blend together in a more trustworthy space.

Technical description

Industrial Programmable Logic Controllers have been controlling the heart of any production machinery since the mid-70s. However have these devices never been built for the usage in completely unprotected environments such as the Internet. Currently most PLCs out in the wild have absolutely no means to protect them from malicious manipulation (Most don't even have an effective password protection). Unfortunately "Industry 4.0" is all about connecting these devices to the Cloud and hereby attaching them to potentially unsecure networks. In the "Securing PLCs via embedded Open-Source protocol adapters" initiative we are planning on porting the Apache PLC4X drivers to languages that can also be used in embedded hardware. Additionally we also want to create secure protocol-adapters using these new drivers together with Apache MyNewt, to create protocol-adapters that could eventually even be located inside the network connectors which are plugged into the PLC in an attempt to reduce the length of the unsecured network to an absolute minimum without actually modifying the PLC itself.

Visit https://NLnet.nl/project/PLC4X-adapters

NGI0 PET Hardware PLC

CryptPad: Project Dialogue

Collaboratively writing a document together in real-time with others is still a bit magic. Someone else, perhaps on the other side of the planet, is typing something. And within a fraction of a second, the text magically appears on your screen. If you insert some text in the text just typed,

this travels to all people you are in the session with. This amazing technology is the ideal companion for say an online meeting - everyone can contribute, and correct any flawed minutes without much effort.

For this kind of collaboration in real-time, there is a limited set of options in the market you can use. Most available services in the market like Google Docs, Microsoft Office or LibreOffice Online share one very undesirable characteristic: you need to fully trust the company running the service you use. Whomever has access to the servers used to connect everyone together, can read everything you have written - and deleted. That means that if you need to work on something confidential like an important contract, you may want to reconsider using the service. If you by accident cut and paste a password in the wrong window, you probably need to change it.

Especially if you write about sensitive topics like corruption, money laundering or state surveillance this open backend you cannot control is a really significant problem. If the server is located in another jurisdiction, you probably want to watch carefully what you write - you may inadvertently violate some laws you are literally unaware of.

Cryptpad is different: it is free and open source software you can run anywhere you want yourself. This means you can choose someone you really trust, rather than being forced to trust. But even better, CryptPad will make everything you do undecipherable to the outside world before anything is sent to the service to be distributed among all the participants. From a user perspective it works as any other application. That means CryptPad puts you square back in control.

In this project, Cryptpad will introduce new, useful applications for polls and surveys that are equally secure, private and user-friendly. When you draw up a poll and survey, you have exclusive control over the content, format and submitted results, ensuring the privacy of everyone involved.

Technical description

Cryptpad is a real-time collaboration environment that encrypts everything clientside. The project will incorporate structured group interaction other than collaborative editing (e.g. gathering input through forms, polls) is a useful addition to this. This will replacing the current basic implementation of polls (like Doodle), and introduce surveys (like Google Forms). Authors will have exclusive control over the content and format of the polls and surveys, such as which questions are asked and the acceptable format of their answers. They'll also have control over the cryptographic keys which decrypt the submitted results, granting authors control over publishing. In addition, the project will develop an extension of its current notifications system to allow instance administrators to publish translatable messages visible to all their users. We'll use this broadcast system to distribute language-specific surveys and recruit willing users into a series of usability studies which will guide a second round of development for these applications.

XWiki SAS — **Visit https://NLnet.nl/project/CryptPadForms**

NGI0 PET **Forms** **NGI0** **Surveys**

Build Transparency (Trustix)

When you start up your computer, you will probably think twice before you download some random piece of software from the internet and run it. You know that doing so could allow unwelcome guests to your computer and your data. Your computer might even end up in a bot net. So when you see some nice piece of software, you will ask yourself the question: can I really trust the software? Perhaps you will check the origin it comes from. Better safe than sorry.

But even when you are sure you download a program from a trusted source, can you really trust the files themselves? To install something, usually you need to rely on binary files, like the executable installer you click to get things started. These are files that your computer understand, but are practically impossible for people to read and understand, let alone verify or audit. This project will develop a tool that compares software binaries across different providers and check whether they all work the same, identifying any binary that does something unexpected as compromised. Without any central point of trust (and failure), this way anyone can actually trust the software they run in their workspace or at home.

Technical description

When we install a program, we usually trust downloaded software binaries. But how do we know that we aren't installing something malicious? Typically, we have confidence in those binaries because we get them from a trusted provider. But if the provider itself is compromised, the binaries can be anything. This makes individual providers a single point of failure in a software supply chain. Trustix is a tool that compares build outputs across a group of providers - it decentralized trust. Multiple providers independently build the software, each in their own isolated environment, and then can vouch for the content of binaries that are the outcome of reproducible builds - while non-reproducible builds can be automatically detected. This is the first step towards an entirely decentralized software supply chain that can securely distribute software without any central corruptible entity.

Tweag IO — **Visit https://NLnet.nl/project/Trustix**

NGI0 PET **Decentralisation** **Reproducibility** **SupplyChain**

Verified Differential Privacy for Julia

Worries over our health and safety will in many cases take precedence over the perceived value of

our privacy. When it comes to our physical health and well-being, we are often in a strongly dependent position. Especially in times of great mental stress (like when a medical doctor breaks bad news to us) or fear (my daughter is late from school) we often lack the time and knowledge to really consider what data we actually want to make available and under which conditions. Many people in such situations reach a point of detachment and panic, where they hand out whatever data requested from them by whomever promises to resolve the stress. And once data is out there, it is hard to trace back.

But what if we do not have to give up our privacy for the sake of better, and more personalized health care, or our safety? What if we can have both? To know more about our physical and mental health, research needs to be done using sometimes very sensitive and personal information as data to be analyzed. What if we could make the analysis of this data worthwhile and beneficial to scientific knowledge, without compromising the privacy of anyone involved? That is where the concept of differential privacy comes in, which aims to publish aggregate, useful information out of a database without disclosing anything too personal. This project wants to build this differential privacy into a safety-centric programming language called Julia. This way analysis software programmed in Julia will protect personal data by design, offering scientists and experiments' subjects provable privacy guarantees.

Technical description

Differential privacy can be used to prevent leakage of private information from published results of analyses performed on sensitive data. Doing so correctly requires handling the extra complexity introduced by this technique, on top of the complexity of the analysis procedure itself. A proposed relief comes in the form of type systems. They allow tracking privacy properties of functions in types, where successful typechecking is equivalent to proving sound privacy guarantees. This aids the programmer in reasoning about code, detects implementation errors that are really hard to notice before one falls victim to privacy breach, and can give formal guarantees to the people whose privacy is claimed to be protected. This project will implement a typechecker based on the type system of the Julia programming language. Julia is a high-level, high-performance, dynamic programming language. While it is a general purpose language and can be used to write any application, many of its features are well-suited for high-performance numerical analysis and computational science. This should enable data scientists to compute privacy guarantees for any Julia function before they start working with real user data.

Visit https://NLnet.nl/project/Julia-DifferentialPrivacy

NGI0 PET **DifferentialPrivacy**

Etebase (EteSync) - protocol and encryption scheme enhancements

EteSync

People and organisations use both free and paid online services to manage their private address books, calendars and tasks. These services allow them to back up their data and share the same

information across different devices - so they can add an appointment or new contact while they are on the mobile phone at the train station, or on the couch at home, and it magically emerges on their desktop calendar. Other tools allow our loved ones to know where we are at any given moment in time. Given how personal and confidential such information is, use of these convenient services can make users vulnerable to all kinds of abuse.

That risk is not necessary. Service providers can perform the core services (sharing and backup) just as well without any knowledge about user data. Given how normal encryption has become elsewhere on the internet, for instance in instant messaging, it is high time that we start applying it to the information we store about the people we meet, the places we go and the things we do. The overarching goal of the open source EteSync project is to enable users to end-to-end encrypt all of their information, and the expected outcome of this project is to improve the EteSync-protocol that ensures this very sensitive data is well-protected.

Technical description

Etebase is an open-source and end-to-end encrypted software development kit and backend. Think of it as a tool that developers can use to easily build encrypted applications. Etebase is the new name for the protocol that powers EteSync, an open source, end-to-end encrypted, and privacy respecting sync solution for contacts, calendars, notes, tasks and more across all major platforms.

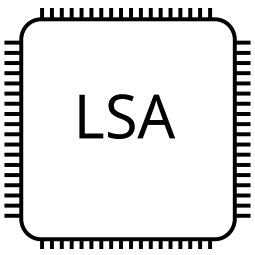
Many people are well aware of the importance of end-to-end encryption. This is evident by the increasing popularity of end-to-end encrypted messaging applications. However, in today's cloud-based world, there is much more (as important!) information that is just left exposed and unencrypted, without people even realising. Calendar events, tasks, personal notes and location data ("find my phone") are a few such examples. This is why the overarching goal of Etebase is to enable users to end-to-end encrypt all of their data.

While the Etebase protocol served EteSync well, there are a number of improvements that could be made to better support EteSync's current and long-term requirements, as well as enabling other developers to build a variety of encrypted applications.

Visit https://NLnet.nl/project/EteSyncEnhancements

NGI0 PET **Calendaring** **ClientSideEncryption** **Cryptography** **DAV** **E2EE** **Sync**

Standard Cell Library



Behind the screens of every mobile phone, laptop or tablet you will find essentially the same

components that are produced by a small number of companies. Using patents and closed-off work methods these monopolists hold a firm grip on how essential technical building blocks of consumer electronics are actually made. Not only does this prevent innovation in the market, it also makes the devices that users, companies and governments across the world rely on for vital services and infrastructures essentially untrustworthy. If you cannot verify that the parts that make your device work are secure, can you really trust the device at all?

One of the ways to break through this standstill, is to construct computer parts from the ground up and make your designs open for everyone to check and verify. Combine this open hardware with open source software and you have a device that, with the right knowledge and skills, is completely transparent and customizable. This project aims to develop an open source production process for custom computer chips, making manufacturing of these chips quick, easy, inexpensive and auditable. NGI Zero funds various parts of this project, like this effort to create open and transparent design plans for computer chips. NGI Zero funds various parts of this project, like this effort to create a free and open source standard cell library, which contais the most fundamental building blocks used to design computer chips.

Technical description

Without having an open standard cell library, any open hardware project depends on unknown components. This significantly hampers innovation, and is on the critical path of delivering truly open hardware chips. LibreSilicon's approach to this problem is generative, working from a (potentially verifiable) algorithm for automated sizing of transistors. All commercial available Standard Cell Libraries contain a small subset of all useful cells only, limited by the manpower of the vendor. They are hand-crafted and error-prone, and typically require Non-disclosure agreement (NDAs) while heavily depending on the underlaying PDKs - meaning that the outcome is hard to verify and trust. Goal it so produce a production quality free and open source Standard Cell Library.

Visit https://NLnet.nl/project/LibreSiliconStandardCellLibrary

NGI0 PET **CellLibrary** **Silicon**

Open Source DRTM implementation with TrenchBoot for AMD processors

Software security today can be a matter of life and death, because we rely on all sorts of applications and programs to keep our lights on, our vehicles moving and our money available. Backdoors in software leave room for attackers to steal data or disrupt important processes. To make this less abstract: hackers have already managed to get inside the control rooms of power plants to potentially cause blackouts.

So how can you know for sure you can trust the computer on your desk or the server you just connected to? Security experts will first go through the system and check which parts are critical to its security and

are potential points of failure. This is summarized as the trusted computing base, or TCB, of a computer system. To prove your setup is secure, you would need to reliably show that this trusted computing base is actually safe, which you can do with so-called roots of trust for measurement. This project develops open source tools that can provably verify the security of all computer platforms, as an important alternative to closed-of proprietary solutions which can infringe on user privacy and even their security.

Technical description

The Trenchboot project aims to create a unified framework for dynamic RTM (DRTM) implementation for all platforms. (D)RTM is used to verify if bugs or vulnerabilities have compromised a system, and as such is an important component to get to advanced stages of trustworthiness for our hardware.

3mdeb Embedded Systems Consulting — **Visit https://NLnet.nl/project/OpenDRTM**

NGI0 PET **Bootloader** **RTM** **TPM**

Indigenous

Social media are important for many people to stay in touch, but they also influence the way we act ourselves. Their persistent presence in our lives creates a lot of pressure to be someone else then we are. This is not a coincidence: the developers have designed the software to keep us 'engaged' at any cost. If we sleep less well because of a message we got just before we went to sleep, that is fair game. We may wake up with a hundred messages we "missed", which creates a lot of stress. But it keeps us engaged, and keeps the advertising cash flowing.

Another problematic issue to address is monoculture. Social networks do not allow to cross the boundary of their service in an easy way, leading to social lock in and a "winner takes all" scenario. This limits choice, but also exposes users to legal dangers. Confidential discussions through "private" messages for instance turn out to be not so private, such as the case where a United States got the social network Twitter to hand over the personal communication from European human rights activists and a member of the Icelandic parliament over a severe human rights violation by the USA military. The European Court of Human Rights would certainly not have allowed this, but it happened outside of our jurisdiction - even if all the actors never left Europe.

What if you could set your own rules for what content you see, who you can share it with and where you post your daily stories, pictures of your dog, or critical think pieces? One of the most independent places you can have on the internet, is your own website. You own the domain, you determine how your content can be analyzed and scraped, you determine what software you use to share your views with the world.

The global IndieWeb movement supports independent personal websites and helps users easily post,

share and connect with open standards and protocols. This project develops an app that lets users post and share as they would on social media sites, but instead of giving away their personal data and content to an ad-based company, post everything through their own site and see what their friends are doing on their website.

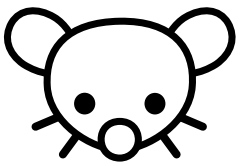
Technical description

Indigenous is a collection of native, web and desktop applications which allows you to engage with the Internet as you do on social media sites, but posts it all on your website. Use the built-in reader to read and respond to posts across the internet. Indigenous doesn't track or store any of your information, instead you choose a service you trust or host it yourself. Posts are collected on your website or service which supports W3C Microsub, writing posts uses the W3C Micropub specification. Popular services that support both are Wordpress, Micro.blog and Drupal, with more coming soon.

eps and kaas — **Visit https://NLnet.nl/project/Indigenous**

NGI0 Discovery **ActivityPub** **MobileApp** **SocialNetworking** **Webmention**

L e m m y



A lot of the people we talk to, the media we watch and the services we search for are found in or through using social media. For users these platforms offer easy and usually free services to send public and private messages, stay updated on relevant news and promote your business or product.

But the services these social media offer do actually come at a personal and societal cost. The platforms are not neutral exchange platforms like the rest of the internet. They do not just deal with all messages they receive in the same way. Part of the corporate social network model is to give some messages preferential treatment over others, i.e. there is a noticeable bias towards those that pay. People only have so much attention they can spare every day, and the companies decide what you cannot skip based on what they get paid. This would be equivalent to you always seeing the newsletter from Coca Cola at the top of your email client, but only half of the emails from your father or local charity because they are automatically put in a folder out of sight. This "pay to play" creates a knockout race for attention fueled by commerce, not by arguments, emotions, ethics or societal considerations.

This exposure is worsened by the fact that the platforms monetize your data and behaviour. Social media companies create fine-grained personal profiles, that even include attributed political, relational and other deeply personal matters. By clustering people, profiles becomes more crisp and valuable. But they tend to push people step by step to more extreme options. You liked marijuana. You like drugs. Maybe

you like cocaine? You visited a site with conspiracy theories. Well, here is another one which is even more incredible. When these profiles are made available to advertisers at a premium price, psychometrics such as used by Cambridge Analytica (and others), these allow to influence subsets of the population in both subtle and crude ways.

These selfish business practices continuously raise fundamental societal questions: how do we feel about social media being used by foreign state actors to influence democratic elections through very personalized (and misguided) political campaigns? And how do we contain the algorithmic pressure towards global extremes, rather than brings people together as one would expect from a social network?

Another problematic issue to address is monoculture. Social networks do not allow to cross the boundary of their service in an easy way, leading to social lock in and a "winner takes all" scenario. This limits choice, but also exposes users to legal dangers. Confidential discussions through "private" messages for instance turn out to be not so private, such as the case where a United States got the social network Twitter to hand over the personal communication from European human rights activists and a member of the Icelandic parliament over a severe human rights violation by the USA military. The European Court of Human Rights would certainly not have allowed this, but it happened outside of our jurisdiction - even if all the actors never left Europe.

The federated universe, abbreviated to fediverse, wants to offer social media users a more transparent, ethical and decentralized environment to talk, find and connect. This is done through a plethora of completely independent servers hosted by organisations and individuals around the world. Each has their own policy, each has their own community and reputation. But they can all interoperate. If you don't like any of the existing options, or want to do something different or innovative, you download some open source software and start your own. If you feel some server is toxic, or misbehaves, it just takes one click to stop listening to what is being said. And there is no need to share data with anyone, if you want to. Every node can essentially be a complete social network in itself.

The fediverse is not confined to what a single company wants to do - in every way. That means a broader offering in terms of design, usability and user experience, in terms of technology, ethics and culture. Essentially every server is a full-fledged social network in itself, able to talk to other social networks when it wants. People can use the fediverse for traditional social networking, but they can also integrate it with other services such as online video sharing, all without the fear of having their data being monetized or their activity profiled. Switching from closed social networks to the fediverse contributes to privacy and trust, by enabling users to understand and control who sees their data. The fediverse as a network of social networks, is also more resilient than a single network could ever be.

Lemmy is an open source tool that helps users discover what the fediverse has to offer as a decentralized alternative to for example Reddit. Everyone can host their own instance of Lemmy, determine their own moderation policy to keep discussion as civil as you would like and let users share, post, vote and interact without any corporate interference, all from the comfort of their server of choice. Search and discovery on the fediverse becomes easier, more fun and social, without forgoing independence and agency.

Technical description

Lemmy is an open-source, easily self-hostable link aggregator that you can use to share and discover interesting new ideas - and discuss them with the world. Its designed to work in the Fediverse, and communicate natively with other ActivityPub services, such as Mastodon, Funkwhale and Peertube.

Lemmy aim to create a decentralized alternative to widely used proprietary services like Reddit. For a

link aggregator, this means a user registered on one server can subscribe to communities on any other server, and have discussions with users registered elsewhere. The front page of popular link aggregators is where many people get their daily news, so Lemmy has the potential to help alter the social media landscape.

Visit https://NLnet.nl/project/Lemmy

NGI0 Discovery **ActivityPub** **LinkAggregation**

MeiliSearch

Search and discovery is one of the most important and essential use cases of the internet and information society in general. When you are in school and need to give a presentation, when you are looking for a job, trying to promote your business or finding relevant commercial or public services you need, most of the time you will turn to a search bar on- and offline to find answers. Searching information is crucial for users, but they actually have little control over how it precisely works. Search engines set the terms for what results you see, how your website can be discovered and what information is logged about your searches. What terms are set remains obscure for users and they can only follow the rules laid out for them, instead of deciding on their own what, where and how to find the information they are looking for.

What if instead of search and discovery as a closed box, we use transparent technology that like building blocks, we can put together to make tools that serve us best? MeiliSearch is such a customizable building block, and a powerful one too: as a so-called instant search engine users can search-as-they-type, speeding through indexes and data in milliseconds, all the while being simple and accessible to use for anyone who wants to make their own search solution. This project aims to push uptake of this promising technology in as many programming languages, platforms (think Android) and mainstream applications like WordPress and Drupal as possible, making a lot of search tools much faster with reliable, transparent technology.

Technical description

Advanced content search for apps and websites has become an increasingly protected craft. When owners of big content repositories need search at scale, they have to choose between hiring expensive search specialists or outsourcing search in its entirety. Search doesn't need to be this complicated. It should be simple enough to be self-hosted with the developers you already have, and it should be understandable & open enough that you can resort to a managed cloud without fear of lock-in.

MeiliSearch is blazing fast and very light on resources. It packs advanced search capabilities like search-

as-you-type, relevancy , typo-tolerance, synonyms and filters, all set up and configured in minutes. Our primary path to widespread adoption is integration with other developer ecosystems. Every new language, framework, platform or application that's supported brings in a new audience of developers that wouldn't otherwise know we even exist.

Meili — **Visit https://NLnet.nl/project/MeiliSearch**

NGI0 Discovery CMS Database Integration SDK SearchEngine e-Commerce

---

K a z a r m a

Social media are important for many people to stay in touch, but they also influence the way we act ourselves. Their persistent presence in our lives creates a lot of pressure to be someone else then we are. This is not a coincidence: the developers have designed the software to keep us 'engaged' at any cost. If we sleep less well because of a message we got just before we went to sleep, that is fair game. We may wake up with a hundred messages we "missed", which creates a lot of stress. But it keeps us engaged, and keeps the advertising cash flowing.

Another problematic issue to address is monoculture. Social networks do not allow to cross the boundary of their service in an easy way, leading to social lock in and a "winner takes all" scenario. This limits choice, but also exposes users to legal dangers. Confidential discussions through "private" messages for instance turn out to be not so private, such as the case where a United States got the social network Twitter to hand over the personal communication from European human rights activists and a member of the Icelandic parliament over a severe human rights violation by the USA military. The European Court of Human Rights would certainly not have allowed this, but it happened outside of our jurisdiction - even if all the actors never left Europe.

Problems of social engineering, toxic spaces and surveillance can be addressed by taking matters into our own hands: organizing social media and communication yourself, using open and decentralized solutions precisely as you want it. That is what the protocols ActivityPub and Matrix respectively offer. ActivityPub allows independent servers, each hosting their own social network, to talk to each other on the same level and let users chat, share and interact seamlessly. Matrix offers equally decentralized communication that for users, can be compared to email (but a whole lot safer). You can talk to anyone regardless of what app or server they are using. As these decentralized protocols already build bridges between independent networks, servers, apps and users, now the time has come to link Matrix and ActivityPub together, which is what this project will aim to do.

Technical description

Matrix-Appservice-CommonsPub is a bridge between two decentralized protocols: Matrix and ActivityPub. The development includes polishing CommonsPub, an Elixir generic ActivityPub implementation, and creating an Elixir library to build Matrix bridges. We will first focus on private messages between Matrix users and users of an ActivityPub-enabled platform, like PeerTube or

Funkwhale, then explore the possibilities of synchronizing ActivityPub feeds (e.g. "toots" feeds) in Matrix. The bridge comes as an easy-to-deploy, secure and scalable solution.

Etape 2 (non-profit), physically based at the Fuz hacklab in Paris — **Visit https://NLnet.nl/project/Kazarma**

NGI0 Discovery **ActivityPub** **Bridge** **Library** **Matrix**

Hubzilla

One of the oldest questions on the internet is: how do you adequately prove you are you? Or perhaps the reverse formulation offers a better mental model: how do you prevent others from succeeding in pretending they are you? Now lets flip this question around once more: how would you like to see this managed yourself, if you could? How heavy-weight or convenient do you want to be proven that you are you, to allow you to get into your own environment or have something done on your behalf? And what is it worth to you in terms of effort? Would you be willing to spend a minute to have some clever secure device you have in your pocket involved? Authenticate via your mobile phone? And what if you are in a rush, or on the go? Are you happy with some company like your email provider or a large social network having the ability to make that judgement, based on a user login a few hours ago? And what if that company is based in some other jurisdiction, and could be forced to let others in as well? Or would you rather choose your own identity, and formulate direct rules to have complete control at any given point?

Instead of handing over all your credentials to some company that then sets the rules for how you authenticate yourself, or having countless accounts and passwords that get lost (or hacked), how about you simply own your identity? Something like a passport that you use to prove that you are who you are and when you want to leave a forum, or stop a particular service, you put that passport back in your pocket and move on.

This is what Hubzilla makes possible: technology that lets independent websites connect to each other and users move around freely, carrying their own identity with them. On top of this infrastructure you can make all sorts of apps, build a webshop, interactive chat, whatever you want to make a rich, collaborative and community space, for example for your school or local business. Users can seamlessly move across sites, access files, send each other messages, with highly detailed permissions for what they can and cannot do and in complete control of their identity. This way communities can organize themselves and chat, trade, collaborate independently and on their own terms.

Technical description

Hubzilla is one of the most mature stacks within the so called Fediverse, and is able to run different

protocols such as ActivitPub, Diaspora and Zot. Hubzilla provides powerful tools for communities and individuals to help organise themselves, while providing a possibility to interact with each other. It is a decentralised identity, communications and permissions framework built, using common webserver technology. The software features many useful apps to enable discussions, event organisation, file sharing etc. with built-in internet-wide access control. With Hubzilla you don't have an account on a server, you own an identity that you can take with you across the network.

With the help of the NGI Zero grant, the new version of the zot protocol (zot6) will be implemented as the primary communication protocol and the UX/UI will be improved to lower the entry barrier for less experienced computer users. And of course you can easily search your Hubzilla server for topics, users, fora and tags.

Visit https://NLnet.nl/project/Hubzilla

NGI0 Discovery **ActivityPub** **SocialMedia** **SocialNetwork** **Zot**

Connect by Name

You mobile phone doesn't really understand what for instance "NLnet.nl" or "www.wikipedia.org" mean, when you type either name into a web browser. Being a web browser, it will not come as a surprise that the software will assume you want to visit some website. But it doesn't really know where that website is located on the internet. It doesn't need the physical place of course, but it needs the number that unique identifies the web server so it can connect.

All your mobile phone does know, is how to ask that question to other, specialised computers. These computers actually also probably don't know, unless they have recently answered the same question for another user. Names can change really fast for good reasons, so you would need to refresh this data a lot - otherwise users would end up on the wrong computer. The computers you send your question to, will have a good working understanding how the so called "domain name system" of the internet works. More in particular, the name we asked for needs to be cut up in smaller pieces that need to be read backwards.

There is a short code at the end, which points to a country - or provides some other meaningful clue as to where more information can be learned about the still unknown parts of the name. The short code (which people tend to call a "top level domain") is uniquely managed by a single professional organisation. It is actually called a registry because that is literally what it does: it registers all the names people use. One organisation registers names which end in ".nl", others take care of ".org" or ".eu". There is an invisible list that has all the top level domains on it. This list is called the "root zone" of the internet, and it is quite important because everything that uses a name will need to start its search there.

It is the registry organisation which can provide additional details about the segment next up, in this case "wikipedia" or "NLnet". But it will still not know all the answers itself, so your question will travel to yet more computers. We are getting close now to the computers that these organisations have selected to take care of their domain name. In the case of NLnet this computer will be able to give the right answer straightaway, and this answer needs to be sent back across the entire chain of computers. In the case of wikipedia, the fact we still have a "www" part to look for, could mean that inside Wikimedia foundation there would still be another computer which could be responsible for everything under that label. The

same could go for fr.wikipedia.org or ro.wikipedia.org - the label www is only meant for human consumption, but computers actually don't need it. After just a few steps, we started getting part of the answer we were looking for, and all of these parts are sent back to your phone. And at some point in time, we have the entire answer.

Now how do we know that the answer we obtained in this recursive way really can be reliably traced back to the right computers running the root zone of the internet - the so called root servers? Simple, because there are digital signatures on each part of the answer. For the root zone, there is a so called cryptographic key which is distributed widely - there is only one for the whole world. Chances are you have that key on your phone or computer, and your internet provider certainly has. When the question arises where .org is, this digital signature will make sure you know the right internet address to go to. There you can ask the organisation that is responsible for the next part of the answer. For each computer that gives another level of detail, new signatures are added. So in the end you should have a complete proof for every step: or in other words, a trust chain.

Those signatures on the answers are really important: your computer has nothing else to underpin trust. If someone is able to falsify these signatures, they could use this to manipulate answers for everything "below". This includes not just domain names, but also other things people have put into the DNS like certificates. So great effort is spent on making sure everything happens in a really safe way, leaving nothing to chance. And as a matter of technical hygiene, the cryptographic key needs to be changed regularly. For the root of the internet, there is in fact a grandiose ceremony which involves flying in people from all over the world to closely watch how the keys are replaced. The event is attended by journalists and observers. Of course this kind of public event is really expensive, but there is only one root zone of the internet and it only happens once every couple of years - so it is kind of a special event.

Organisations running a top level domain, also need a thorough procedure. They may not have the same budget, however. True, some of the larger organisations may have multi-million euro annual budgets, but others certainly do not. So far there was not a canonical procedure shared among these organisations, meaning that there was room for ambiguity and misinterpretation that could have serious consequences for the economy and society alike. Also, policy makers responsible for national and regional policies were unsure what was expected from them.

Luckily, there are seasoned experts hard at work to develop tools and services that make such procedures easy, or even better, practically invisible. That is what this project contributes to with a new interface for software developers that need to connect their app or program to the internet. You simply state what domain name your service connects to and the interface makes sure any lookup follows the trust chain through a secure, private connection. This way following best practices and adhering to existing standards becomes 'plug and play', something that happens under the hood and any app or technology can simply plug in and use, ultimately making internet connections everywhere a lot safer.

Technical description

Connect by Name will be a C library providing an interface that allows a software developer to setup internet connections from an application in the most private and secure manner using well-established and open standards. The interface provided to the software developer will be as simple as “Connect to a service on a domain name” and be flexible enough to fit with different programming paradigms and environments. The library will facilitate composability with other systems and will be extensible with future standards. Our goal is to lower the barrier for developing high-quality software and thereby improve the security and privacy of end users.

NLnet Labs — **Visit https://NLnet.nl/project/ConnectByName**

NGI0 Discovery DANE DNSSEC DoH DoT Foundation IPv6 Library

M P T C P

If you want to share your message (or data) with the world, you send a packet that travels across a great deal of the networks that make up the internet to finally reach its destination and deliver someone the file, video or software they were looking for. There are actually quite a number of routes your data can take and ways to deliver messages over the internet, each with their own ups and downsides. The Next Generation Internet intends to create a more private, resilient and decentralized internet. One of the ways to reach this goal is by making internet routing itself more privacy-friendly, fault-resistant and decentralized.

Instead of following one path, your connection could be faster and more secure by using several routes, like for example the 4G on your phone as well as available wifi. Next to speed and resilience, this multipath approach also makes it harder to snoop on your internet traffic, as it is spread over more than one path. To make multipath protocols easier to use and more energy efficient, this project will develop tools to analyze and optimize its performance. Once multipath connectivity works as automatically as the single paths we are used to taking online, networks worldwide can benefit from a stronger, more private and seamless online experience.

Technical description

How do you find the best way to communicate with a computer on the other side of the internet? And why bet everything on a single connection? Multipath TCP (MPTCP) extends the most widely used transport protocol on the internet (TCP) so that it can discover and use several physical paths (e.g., Wifi, cellular, between multihomed servers) in parallel. This allows to speed up transfers, smoothly transition from wifi to cellular when leaving one's house or potentially prevent traffic spying.

While the protocol is proven to work well in certain conditions (the fastest TCP connection ever was using MPTCP), it is configuration-sensitive and can degrade badly under adverse conditions (for instance in heterogeneous networks with small buffers). The aim of this project is to provide the tool to help analyze the performance of a multipath protocol as well as the software to (auto)configure the system depending on the application objective and network conditions.

Visit https://NLnet.nl/project/MPTCP

NGI0 Discovery IETF Multipath TCP

In the 'real world', you instinctively know what information you should keep behind locked doors and what is safe to share. Your bank statements are stored in a folder somewhere in the attic instead of leaving them laying around on your kitchen table. You do not tell random people on the street what your phone number is, or where your children go to school. In the virtual world, this type of common sense can work differently.

Users are quicker to trust service providers to keep their personal data safe from theft and prying eyes, and do not always see the dangers of storing passwords in an online text file, or sharing sensitive financial documents via email. The dangers are unmistakably there, but until someone close to you suffers the consequences of a hack or a privacy breach, the risks of online data storage are vague and its convenience is too tempting to pass up.

People are accustomed to easy, accessible and convenient online tools and services. More private and secure open-source alternatives should not exclude users because of an overly technical setup or incompatibility with existing proprietary solutions.

Solid (or Social Linked Data) is a new approach to protecting personal data initiated by Tim Berners-Lee, the inventor of the world wide web and developed in collaboration with the Massachusetts Institute of Technology (MIT). The project aims to give users back full control over their personal data, which they can store in personal online data stores (or pods) and then give applications that run on the Solid platform access rights as they see fit. Users always retain ownership over their data, decide for themselves where it is stored and can change the permissions of any application that can access the data. Eventually the Solid ecosystem should offer decentralized and user-centric alternatives to centralized social media like Facebook, Twitter, LinkedIn etcetera.

Nextcloud is an open source file hosting (cloud) solution that follows the same principles as the Solid project: users are in control over their data, where it is stored, and who can access it. This project will draw a bridge these two efforts and create a Nextcloud app that converts a Nextcloud-account to a Solid-identity. This combines the strengths of both projects, allowing users even more precise control over which people and organizations can access their private data.

Technical description

This project connects the world of Solid with the world of Nextcloud. The aim is to develop an open source Nextcloud app that turns a Nextcloud server into a spec-compliant Solid server. It gives every user a WebID profile and allows Solid apps to store data on the user's Nextcloud account. It also exposes some of the user's existing Nextcloud data like contacts and calendar events as Solid user data, so that Solid apps can interact with the user's Nextcloud data, and allow the user to manage which Solid apps can access which specific aspects of the user's personal data. We will make our implementation compatible with the latest version of the Solid spec (including DPop tokens and the WebSockets AUTH

command), and contribute the surface tests we create for this as a well-documented independent test-suite, for other Solid server implementers to benefit from. We will also publish a stand-alone version of our PHP components, which can run independently of Nextcloud.

Unhosted — **Visit https://NLnet.nl/project/Solid-NextCloud**

NGI0 Discovery **IdentityManagement** **Solid**

C a s t o p o d

The internet can be a great place to discover new content, be it music, movies or podcasts. Cheap hosting and reliable streaming technology lets users listen or watch new content then and there as they discover it, almost like a candy store. Unfortunately major streaming services decide for users what candy they should and should not have, ranking and rating content based on who happens to be the best selling artist of the day. Both users and artists on these platforms are left with little control over search and discovery, content and privacy.

What if instead of handing over independently produced music, videos and podcasts to a few companies, artists and users watch and listen on their own terms? Like Mastodon, Pleroma and Pixelfed, Castopod offers a decentralized alternative to social networking, hosting and sharing content without some platform pulling the strings. Castopod allows users to host their own podcasts themselves and then easily share it for discovery with everyone on the federated social universe, better known as the fediverse. Anyone on Mastodon, Pleroma and elsewhere can seamlessly interact with podcasters on Castopod, follow their favorite channels and keep up to date through feeds. This adds yet another rich layer to the fediverse where users can search and discover content, artists and possible friends entirely on their own terms.

Technical description

Castopod is an open-source podcast hosting solution for everyone, that can connect to the Fediverse through the W3C ActivityPub standard (Pixelfed, Mastodon, Pleroma…). Castopod is user friendly, and allows for easy discovery everywhere. Whether you are a beginner, an amateur or a professional, you will get everything you need: you can create, upload, publish, manage server subscriptions (WebSub embedded server). You can allow users to listen to your podcast directly, but just as easily connect to commercial directories (Apple, Google, Spotify…).

Take back control: interact with your audience on your platform (like, share, comment), the social network IS the podcast. In addition to supporting W3C ActivityPub, you can also export to proprietary social networks (Twitter, Instagram, Youtube, Facebook). Castopod is easily hosted on any PHP/MySQL

server: unzip it and you and other podcasters are ready to broadcast professionally.

Ad Aures — **Visit https://NLnet.nl/project/Castopod**

NGI0 Discovery **ActivityPub** **Podcast**

In-document search

Searching usually starts with a vague memory, a name or number that is in the back of your mind, some little detail that sticks with you, but unfortunately does not tell you where you need to start your search and how. How humans search and how computers handle your query does not always overlap, which can be frustrating: you end up shouting at your screen.

One of the technologies that can make search and discovery more intuitive is semantic search, which is poetically explained as 'search with meaning'. Essentially, instead of searching for a literal number or letter, semantic search tools better understand the context, location, intent, word variations, and other important points you would imply when typing in a query. You do not know what file you are searching for, but you know it has something to do with your upcoming tax report. Or you cannot remember for the life of you that person's name, but you know who their colleagues are. Semantic search takes your vague plan and scraps of information and instantly knows how everyone is connected, giving you the information you were looking for.

Intuitive semantic search requires rich data structures, for example the information hidden in the documents we make. Not only the metadata that states you saved a text file at a certain time in a particular folder, but also more detailed information and connections like what images in your presentation are copyrighted, what links in your reports do not work anymore, etcetera. This project aims to develop these search capabilities for OpenDocument files, a widely used open document standard. Especially when handling a lot of documents and data, such detailed search can be an incredible time saver, or even provide unique new insights, for example in data-centered research or journalism.

Technical description

There is a relatively unexplored layer of metadata inside the document formats we use, such as Office documents. This allows to answer queries like: show me all the reports with edits made within a timespan, by a certain user or by a group of users. Or: Show me all the hyperlinks inside documents pointing to a web resource that is about to be moved. Or: list all presentations that contain this copyrighted image. Such embedded information could be better exposed to and used by search engines than is now the case. The project expands the ODF toolkit library to dissect file formats, and will potentially have a very useful side effect of maturing the understanding of document metadata at large and for collaborative editing of documents in particular.

independent freelancer — **Visit https://NLnet.nl/project/InDocumentSearch**

E D e A

EDeA hub

Open collaboration (like for example on open source software) is based on the premise that together, we know more than we do alone. For open source software development, there is a long history of tools and infrastructure that you can easily setup and maintain for your project, so you can involve as many viewpoints and contributions as you can to make your program versatile, secure, user-friendly, creative, and so on. What's more, the community created around a project can keep software going long after an initial creator has left, updating and expanding it as needed.

For open source hardware, similar tools and spaces unfortunately lack specific features to make open collaboration on devices actually beneficial. Circuit design for example is one of the most fundamental parts of hardware development, yet there are not tools for open hardware projects to design circuits together. This project wants to create such a space, not only to allow creative people all around the world make wonderful new devices, but also prevent designers from reinventing the wheel when they should just pick up an existing open design and get started adding their own unique functionalities, switches and screens. As open source software has thrived because of accessible and usable collaboration tools, now open hardware can take the same leap forward.

Technical description

The short version: EDeA is a novel approach to allow exploration of and improve discovery within the open hardware ecosystem - in order to help make open hardware designs and components discoverable and reusable.

At this moment in time, pretty much everything surrounding open hardware development is manual. Beyond just typing something into a generic search engine there isn't really suitable tooling available to search across what already exists. Accessible and usable distributions, collaboration tools and version control are what drove the free and open source software revolution, now open hardware needs to take the same leap forward.

Open hardware electronics projects are growing in numbers, thanks to crowdfunding, a strong developer community, and sophisticated open source electronic design automation (EDA) tools like KiCad. Between circuit schematic and printed circuit board (PCB) layout there is a logical association, but are being handled by separate programs, and therefore one can’t simply copy-paste design blocks. In 2020 it is still next to impossible to reuse proven parts of different designs without needless reimplementation. By leveraging KiCad’s pcbnew and eeschema scripting, a new way of building modular, reusable electronics opens. We are creating a catalog and community portal for discovery and development of proven circuit modules: power management, signal conditioning, data conversion, micro-controllers, etc.

Fully Automated OÜ — **Visit https://NLnet.nl/project/ElectronicsHub**

NGI0 Discovery OpenHardware PCB Portal

In the new mobile world we live in now, control as a user is limited to the very surface of things. Significant privacy and security issues start directly below that surface. You don't really know what the platform actually does while executing apps, and more importantly, who sees your data - or if you are a business, looks at the data of your customers. When you use one of the hundreds of thousands of existing apps and games, you only see the service they provide. But you can't inspect or even see what more they take. What does an app do exactly when you click on the pretty icon? This is very much unlike for instance interacting with a web page, which is fully transparent. As it turns out, mobile apps do lots of things users do not know about, and would not agree with if they did. In some cases literally hundreds of companies have been known to get access to data on the phone.

A consumer-friendly platform should empower the user to notice and take action, or even make it technically impossible. However, the companies that produce the operating systems seem to have other interests. Have you ever wondered why everyone tells you your desktop computer needs a firewall and you are allowed full control to see everything happen. Now stop and think about why your cell phone does not have the very same level of firewall capabilities, but only very much simplified and less capable?

So what can we as a society do in the face of such a complex situation of market failure, anti-competitive practices, perverse incentives and general confusion? How do we give control back to the users? How do we create equal opportunities for European phone manufacturers? How do we stop the unfair "platform tax" on app developers, stimulating employment and startups?

One reasonable direction is to try and lay the ground work for creating viable alternative platforms. Such a fundamental approach is necessary in order to end these extractive practices and the resulting lack of consumer freedom. This project develops a mobile operating system independent of Android based on the widely popular and open source Linux-system, complete with trustworthy and privacy-focused free (as in freedom) software. Staying clear of device-specific software, postmarketOS gives meaning to its name by ensuring all mobile phones running this operating system can be updated until they physically break.

Technical description

postmarketOS is a mobile phone operating system for phones (and other mobile devices), based on Alpine Linux. Just like desktop Linux distributions, we have a package manager and a carefully crafted repository of trustworthy and privacy focused free software that will actually serve the users and not exploit them for their data. By sharing as much code as possible between various phone models,

postmarketOS scales well and it becomes feasible to maintain devices even after OEMs have abandoned them.

Postmarket OS — **Visit https://NLnet.nl/project/postmarketOS**

NGI0 PET **MobileOS** **Privacy** **Security** **Sustainability**

Nym Credentials

One of the oldest questions on the internet is: how do you adequately prove you are you? Or perhaps the reverse formulation offers a better mental model: how do you prevent others from succeeding in pretending they are you? Now lets flip this question around once more: how would you like to see this managed yourself, if you could? How heavy-weight or convenient do you want to be proven that you are you, to allow you to get into your own environment or have something done on your behalf? And what is it worth to you in terms of effort? Would you be willing to spend a minute to have some clever secure device you have in your pocket involved? Authenticate via your mobile phone? And what if you are in a rush, or on the go? Are you happy with some company like your email provider or a large social network having the ability to make that judgement, based on a user login a few hours ago? And what if that company is based in some other jurisdiction, and could be forced to let others in as well? Or would you rather choose your own identity, and formulate direct rules to have complete control at any given point?

As could be guessed, individual people have a need for different levels of confidence and security in different contexts. A security breach matters perhaps less if you just want to login to a music service to change a playlist. After all, the worst that can happen is that someone messes things up and you have to create a new one. It matters a great deal more if you want to do a significant financial transaction at work, or open the door of your house remotely to let the babysitter in while you are delayed in traffic. Perhaps you can think of scenarios where you want even more control.

So what proof to use as the basis of your trust, and the subsequent actions taken? Historically people rely on some authority they collectively trust. Such an authority has typically taken high tech countermeasures to make the channel through which that trust is conveyed hard to fraud. A passport or banknote are quite tricky to fabricate due to the use of special techniques. Online we have only a very limited amount of trust "anchors" of varying quality. The domain name system is such an anchor, digital certificates or customer relationships are another. Today, having access to a certain mail account or phone which is known to be yours is the most common proof used. Email is often called the "poor man's solution" to identity management, and it is what most organisations and businesses fall back on. Can't log in? We will send you an email to reset your login. Just click on the link. And of course, email was never designed to be safe. It kind of works, but really we can do better.

Perhaps your use cases require more strict proof than that of normal consumers, or less strict proof. Even for a single large service provider, it would be hard to figure this out satisfactorily for all users. For the same reason people write their own testament to document what should happen with things they own or control after they die, you want to document what should happen with things you own or control what

happens when you are physically absent. There is no universal will that is acceptable to all, nor is there a universal policy that satisfies all use cases.

So what if you yourself would be able to create and control your own identity, and determine your own proofs and methods? In order to function in a global internet, you would need to be able to convey your requirements and demands in a portable way. There would be no central authority dictating you what to do here. That would mean you you yourself would have to make things explicit upfront in a foolproof way - so that elsewhere on the internet people and services would know what you expect them to do to distinguish the real you from fraudsters.

This project provides the tools and infrastructure for users to authenticate themselves and share personal data (and proofs of data) without a centralized authority, where your credentials are protected through modern technologies built for privacy and security. Together these tools and infrastructure provide a state of the art European alternative for authentication that puts users (and no one else) in the driver seat.

Technical description

Nym Credentials provides open-source code for privacy-enhanced authentication and authorization in a decentralized environment. Today, when using "single-sign in" solutions, users hand over their personal data to third-party identity providers such as Facebook Connect and Sign-In with Google. Nym Credentials tackles this problem by allowing users to securely authenticate and transfer personal data (and proofs of private data) while maintaining privacy without a centralized identity provider. Each credential is cryptographically unlinkable between usages and multiple decentralized identity providers can verify this data. Open-source Nym credential libraries can be easily integrated into existing services, with a focus on federated and decentralized European environments.

Nym Technologies SA — **Visit https://NLnet.nl/project/NymCredentials**

NGI0 PET **Cryptography** **DistributedLedger** **Library** **Rust**

L i g h t m e t e r

LIGHTMETER

One of the things people enjoy the most about the internet, is that it enables them to talk to others remotely almost without limit. Internet allows anyone to keep closely connected with friends and family, and help their kids solve a math problem while they are at work. People collaborate with their colleagues from the couch of their living room, the cafe where they enjoy lunch or on their cell phone on the bus to the gym. Businesses can easily service their customers where this is most convenient to them, without having to travel themselves. This is so convenient, that some businesses have already moved entirely online. Internet communication has become the nerve center of whole neighbourhoods, where people watch over the possessions of their neighbours while these are away for work or leisure.

Email to this day is among the most popular online communication services and is used by governments, companies and organizations to talk to clients and share files. Even though email was designed without

privacy or security in mind. When you send an email, anyone that can gain access to your mail server or the mail server of the recipient can read your mail, from top to bottom. And copy it, for later usage. Or modify it. It is often compared to sending a post card, and of course in many cases there may be little harm in others reading what the weather is like in Athens. But what if you want to use email to send something confidential, something you do not want to share with others? Like a love letter, a political rant or an important contract? And what if you can't actually trust the mail man, for instance because the other party is using a free email service known to search through everything? Or what if you live in a country that has an unhealthy interest in bringing down certain political voices, or are part of a cultural minority that is at risk?

Users could try to host their own email server, or if they are not so technically inclined switch from one of the 'free' email providers (that are usually after your (meta)data and are known to read your messages) to hopefully more trustworthy independent parties that simply charge a monthly fee and in exchange, keep your email safe, private and abuse-free (no spam coming from your address, for example).

Most email servers rely on open source mail handling software that is extremely configurable but also quite old and not developed to solve the many privacy and security issues of email, one of which is whether your mail is delivered correctly. This is an important question to answer, for example if some important legal document was attached, or you sent someone your password (both of which you should not do, but happens everyday). This project helps mail providers make sure that email is not lost anywhere with open source tooling to monitor delivery. Users and independent hosts are given more control over how they can protect and control their email this way.

Technical description

Lightmeter will make it easy to run email servers large and small by visualising, monitoring, and notifying users of problems and opportunities for improved performance and security. People will regain control of sensitive communications either directly by running their own mailservers, or indirectly via the increased diversity and trustworthiness of mail hosting services.

Lightmeter — **Visit https://NLnet.nl/project/Lightmeter**

NGI0 PET DMARC Email SPF SelfHosting

GNU Mes: Full Source bootstrap

When you start up your computer, you will probably think twice before you download some random piece of software from the internet and run it. You know that doing so could allow unwelcome guests to your computer and your data. Your computer might even end up in a bot net. So when you see some nice piece of software, you will ask yourself the question: can I really trust the software? Perhaps you will check the origin it comes from. Better safe than sorry.

Did you miss checking something, though? What about the software that is already on your computer before you started? A computer is not much use without an operating system. While most computers are sold with an operating system, actually you have the choice to remove that and install something different. Have you thought about the trustworthiness of that fundamental piece of software - your most fundamental travel companion on the wild west of the internet? Trustworthiness is essential. When an operating system has a so called 'back door' (either intentionally or not), someone could extract whatever user data - like personal pictures or home movies - from your computer. And the worse thing: without you ever finding out. The operating system guards all the other software, and warns you when you install software from the internet. But itself, it doesn't have to ask for permission. Ever. It doesn't just have "access all areas": in fact, it runs the whole show.

With commercial software like Microsoft Windows or Mac OS X that you get delivered when you buy a computer, trust in what their closed operating system does will of course always be a leap of faith: as a user you essentially are given no choice. In proprietary systems you do not have the freedom to study the source code, or to control what really happens. So you either trust the vendor, or you'd better not use it. For an increasing amount of people, after the revelations from whistleblowers like Edward Snowden, that "leap of faith" is not so obvious anymore. They prefer to use free and open source operating systems like GNU Linux, FreeBSD and OpenBSD. These are technology commons: the people that wrote the software allow you to inspect the source code. Even more so, they give you the source code to do anything with it that you like. So you don't just blindly have to take their word for it and trust them, you can take matters into your own hands.

But until now, there were some parts that would escape introspection. You would have to trust them, not because the people involved didn't want to share everything with you - but because they couldn't. When an operating system is loaded, you need to get the computer into a state from where it can manage itself. The necessary software is poetically called a "binary seed", because it is, well, a very very long string of bits. In fact, a few hundreds of millions of bits. And of course, that amount of information without any hints or cues as to how they interact are rather hard to grasp - and thus a potential point of risk.

What if we could get the computer into the right state through a different path? The GNU Mes project aims to replace the traditional "binary seed" by something orders of magnitude smaller. The really clever and innovative part is that they will add the more complex parts to a "second stage", which is being created from scratch by the project, in a human understandable programming language. This two stage approach allows to make all of computing more trustworthy, in a very controlled way - and will grant our future selves the ability to use computers without taking a leap of faith. If the project succeeds, it will make a very fundamental contribution to the security of the next generation internet. NGI Zero has funded this project before and continues to support the work done.

Technical description

GNU Mes was created to address the security concerns that arise from bootstrapping an operating system using large, unauditable binary blobs, which is common practice for all software distributions.

Mes is a Scheme interpreter written in a simple subset of C and a C compiler written in Scheme and

comes with a small, bootstrappable C library.

The Mes bootstrap has greatly reduced the size of opaque binaries that were needed to bootstrap GNU Guix, a functional GNU/Linux distribution that focusses on user freedom, reproducibility and security.

That reduction (from ~250MB to ~60MB) was achieved by first replacing GNU Binutils, GNU GCC and the GNU C Library with Mes. The second step was funded by NlNet and replaced GNU Awk, GNU Bash, the GNU Core Utilities, GNU Grep, GNU Gzip, GNU SED, and GNU Tar with a more mature Mes, Gash and Gash-Utils.

The final goal is to help create a full source bootstrap for any interested UNIX-like operating system and non-intel architectures. This funding will enable us to take another big step forward and reach an important new milestone in creating more auditable secure software distributions.

joy of source — **Visit https://NLnet.nl/project/GNUMes-fullsource**

NGI0 PET **ArchitecturePortability** **Bootstrap** **NGI0**

JavaScript Shield

As you fire up your computer, laptop or smartphone and click your browser icon to connect to your favorite site, do you know what happens behind the scenes? Modern websites offer their users a ton of functionalities, but it is becoming increasingly difficult to know just how all these slick graphics, popups and interactive elements actually work, and what they do precisely. This is very true for most users, but even those more technically inclined may not be entirely sure what happens on their browsers exactly. Not because they lack the knowledge or tools, but because a lot of these little bits of software that come with visiting particular websites are not transparent.

Simply put, you open a site, your browser is sent some programs that immediately run on your computer and you do not and cannot know what is going on. This poses many problems, not just for user agency and freedom, but also for privacy and security when we have some unrecognizable piece of software from some unknown source run on our system, that might hold sensitive personal data or run vital services. Your browser may know how to protect you from harm, but would it not be better to go straight to the source and make sure we can actually trust what we run?

One of the ways to make sure we can browse the web more privately and securely is to actually understand the programs that websites request our browsers to run. This project is one part of the core mission of the Free Software Foundation's to make all software free. Free as in freedom, not free beer: users should be able to study how software works, modify it and share it. If not, software not only limits your freedom, it can also be insecure and harm your privacy. There simply is no way of knowing.

This is why the FSF intends to take a browser add-ons that protects you from threats on the web and make it into a control room for users: with the push of a button, you can allow, block or spoof requests coming from some program that a web server sent you. Instead of a simple blocker tool that can make browsing a pain, users have granular control over what a website can and cannot do, and in the meanwhile learn the benefits of software that is transparent (and the many downsides of programs that

are not).

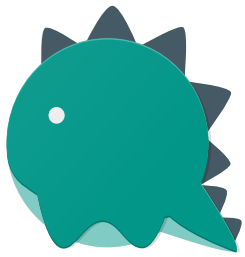
Technical description

The Internet is vital to the everyday lives of billions of people. That's why it's especially problematic that, in the course of using the Web, even from an otherwise fully free machine, browsers run nonfree programs that are outside the control, and even awareness, of many users. These programs run behind the scenes -- but on the user's system -- whenever the Web server says to run them. They are typically served to the user as minified JavaScript, and few provide the corresponding human readable source code, or a free license allowing users to lawfully inspect and modify the program. By definition, these programs infringe user freedom. In practice, this also means they pose serious threats to users' privacy and security -- such as by surreptitiously using a user's CPU to mine cryptocurrency, or by capturing and manipulating keystrokes. The Free Software Foundation is working to make all JavaScript on the Web be free software; its JavaScript Shield project is a freely licensed anti-malware browser add-on to limit potential threats from JavaScript, such as fingerprinting, tracking and data collection. It would ask -- globally or per site -- if specific native functions provided by the JavaScript engine and the DOM are allowed by the user. It would also link to an explanatory page for each function, to raise awareness of related threats. Depending on the function being addressed, the user would have the option to allow it, block it, or have it return a spoofed value. This extension will help protect users from critical threats now, and contribute significantly to progress on the necessary longer-term cultural shift of moving away from nonfree JavaScript.

Free Software Foundation — **Visit https://NLnet.nl/project/JavascriptShield**

NGI0 PET **BrowserExtension** **Fingerprinting** **Foundation** **Javascript**

D i n o



One of the things people enjoy the most about the internet, is that it enables them to talk to others remotely almost without limit. Internet allows anyone connected to keep closely connected with friends and family, and help their kids solve a math problem while they are at work. People collaborate with their colleagues from the couch of their living room, the cafe where they enjoy lunch or on their cell phone on the bus to the gym. Businesses can easily service their customers where this is most convenient to them, without having to travel themselves. This is so convenient, that some businesses have already moved entirely online. Internet communication has become the nerve center of whole neighbourhoods, where people watch over the possessions of their neighbours while these are away for work or leisure.

However, users have a hard time to understand how privacy is impacted if they use the wrong technology. Because internet works almost everywhere, the natural privacy protection of the walls of a house, a school or an office is gone. Unlike the traditional phone companies, many of the large technology providers run their business not on delivering an honest service but on secretly eavesdropping on their users and selling information to others. It is mostly not about what you say, so it is relatively easy for providers to allow some form of privacy by encrypting messages. The more interesting parts are who talks to whom, when, and where they are in the real world while they meet on the internet. if you want to be reachable across the internet, you have to constantly let the communication provider follow you wherever you go. This makes the private and professional lives of citizens an open book to companies that with the help of AI and other technologies make billions from selling 'hidden data' normal people are completely unaware of even exists. And of course in societies that are not so democratic, this type of information is critical to bring down opposition and stifle human rights.

Users assume the confidentiality and privacy when they communicate, and they are morally justified to do so. There is nothing natural or final about internet communication providers having access to all this very personal information - or going down the dark path of selling data about customers. The cost of this in terms of internet usage and computer power needed is actually negligible, and so all it takes it the availability of open alternatives that people can use.

Dino is an open source messaging client that makes sure your privacy and security are guaranteed using open standards and existing technology. This project will add an important missing part to the puzzle, namely encrypted audio and video calling between two or more people. Since this new feature will also be built on existing standards, other similar messaging clients can benefit from this work and better protect their calling capabilities as well.

Technical description

Dino is an open-source messaging application. It uses XMPP as an underlying protocol, which allows federated, provider-independent communication and offers a world-wide network of interconnected servers. Dino aims to be secure and privacy-friendly while at the same time offering a good user experience and a modern feature set. This project will add encrypted audio/video calling functionality between two or more parties. The implementation will rely on existing standards to interoperate with other XMPP applications.

Dino Team — **Visit https://NLnet.nl/project/Dino**

NGI0 PET **AudioCall** **Encryption** **InstantMessaging** **Jingle** **Videocall** **WebRTC** **XMPP**

---

x86-64-bit Virtual Machine Monitor for seL4 verified microkernel

Security. Performance. Proof.

How can you understand and trust a complex system, like the operating system managing the hardware and software on your computer? You can make the complexity simpler by cutting it up into parts, compartmentalizing what does what, where information is stored, which processes talk to each other. This way users can be sure their system only does what it is supposed to do and know precisely what goes in and what comes out. This can be done through virtual machines, which are isolated simulations of operating systems or programs on a computer. Simply put, you create virtual rooms where only one thing happens and only you have the keys to each door. This can give users complete control over what happens on their computer and ensures that if some malicious software finds a way in, it cannot get to the other rooms. This can be very important if your device contains sensitive information, if some ill-meaning third party tries to listen in, or when the device is part of some crucial infrastructure and is targeted for attacks.

The Qubes operating system is a pioneer in creating an isolated yet workable desktop. Users can segment programs and data into separate cubes, based on trust. The default cubes are 'work', 'personal' and 'untrusted', that are each run in an isolated virtual machine. If you open a phishing email in your 'untrusted' cube and malware manages to make its way into this specific environment, it cannot get to 'personal' or 'work' and therefore cannot compromise that data (or the entire operating system, which is the case with popular operating systems like Windows that have a huge attack surface). Various colors (think green, yellow, red) can be used to indicate what window and program works in what qube.

Security by isolation can and should be a great way to make operating systems more secure by design. Unfortunately even operating systems like Qubes need other programs to work that may be insecure (and have actual reported vulnerabilities). This project will make Qubes-like systems more secure by switching from a vulnerable dependency to a verified and well-maintained alternative.

Technical description

The security of any software system depends on its underlying Operating System (OS). However, even OSes such as Qubes, which are "reasonably secure" depend on large trusted computing bases (e.g. hypervisors) with hundreds of thousands of lines of code. For example, the Qubes' Xen Security Advisory Tracker reports that 53/283 (18%) of Xen vulnerabilities over the last eight years affected Qubes. As a step towards facilitating the implementation of more secure, Qubes-like systems, we propose to retarget it to the seL4 microkernel. seL4 is an open-source, formally-verified microkernel that has matured and been maintained for over a decade. seL4's small size (10,000 Lines of Code) and formal verification make it an appealing Xen replacement for Qubes, however, its virtualization support is currently limited. As a first step to enabling Qubes on seL4 we will implement a hardened, open-source, x86 64-bit Virtual Machine Monitor (VMM) for the seL4 microkernel capable of hosting the core Qubes OS virtual machines.

Data61 — **Visit https://NLnet.nl/project/seL4-64bitVMM**

NGI0 PET **FormalVerification** **PublicSector** **Virtualisation**

ArtistHub

Technical description

The Artist Hub is a progressive web app developed by The Creative Passport MTU, that allows users - Music makers - to connect different data sources and display their feeds all in the same global wall arranged in chronological order. Music makers will be able to create a custom fan page on a self-hostable server where all their music and related content can be placed and shared with their fans.

The underlying architecture for subscribing to and receiving posts/updates from connected services will be built using ActivityPub. The idea behind this architecture is a free and open-source way for music makers to share their content without needing to post to a number of different websites and social media and for fans to have the freedom to choose their platform of choice for engaging with that content.

We will use ActivityPub to aggregate data from a number of platforms. This will enable us to offer support for video (using PeerTube), audio (using Funkwhale), images (using PixelFed) and text (using Mastodon).

Creative Passport MTU — **Visit https://NLnet.nl/project/CreativePassport**

NGI0 Discovery **ActivityPub** **DataManagement** **SocialNetworking**

URL Frontier

Technical description

Discovering content on the web is possible thanks to web crawlers, luckily there are many excellent open source solutions for this; however, most of them have their own way of storing and accessing the information about the URLs. The aim of the URL Frontier project is to develop a crawler-neutral API for the operations that a web crawler when communicating with a web frontier e.g. get the next URLs to

crawl, update the information about URLs already processed, change the crawl rate for a particular hostname, get the list of active hosts, get statistics, etcetera. It aims to serve a variety of open source web crawlers, such as StormCrawler, Heritrix and Apache Nutch.

The outcomes of the project are to design a gRPC schema then provide a set of client stubs from the schema as well as a robust reference implementation and a validation suite to check that implementations behave as expected. The code and resources will be made available under Apache License as a sub-project of crawler-commons, a community that focuses on sharing code between crawlers. One of the objectives of URL Frontier is to involve as many actors in the web crawling community as possible and get real users to give continuous feedback on our proposals.

DigitalPebble — **Visit https://NLnet.nl/project/URLFrontier**

NGI0 Discovery **API** **CrawlFrontier** **Crawler**

Record Federation for Corteza Clouds

Technical description

Corteza is a low code platform for building cloud-based web applications. This is typically for private, records-based management purposes (e.g. case management, insurance claims processing, public sector management applications, CRM, ERP), but the uses can also be public if required. It has a modular architecture and its data later, presentation layer and automation layer can each be treated individually. Corteza Record Federation makes innovative use of the ActivityPub standard to describe how content from the Corteza data layer can be broadcast across large federations of Corteza clouds. All data types, simple or compound, entire records and entire data models are supported.

Whether it be energy, finance, health, education or smart cities, many industries need to share complex data in real-time or near real-time, while preserving the digital sovereignty of a large number of disparate actors, protecting the privacy of user data and acknowledging the law of whichever territories in which they find themselves operating. Corteza Record Federation allows for the creation of private networks of decentralised “mini-clouds”, all self-hosted and controlled by their owners, where this data exchange can happen as efficiently and more effectively than on any single centralised cloud.

Crust Technology Ltd. — **Visit https://NLnet.nl/project/CortezaActivityPub**

NGI0 Discovery **ActivityPub** **Federation** **LowCode**

XWiki ActivityPub

There are all sorts of places on the web where people come together to share knowledge and store information for others to benefit from. Whether you are documenting the internals of vintage cars, collect knowledge of procedures within an organisation or project, or gather technical specifics of software - a very common way of empowering everyone to collaborate is a wiki. A wiki is a website users can edit themselves. You will surely know the free community-backed encyclopedia Wikipedia, but there are many other instances that bring together a wealth of communities. On a wiki, people can effectively organize their own knowledge base, decide how their information is organized and linked, making it easily findable. Wiki's are used by organizations, governments and businesses everywhere, sometimes storing data essential for everyday operations, or with sensitive credentials. Some cities have their own wiki's, containing rich localized content useful for inhabitants, shop owners and tourists.

To make a wiki work, you need active and involved users. Xwiki is a mature free and open source platform that allows organizations to create their own knowledge base, extending and modifying how their wiki works as they please. Extensibility is essential, which is why Xwiki in this project wants to connect itself to the larger federation of decentralized social networks, also known as the federated universe or fediverse. Connecting to content and interacting with users of for example Mastodon, Nextcloud and PeerTube makes Xwiki an even richer wiki platform, allowing all sorts of useful extensions of your knowledge base, website, or collaborative intranet using Xwiki. And because the project is built on open source software and protocols, other communities can learn from these efforts to tie all sorts of public and hidden treasure troves of knowledge together precisely how they want to, while staying in control over their social data and the information they want to share online.

Technical description

XWiki is a modern and extensible open source wiki platform. XWiki is the first wiki that is part of the larger federation of collaboration and social software (a.k.a. fediverse), allowing users to collaborate externally. XWiki is embracing the W3C ActivityPub specification. Specifically we're implementing the server part of the specification, to be able to both view activity and content happening in external services inside XWiki itself and to make XWiki's activity and content available from these other services too. A specific but crucial use case, is to allow content collaboration between different XWiki servers, sharing content and activity.

XWiki SAS — **Visit https://NLnet.nl/project/XWikiActivityPub**

NGI0 Discovery ActivityPub Commenting Wiki

Practical Tools to Build the Context Web

NGI ZERO

Technical description

In a nutshell, the Perspectives project makes collaboration behaviour reusable, and workflows searchable. It provides the conceptual building blocks for co-operation, laying the groundwork for a federated, fully distributed infrastructure that supports endless varieties of co-operation and reuse. The declarative Perspectives Language allows a model to translate instantly in an application that supports multiple users to contribute to a shared process, each with her own unique perspective.

The project will extend the existing Alpha version of the reference implementation into a solid Beta, with useful models/apps, aspiring to community adoption to further the growth of applications for citizen end users. Furthermore, necessary services such as a model repository will be provided. This will bring Perspectives out of the lab, and into the field. For users, it will provide support in well-known IDE's for the modelling language, providing syntax colouring, go-to definition and autocomplete.

Real life is an endless affair of interlocking activities. Likewise, Perspectives models of services can overlap and build on common concepts, thus forming a federated conceptual space that allows users to move from one service to another as the need arises in a most natural way. Such an infrastructure functions as a map, promoting discovery, decreasing dependency on explicit search. However, rather than being an on-line information source to be searched, such the traditional Yellow Pages, Perspectives models allow their users (individuals and organisations alike) to interact and deal with each other on-line. Supply-demand matching in specific domains (e.g. local transport) integrates readily with such an infrastructure. Other patterns of integrating search with co-operation support form a promising area for further research.

Meneer Zelf B.V. — **Visit https://NLnet.nl/project/InPlace**

NGI0 Discovery Collaboration Federation Modeling P2P Runtime Search

---

Namecoin: Core Infrastructure

namecoin

Technical description

Namecoin is a blockchain project that provides a decentralized naming system and trust anchor. Our flagship use-case is a decentralized top-level domain (TLD) which is the cornerstone of a domain name system that is resistant to hijacking and censorship. This project is meant to improve the security and usability of core components of Namecoin.

The Namecoin Project — **Visit https://NLnet.nl/project/NamecoinCore**

NGI0 Discovery Blockchain DNS Decentralisation Naming

Mangaki

Technical description

Within a set of search results, what should you do to find the optimal solution for not just a single user but a group? Mangaki is building an open source library for privacy-preserving group recommendations of items. While many content providers suggest recommendations at a personal level, these are often directed to a single user, or are restricted to a generic “family” category. Whenever say a group of friends want to watch a movie, it is often hard to decide what to watch, because people can have really different tastes.

Recommendations are also very privacy-sensitive. A straightforward way might be to share our complete viewing history, but that certainly can lead to embarrassing and awkward situations. So how can we collectively compute a list of relevant items without disclose all of our data unencrypted. The Mangaki project is making an open source library for group recommendations that works in a scalable and distributed way.

Mangaki/Inria — **Visit https://NLnet.nl/project/Mangaki**

NGI0 Discovery Library Recommending

Inventaire

Books are still a phenomenal carrier of human knowledge - immersing us into deep thought constructs, and their creative expression allows us to investigate and perhaps understand some of the complexity and richness of the world around us, but also to execute thought experiments

beyond the boundaries of the realistic, stretch our imagination living in virtual worlds and having the chance to hitch a ride into the brain of people in very different circumstances.

One of the great challenges for readers is that - unlike pages on the world wide web - you cannot automatically jump from one book to the next in a logical way. Sure, you can read any book that online bookstores recommend, their algorithms will in fact always recommend something - but would that give you the best and richest results? And will you not end up reading the same block busters as everybody else? Each reader has their own preferred ways to discover new books, beyond mere serendipity. One obvious way is through other readers with whom one shares the love for certain less popular books: maybe a book that was misunderstood by the general book buying audience, following a harsh rejection by critics because its unique voice was not recognised when published, will become your favorite book ever. Or maybe it already is, and you would love to talk about it some more.

Inventaire is a project that helps people share books they love, and learn about books they were not aware of. Because Inventair will adopt the W3C ActivityPub standard, people can not only share good reads, but also connect with others to discuss - similar to what book clubs do, but less hampered by time and space.

Technical description

The Inventaire Project is an effort to move forward on the front of accessing information on resources using libre software powered by open knowledge. This ideal is being materialized in the form of inventaire.io, a libre book sharing webapp, inviting everyone to make the inventory of their physical books, declare what they want to do with it (giving, sharing, selling), as well as who should be able to see it (shared publicly through e.g. ActivityPub, or only visible by your friends and groups).

To power those inventories with structured bibliographic data, inventaire.io is also playing the role of a Wikidata-federated open and contributive bibliographic database, extending wikidata.org data with Wikidata-compatible entities (CC0, shared data schema) tailored to our needs, but ready to be pushed to Wikidata when the data contributor deems it appropriate. This linked open data architecture allows users to build their inventories on a huge open knowledge graph, that we believe will, in time, offer exceptional discovery capabilities. Now that this first base of inventories and contributive bibliographic data reaches a certain level of maturity, we want to start moving forward on the next challenges: introduce curation and recommendation mechanisms, improve search tools, offer finer privacy settings, and move forward on decentralization.

Association Inventaire — **Visit https://NLnet.nl/project/Inventaire**

NGI0 Discovery OpenData WebApp

Privacy Preserving Disease Tracking

NGI ZERO

The saying goes <q>desperate times call for desperate measures</q>, but when you really think about it that is not really the case. It makes much more sense to keep ones head cool, and start

taking serious coordinated action with a longer term perspective. Both the SARS-CoV2 pandemic (aka COVID-19 or the Corona virus) and the measures to slow down the spread of the virus have a major impact on society. Unfortunately, a significant number of people has already lost their lives, and the healthcare sector is in parts of the world overheating.

In fighting a disease like this, oversight is everything. The most drastic of measures - like an area lockdown - are extremely expensive and invasive. And not a lot is known about the actual propagation of the virus in the real world. When is it safe to let people shop? Or go to school?

As a citizen, you might be on the one hand inclined to help out - as the virus can pop up anywhere next. These days there is quite some technology that could be put to good use: the smartphones we carry around are amazingly capable devices, and they pack many features such as sensors and antennas. By levering those, we can gather many valuable insights.

Helping to gather this kind of data is probably something good for yourself, others and society at large. In Asia, where the current pandemic started, there have been good experiences with mobile apps that let citizens create a collective measuring system. But before we rush into installing these apps: that data can also be quite sensitive in terms of privacy, and in some parts of the world you might have to fear more for your work as a journalist, whistleblower or activist than for this virus. And of course, cybercriminals as well as state actors currently have a perfect pretext for manipulating people into doing things they will very much regret later - whether using the "desperate times" mantra or not. Fear is not the best counsel, and no doubt some of these malicious actors will have success.

Again, lets keep our head cool and lets get technology in place to help move things forward while at the same time keeping us out of the clutches. The PPDT project set out to design a privacy preserving contact tracing mechanism for mobile apps for disease tracking. This would allow to notify users if they have come in contact with potentially infected people, but would not leak other data such as who was where and met whom. It was meant for citizens first, while science and policy stand to benefit from the additional adoption that a carefully vetted, fully transparent and thus trustworthy open source solution brings.

After it became clear that there were a number of contact tracing apps in development the project steered towards consolidation of its efforts with others in the Temporary Contact Numbers Coalition (https://github.com/TCNCoalition/TCN).

Want to help?

If you can contribute something to this project, please contact us.

Technical description

In case of a pandemic, it makes sense to share data to track the spread of a virus like SARS-CoV2. However, that very same data when gathered in a crude way is potentially very invasive to privacy - and in politically less reliable environments can be used to map out the social graph of individuals and severely threaten civil rights, free press. Unless the whole process is transparent, people might not be easily convinced to collaborate.

The PPDT project is trying to build a privacy preserving contact tracing mechanism that allows to notify users if they have come in contact with potentially infected people. This should happen in a way that is as privacy preserving as possible. We want to have the following properties: the users should be able to learn if they got in touch with infected parties, ideally only that - unless they opt in to share more information. The organisations operating servers should not learn anything besides who is infected,

ideally not even that. The project builds a portable library that can be used across different mobile platforms, and a server component to aggregate data and send this back to the participants.

Visit https://NLnet.nl/project/ppdt

NGI0 Discovery **ContactTracing** **Covid** **SARS-CoV-2**

---

eduVPN on Apple

Technical description

eduVPN is a program under the Commons Conservancy, a non-for-profit foundation focusing on free and open source projects. The goal of the project is to provide a comprehensive and reliable, open source VPN solution for all platforms. This project aims to improve the security and usability of the macOS- and iOS-apps.

Visit https://NLnet.nl/project/eduVPN-apple

VPN Fund

AI-VPN

When you go on the internet in a public place, or on a network you cannot trust, you can use so called 'virtual private networks' to teleport your internet traffic to somewhere else before it goes out on the internet. The term 'virtual' is used because your traffic of course still goes on the same physical network the same way as it did before. The term 'private' signals that you send the traffic to (or at least intend to) somewhere you yourself chose as a trusted intermediate spot on the net. That could be a private home router you control yourself, or an external service like a VPN provider. Using a VPN is in many scenario's a sane approach, that is if (and that is a big if) you can trust the VPN provider itself - otherwise you might actually be worse off. So choose your VPN provider carefully.

For some security risks, for instance when your computer is already infected with malware, using a

regular VPN will not do much. The harmful traffic will just be picked up like the rest of your internet traffic, and delivered to its destination via a slightly longer out. But what if you could use the occasion that all your traffic is sent through a tunnel via a point you control to inspect the traffic that goes into that tunnel, to see if there are any known malicious patterns?

That is what the civisphere project intends to achieve: it wants to help you understand what kind of traffic comes from and goes to your device. It will point out suspicious patterns, reveal any privacy-sensitive data it sees leak out and let you know when it spots a security issue. Obviously, such a solution cannot be a panacea to every risk you as a user are exposed to, but it provides a useful building block to look at the security of your device from the 'outside' - and applying machine learning to spot ongoing issues.

Technical description

Our security decreases significantly especially when we are outside our offices. Current VPNs encrypt our traffic, but they do not protect our devices from attacks or detect if there is an infection. The AI-VPN project proposes a new solution joining the VPN setup with a local AI-based IPS. The AI-VPN implements a state-of-the-art machine learning based Intrusion Prevention System in the VPN, generating alerts and blocking malicious connections automatically. The user is given a summary of the traffic of the device, showing dectected malicious patterns, privacy leaked data and security alerts, in order to protect and educate the users about their security status and any risks they are exposed to.

Czech Technical University — **Visit** **https://NLnet.nl/project/AI-VPN**

NGI0 PET **LocalAnalysis** **MachineLearning** **Security** **VPN**

R I S C - V P h o n e

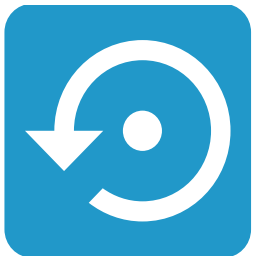
Technical description

The goal of the "RISC-V Phone" project is to develop a simple, fully featured and privacy enhanced mobile phone. It is built using off-the-shelf inexpensive components which are easy to assemble even in a home lab. The software for it is small, simple and easy to audit. Basic phone functionality is running on a secure RISC-V microcontroller (FE310 from SiFive) which controls all peripherals: microphone, speaker, display/touch controller, camera. The phone will be using esp32 for WiFi and Bluetooth, along with industry standard mPCIe modem for cellular communication. Graphics/touch panel controller FT813 enables advanced user experience. The phone will provide VOIP/messaging application using packet data protocol similar to CurveCP which features end-to-end encryption and onion routing. There is also a socket for optional ARM SoM which shares display/touch panel with the main board.

Visit **https://NLnet.nl/project/RISC-V-Phone**

NGI0 PET MobileHardware OpenHardware Phone RISC-V

SeedVault



Technical description

SeedVault is an independent open-source data backup and restore application for Android and derived mobile operating systems. By storing Android users' data in a place the user chooses, and by using client-side encryption to protect backed-up data, SeedVault offers users maximum data privacy and resilience with minimal hassle. SeedVault uses Android's Storage Access Framework (SAF) to read and write encrypted app data. This allows it to backup and restore application data on a wide range of platforms and even USB flash drives. The first part of this project is to improve the current implementation and optimize it to work with widely used self-hosted storage solutions like Nextcloud. The second part of this project is to allow SeedVault to also back up data beyond the installed apps and their data, including the user's photos, videos and music as well as their call logs and SMS.

The Calyx Institute — **Visit https://NLnet.nl/project/Seedvault**

NGI0 PET Android Backup DataStorage E2EE MobileApp

Supersizing the Gun

Technical description

ChipWhisperer is an open hardware and software toolchain that has been a mainstay of hardware security research. ChipWhisperer is used in academic curricula and in industrial R&D implementation security research labs for high speed side-channel power analysis and glitching attacks. The objective of this project is to explore design changes to the current ChipWhisperer hardware, so as to allow capturing of longer power analysis traces and to cater to higher clock speeds than currently supported. Here, the intent is to make it easier to perform side-channel-related analysis of public-key algorithms, without the

need to artificially break down the algorithms into multiple components due to platform constraints. Thid allows for more realistic and practically relevant attacks. This project additionally entails the development of fine-grained post-processing tools, which would make further analysis of captured traces of public-key algorithms easier.

Ultimately, the goal is to work towards candidate post-quantum algorithms, which are known to be more resource-hungry. The project funded by NGI Zero would specifically target design changes to considerably increase the sampling rate (towards 200-250 MS/s) and to provide for a streaming mode (initially envisioned to be roughly 15-30 MS/s). It includes both a new hardware design and a significant update to the current open-source software of the ChipWhisperer platform, as well as demonstration of how to successfully use this with practically relevant ECC public-key algorithms.

Visit https://NLnet.nl/project/Chipwhisperer-Supersize

NGI0 PET **Cryptography** **Hardening** **OpenHardware** **PostQuantum** **PowerAnalysis** **Security** **Toolchain**

Pixelfed Live

After you take a picture of your brand new car, your smiling baby or the food you were just served, what do you do? You want to show it to everyone you know of course. But do you really know who you are actually sharing your private snapshots with when you post them online? With high grade cameras in nearly every mobile phone and numerous instant messaging apps and social media platforms available, sharing photos is just as easy (and perhaps more popular) than typing out what you want your friends and family to know about your life.

Social platforms and apps make us feel like we are only sharing our images with our own social circle and maybe some faraway friends we met online. But because many so-called 'free' social sharing tools like Instagram actually monetize your data and online activity to sell you personalized ads, your online picture book may not be so private at all. And where do those snapshots, that sometimes contain very personal information about where you live, what you are doing and who you know, actually end up after you clicked that upload button?

When you want to show someone your holiday pictures, you simply want to share those pictures, instead of also handing over a copy to the postal service to check where you went to and possibly send you a cheap flight deal for the coming holidays. Pixelfed is a platform that makes this possible on the internet. Users can choose to run and host the service themselves or choose someone they trust to store their pictures and private data with. No one will track what photos you share and which people you follow. The pictures your friends and family share pop up in your timeline one after the other, without ads or algorithms that decide what you can and cannot see. The latest much requested feature that will be added is live streaming, making Pixelfed an even more versatile privacy-friendly alternative to Instagram

and the likes.

Technical description

Pixelfed is an open source and decentralised photo sharing platform, in the same vein as services like Instagram. The twist is that you can yourself run the service, or pick a reliable party to run it for you. Who better to trust with your privacy and the privacy of the people that follow you? The magic behind this is the ActivityPub protocol - which means you can comment, follow, like and share from other Pixelfed servers around the world as if you were all on the same website. Timelines are in chronological order, and there is no need to track users or sell their data. The platform has many features including Discover, Hashtags, Geotagging, Photo Albums, Photo Filters and a few still in development like Ephemeral Stories. After supporting development of social discovery and a mobile app, NGI Zero funds this project to add a much requested live streaming feature to Pixelfed.

Visit https://NLnet.nl/project/PixelFedLive

NGI0 Discovery **Abuse** **PhotoSharing** **ProgressiveWebApp** **UX**

Tooling to improve security and trust in GNU Guix

When you start up your computer, you will probably think twice before you download some random piece of software from the internet and run it. You know that doing so could allow unwelcome guests to your computer and your data. Your computer might even end up in a bot net. So when you see some nice piece of software, you will ask yourself the question: can I really trust the software? Perhaps you will check the origin it comes from. Better safe than sorry.

Did you miss checking something, though? What about the software that is already on your computer before you started? A computer is not of much use without an operating system. While most computers are sold with an operating system, actually you have the choice to remove that and install something different. Have you thought about the trustworthiness of that fundamental piece of software - your most fundamental travel companion on the wild west of the internet? Trustworthiness is essential. When an operating system has a so called 'back door' (either intentionally or not), someone could extract whatever user data - like personal pictures or home movies - from your computer. And the worse thing: without you ever finding out. The operating system guards all the other software, and warns you when you install software from the internet. But itself, it doesn't have to ask for permission. Ever. It doesn't just have "access all areas": in fact, it runs the whole show.

With commercial software like Microsoft Windows or Mac OS X that you get delivered when you buy a computer, trust in what their closed operating system does will of course always be a leap of faith: as a

user you essentially are given no choice. In proprietary systems you do not have the freedom to study the source code, or to control what really happens. So you either trust the vendor, or you'd better not use it. For an increasing amount of people, after the revelations from whistleblowers like Edward Snowden, that "leap of faith" is not so obvious anymore. They prefer to use free and open source operating systems like GNU Linux, FreeBSD and OpenBSD. These are technology commons: the people that wrote the software allow you to inspect the source code. Even more so, they give you the source code to do anything with it that you like. So you don't just blindly have to take their word for it and trust them, you can take matters into your own hands.

One step beyond transparent source code is transparent running code. After all, most software is distributed pre-compiled with no method to confirm whether the binary code you have installed on your system is actually identical to the thoroughly vetted source code. GNU Guix is a package manager and operating system that can guarantee such reproducible builds, proving that no vulnerabilities or backdoors were introduced and the software you are using (for potentially vital services or for handling sensitive data) is certifiably trustworthy.

Technical description

GNU Guix is a universal functional package manager and operating system which respects the freedom of computer users. It focuses on boostrappability and reproducibility to give the users strong guarantees on the integrity of the full software stack they are running. It supports atomic upgrades and roll-backs which make for an effectively unbreakable system. This project aims to automate software vulnerability scanning of packaged software to protect users against possibly dangerous code.

Visit https://NLnet.nl/project/GUIX-securitytracking

NGI0 Discovery CVE Deployment EndUser Monitoring Vulnerability

---

R e - i s e a r c h

“Re-isearch” is a project following in the spirit of the original isearch developed back in the 1990s. Like the original, it is not just about textual words but the design contains a large number of objects: numerical, range, geospatial etc. It is unique among full-text systems in that it also provides numerous object types with their own methods of search and allows these to be viewed parallel as text--- a date field (of which it will be one of the first to support some key parts of the new ISO-8601:2019 standard date semantics), for instance, can be searched as a date but also a text, searching for the words in the field.

These objects don't even have to be part of any document but may be available via interface glue into other systems via ODBC, CORBA or object embedding. This allows indexing content--- for example from RSS/XML--- to be stored in and searched from other systems. This is useful in many dynamic applications in commerce and trading (keeping live counts of goods on hand, selling prices, etc.). Objects don’t even have to always be explicitly defined as various doctypes (document handlers) can automatically (if enabled, resp. not disabled) at index time detect a number of field data types(such as that something is a telephone number or a date or..).

A radical departure from other designs is its concept of search granularity. With typical text indexers one has the concept of document or record and that is the unit of index and the unit of retrieval. Instead we can have a dynamic search time unit of retrieval: user specified or heuristically determined. The structure of of documents can be exploited to identify which document elements (such as the appropriate chapter or page) to retrieve. Retrieval granularity may be on the level of sub-structures of a given document or page such as line, paragraph but may also be as part of a larger collection.

Technical description

*Project re-isearch: a novel multimodal search and retrieval engine using mathematical models and algorithms different from the all-too-common inverted index (popularized by Salton in the 1960s). The design allows it to have no limits on the frequency of words, term length, number of fields or complexity of structured data and support even overlap--- where fields or structures cross other's boundaries (common examples are quotes, line/sentences, biblical verse, annotations). Its model enables a completely flexible unit of retrieval and modes of search.

Initial project outcome: a freely available and completely open-source (and multiplatform) C++ library, bindings for other languages (such as Python) and some reference sample code using the library in some of these languages.

Visit https://NLnet.nl/project/Re-iSearch

NGI0 Discovery **Indexing** **SearchEngine** **Semantic** **StructuredData**

Non-Ranking Comparison Platform

Search and discovery is one of the most important and essential use cases of the internet. When you are in school and need to give a presentation, when you are looking for a job, trying to promote your business or finding relevant commercial or public services you need, most of the time you will turn to the internet and more importantly the search bar in your browser to find answers. Searching information and making sure your name, company or idea can be discovered is crucial for users, but they actually have little control over this. Search engines set the terms for what results you see, how your website can be discovered and what information is logged about your searches. What terms are set remains obscure for users and they can only follow the rules laid out for them, instead of deciding on their own what, where and how to find the information they are looking for.

Not only do most online search solutions set the terms for which results you see, how they are ranked and ordered on the screen is also taken out of your hands. A website buried on the third page might just have the information you were looking for, but is unfortunately not compliant with whatever requirements the site you are using has set for them. Searching, ranking and rating is essentially done for you, leaving you to guess what decisions were made in the process (and who might benefit from being ranked higher than others). This lack of independent ranking and rating functionality becomes even more critical when you think about using such information to make possibly life-altering decisions, such as deciding where to live, to work or to study.

This project aims to develop a non-ranking comparison platform as an alternative to ranking and rating

sites where an unknown third party decides which information ends up where. Instead this will be a collaborative effort, where users can collectively filter and compare items on their own terms, adding data that might be missing and suggest attributes that may make an object more informative and searchable, all the while keeping total ownership over the data they themselves own. This way a community of peers can create a knowledge base for users to base their decisions on in an independent, trustworthy way.

Technical description

Ranking items according to fixed criteria is a common method used to support people in making decisions between multiple options. However, mismatches between the criteria used to generate rankings and the target audience's requirements may lead to uninformed and sub-optimal decisions. This non-ranking comparison platform aims to bridge this gap by providing users with the possibility to compare options based on their custom criteria, and make informed decisions. The goal is to create an open-source general purpose web application allowing the creation of custom comparison platforms.

Zusammenkunft aller Physik Fachschaften e.V. — **Visit** **https://NLnet.nl/project/ComparisonPlatform**

NGI0 Discovery **Comparison** **StructuredData**

LibreOffice P2P

Collaboratively writing a document together in real-time with others is still a bit magic. Someone else, perhaps on the other side of the planet, is typing something. And within a fraction of a second, the text magically appears on your screen. This amazing technology is the ideal companion for say an online meeting - everyone can contribute, and correct any flawed minutes without much effort. For this kind of collaboration in real-time, there is a limited set of options in the market you can use. Most available services in the market like Google Docs, Microsoft Office or LibreOffice Online.

Most online collaborative services share one very undesirable characteristic: you need to fully trust the company running the service you use. Whomever has access to the servers used to connect everyone together, can read everything you have written - and deleted. That means that if you need to work on something confidential like an important contract, you may want to reconsider using the service. Especially if you write about sensitive topics like corruption, money laundering or state surveillance this open backend you cannot control is a really significant problem.

Peer-to-peer collaboration is a way for internet users to connect and work together directly, without the need for a central authority or in-between layer. Search and discovery in this way can be crowd-sourced, instead of organized by one central party (a search engine) that is more vulnerable to attack and misuse. Together, peers can publish data, subscribe to other people's messages and documents, recommend and disseminate information and news and tag correct and informed articles and stories, that can then be searched by others. The group filters what data and information should be spread wide and far and what should be forgotten, not a third party (i.e. the search engine provider) that will not give access to its

search algorithm to protect their commercial interests.

This project wants to add peer-to-peer functionality to LibreOffice Online, allowing users to edit documents together while making sure that the office software runs completely on their own device and that all their work is end-to-end encrypted. This way an office suite that uses open source software and open standards only is also protected against single points of failure (like a server that goes down or is compromised) and can guarantee proper encryption of potentially sensitive documents.

Technical description

LibreOffice Online is the online version of the popular open source office application, and a leading implementation of the ISO/IEC 26300 OpenDocument Format standard. During the project this free software application will be modified so it can run fully client-side inside a regular browser - meaning you can view and edit office documents without an install required. This provides the technical foundations to support true P2P editing of complex office documents. The ability to remove the entire dependency on a server means that document collaboration is moving towards zero-knowledge implementations – where no single-point of architectural failure exists and no data is required to sit unencrypted on a non-user owned (or trusted) server instance. The improved LibreOffice Online will be able to provide end-to-end encryption – both for the peer2peer use case, as well as securely keeping documents encrypted when at rest. That means data is safe when the user is disconnected, whether it is stored on an untrusted server or in the local Web storage.

Visit https://NLnet.nl/project/LibreOfficeP2P

NGI0 Discovery **E2EE** **P2P** **RealTimeCollaboration** **ZeroKnowledge**

How AdTech works

Technical description

The web has become a place where visiting a webpage triggers many effects elsewhere on the globe, and where advertising technology has morphed into a market driven surveillance ecosystem of a size that was unimaginable even a few decades ago. While especially older people may still think of the 'friendly' world wide web of the nineties, the reality is that underneath the surface of many web pages lies a dark technology layer that sprawls data. "How AdTech works" is a project by the European umbrella of digital rights organisations, EDRi. The goal of EDRi is to address the threat these developments hold for our online lives and the shared public spaces. EDRi wants to de-mystify and challenge the complex and

secretive world of online advertising and profiling - and bring attention to these issues at a policy level.

With upcoming platform regulation like the pending EU Digital Services Act (DSA), there is an urgent need to share insights among human rights defenders, academics and the public at large. We need a concerted effort to take on this challenging subject - in order to better understand and subsequently challenge invasive and exploitative monopolistic practices that lead to aggravations of polarisation, spread of disinformation, and other abuses of fundamental rights.

EDRi will engage with legislative efforts across Europe as an opportunity to better protect people's rights online against data-hungry, abusive business models.

EDRi will support this work via creation of a publication on AdTech and online advertising booklet, which will be distributed among policy makers, human rights defenders and the broader public.

European Digital Rights — **Visit https://NLnet.nl/project/EDRi-AdTech**

NREN **Advertising** **Education** **Interoperability** **RealTimeBidding**

Genodepkgs

NGI ZERO

When you start up your computer, you will probably think twice before you download some random piece of software from the internet and run it. You know that doing so could allow unwelcome guests to your computer and your data. Your computer might even end up in a bot net. So when you see some nice piece of software, you will ask yourself the question: can I really trust the software? Perhaps you will check the origin it comes from. Better safe than sorry.

Did you miss checking something, though? What about the software that is already on your computer before you started? A computer is not much use without an operating system. While most computers are sold with an operating system, actually you have the choice to remove that and install something different. Have you thought about the trustworthiness of that fundamental piece of software - your most fundamental travel companion on the wild west of the internet? Trustworthiness is essential. When an operating system has a so called 'back door' (either intentionally or not), someone could extract whatever user data - like personal pictures or home movies - from your computer. And the worse thing: without you ever finding out. The operating system guards all the other software, and warns you when you install software from the internet. But itself, it doesn't have to ask for permission. Ever. It doesn't just have "access all areas": in fact, it runs the whole show.

With commercial software like Microsoft Windows or Mac OS X that you get delivered when you buy a computer, trust in what their closed operating system does will of course always be a leap of faith: as a user you essentially are given no choice. In proprietary systems you do not have the freedom to study the source code, or to control what really happens. So you either trust the vendor, or you'd better not use it. For an increasing amount of people, after the revelations from whistleblowers like Edward Snowden, that "leap of faith" is not so obvious anymore. They prefer to use free and open source operating systems like GNU Linux, FreeBSD and OpenBSD. These are technology commons: the people that wrote the software allow you to inspect the source code. Even more so, they give you the source code to do anything with it that you like. So you don't just blindly have to take their word for it and trust them, you

can take matters into your own hands.

Transparent, auditable operating systems are even more important for security-critical setups that handle sensitive data, provide vital services or operate in harsh environments. In these cases users want to pick and choose exactly (and only) the components they need to create a system they can easily verify. That is what this project will contribute to: a trusted computing base users can build on by picking from a comprehensive collection of available and auditable software packages to create a composable and flexible setup that does exactly (and only) what you want it to.

Technical description

The past decade has seen substantial improvements in the field of operating systems that have raised the standards for building high-assurance and security-critical systems. Unfortunately this technology is rarely utilized by smaller organizations and private users due to the cost of retooling, reconfiguring, and the lack of continuity between OS communities.

The Genode OS framework is a free-software toolkit of components that can be used to construct custom operating systems from a trusted codebase of drastically reduced complexity. Genodepkgs is an extension to the Nix package collection that integrates the Genode toolkit. This package collection, or Nixpkgs, is one of the most comprehensive collections of readily deployable software to date, and contains within it the NixOS Linux distribution. By extending the collection to cover Genode, a new diversity of operating systems can be realized using the variety of microkernels, device drivers, and utilities provided by Genode, as well as hybrid systems composed of an isolating Genode base layer and virtualized NixOS guests. Making such compositions possible by reusing the methods of NixOS can bridge the divide between contemporary Linux system administration and next-generation operating system developments.

Visit https://NLnet.nl/project/Genodepkgs

NGI0 PET CrossCompilation Delivery Hosting Hypervisor Microkernel

---

K a i d a n

Kaidan

One of the things people enjoy the most about the internet, is that it enables them to talk to others remotely almost without limit. Internet allows anyone connected to keep closely connected with friends and family, and help their kids solve a math problem while they are at work. People collaborate with their colleagues from the couch of their living room, the cafe where they enjoy lunch or on their cell phone on the bus to the gym. Businesses can easily service their customers where this is most convenient to them, without having to travel themselves. This is so convenient, that some businesses have already moved entirely online. Internet communication has become the nerve center of whole neighbourhoods, where people watch over the possessions of their neighbours while these are away for work or leisure.

However, users have a hard time to understand how privacy is impacted if they use the wrong technology. Because internet works almost everywhere, the natural privacy protection of the walls of a house, a school or an office is gone. Unlike the traditional phone companies, many of the large technology providers run their business not on delivering an honest service but on secretly eavesdropping on their users and selling information to others. It is mostly not about what you say, so it is relatively easy for providers to allow some form of privacy by encrypting messages. The more interesting parts are who talks to whom, when, and where they are in the real world while they meet on the internet. if you want to be reachable across the internet, you have to constantly let the communication provider follow you wherever you go. This makes the private and professional lives of citizens an open book to companies that with the help of AI and other technologies make billions from selling 'hidden data' normal people are completely unaware of even exists. And of course in societies that are not so democratic, this type of information is critical to bring down opposition and stifle human rights.

Users assume the confidentiality and privacy when they communicate, and they are morally justified to do so. There is nothing natural or final about internet communication providers having access to all this very personal information - or going down the dark path of selling data about customers. The cost of this in terms of internet usage and computer power needed is actually negligible, and so all it takes it the availability of open alternatives that people can use.

Kaidan is a project to develop a user-friendly client for XMPP, an open, free and decentralized instant messaging network. There are public XMPP servers all around the world users can choose from or if they have the devices and time, host one on their own for their colleagues, friends or family. Kaidan aims to provide a single app for all major operating systems to connect to XMPP servers, allowing users to switch from device to device while using the same familiar interface. This project will add user-friendly and accessible end-to-end encryption to the app, further guaranteeing the privacy of instant messaging based on open standards and open source software.

Technical description

Kaidan is a user-friendly and modern chat app for every device. It uses the open communication protocol XMPP (Jabber). Unlike other chat apps, you are not dependent on one specific service provider. Instead, you can choose between various servers and clients. Kaidan is one of those XMPP clients. In contrast to many other XMPP clients, it is easy to get started and switch devices with Kaidan. Additionally, it adapts to your operating system and device's dimensions. It runs on mobile and desktop systems including Linux, Windows, macOS, Android, Plasma Mobile and Ubuntu Touch. The user interface makes use of Kirigami and QtQuick. The back- end of Kaidan is entirely written in C++ using Qt and the Qt-based XMPP library QXmpp.

Visit https://NLnet.nl/project/Kaidan

NGI0 PET InstantMessaging XEP XMPP

Openwifi

When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component, the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

Thankfully, alternatives are underway, as communities of engineers and developers are hard at work to create hardware that is transparently designed from top to bottom, creating components for fully auditable, verifiable devices. This is not an easy task however, as modern hardware consists of countless parts and components each with their own functionalities and possible vulnerabilities. This project aims to create an open hardware alternative to a very critical piece of most smartphones, tablets and computers: the radio chip that for example handles Wi-Fi, Bluetooth and cellular connectivity. Further development will push this open source radio chip design to match current-day wireless networking standards, ultimately providing a crucial component of truly trustworthy open hardware devices.

Technical description

The Openwifi project aims to offer an open source Wi-Fi chip design that could act as a missing piece of the open source software/hardware puzzle. In the past decades, open source software has played a key role towards the open and trusted internet. In recent years, the open source processor project, like openRISC and RISC-V, pushes forward to construct open source devices/computers. However, the radio connectivity of the device still relies on the black-box radio chips (Wi-Fi, BLE, cellular). As the initial step of the open source Wi-Fi chip, openwifi project has implemented the 802.11a/g full-stack on the FPGA based Software Defined Radio (SDR) platform. The FPGA (Xilinx Zynq SoC) also includes a multi-core ARM processor, so that we can have Linux (TCP/IP, mac80211 and driver) and Wi-Fi (Low MAC and PHY) in the same chip. This NGI funding opportunity will support openwifi project development of 802.11n feature, which moves the project closer to the state of art Wi-Fi technology. The development mainly includes 3 tasks: Adding the 802.11n mode to the original 802.11a/g PHY (Physical layer) transceiver; Extending the low MAC (Media Access Control) and processor interface to support the additional 802.11n elements, such as the SIGNAL field and bigger payload size; Improving the openwifi driver to handle the 802.11n elements and expose the 802.11n capabilities to Linux mac80211 framework. The Openwifi project currently focuses on the Wi-Fi functionality, integrity and stability. In the future, the platform independent methodology will be considered: Integrating the openwifi IP with open source on-chip bus

(such as wishbone) and RISC-V processor by open source EDA tools.

IDLab, Gent university - imec — **Visit https://NLnet.nl/project/OpenWifi-80211n**

NGI0 PET **IEEE** **OpenHardware** **Wifi**

R E T E T R A

Cryptography is everywhere in modern communication: when you pick up your mobile phone to answer a call, enter a site URL in your browser bar or send a chat message, there is a complex series of mathematical operations happening behind the scenes to guarantee that no one can spy on your conversation, that the site you visit is legitimate and that your messages can only be seen by the friends you sent it to. These cryptographic solutions need to be secure for communication to be trustworthy or even function in general. This becomes even more important when considering emergency services and governmental telecommunication channels: a faulty or leaking connection could potentially cost lives.

To make sure that the cryptographic algorithms at the core of emergency communication channels work as intended, they should be open to verification and auditing. In case of the European TETRA-standard, unfortunately, this is not possible due to proprietary cryptographic suites that are sealed off to the public. Reverse engineering has shown that the cryptographic algorithms, also known as cipers, are flawed. As TETRA is widely used by governmental agencies, emergency services and critical infrastructure like remote control of oil rigs, transportation and electric and water utilities, these vulnerabilities should be addressed. This project will reverse-engineer the proprietary cipher suites and instead provide a secure, transparent open source alternative, so this critical infrastructure can rely on trustworthy technology that anyone can inspect and audit to guarantee safe communications.

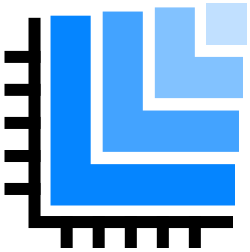
Technical description

Terrestrial Trunked Radio (TETRA) is a European standard for trunked radio used globally by government agencies, emergency services and critical infrastructure. Apart from most European police agencies (such as BOSNET in Germany or RAKEL in Sweden), military operators and emergency services, TETRA is also widely used for SCADA telecontrol of oil rigs, pipelines, transportation and electric and water utilities. TETRA authentication and encryption are handled by secret, proprietary cryptographic cipher-suites known as TAA1 and TEA which are only available to select parties under strict NDAs which runs counter to both the spirit of open technologies and Kerckhoffs's principle. The latter's potential consequences are illustrated by the fate of A5/1, A5/2 and their GMR variants in cellular and satellite communications, allowing ciphers that can be broken in practice to fester in public and critical infrastructure for far too long. This project aims to reverse-engineer and subsequently perform cryptanalysis on these cipher-suites and finally formulate a hardening roadmap in order to provide a research-oriented FOSS implementation of the cipher-suites and aid affected parties in moving away from unexamined, proprietary security mechanisms towards open standards.

Visit https://NLnet.nl/project/TETRA-crypto

NGI0 PET Cryptanalysis Encryption Radio ReverseEngineering TETRA

L i b r E D A



Consumers and businesses overpay for computer hardware, because the market is not working well. When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component, the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

To make open hardware design accessible, this project will simplify the development of chip layout tools. These tools are needed to turn a description of how components in a circuit will work, to an computer chip layout to be produced. This way the project can both help to educate students and hobbyists about chip design as well as provide small enterprises and organizations with publicly available tooling to create their own open hardware.

Technical description

Because digital circuits are a core part of today's society there is a significant value in free and open chips and, equally important, free and open design software that is accessible also to small entities. Not only would this enhance trust through transparency and digital sovereignty through distributed knowledge but it would also be a fertile ground for education, hobbyists and small enterprises. The main goal of this project is to create a new libre-software framework for the physical design of digital integrated circuits. The framework is meant to simplify the development of chip layout tools, i.e. the tools used to convert a gate-level netlist into a fabrication-ready layout. This includes fundamental data structures and algorithms, interface definitions of the design algorithms (e.g. placement, routing or timing analysis), input/output libraries for commonly used file formats as well as documentation and example

implementations. Two variants will be pursued in parallel: One with a clear focus on simplicity and education and another with a focus on performance and scalability. Another part of the project is the continuation of the 'LibreCell' standard-cell generator and characterization tool.

Visit https://NLnet.nl/project/ChipDesignFramework

NGI0 PET **OpenHardware** **StandardCell**

M N T R e f o r m

When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component, the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

To break through this standstill, developer communities are working hard to deliver open, trustworthy and accessible alternative computer hardware that anyone can use, study, modify and distribute, just like they can with open source software. This project leverages existing open source software and open hardware to create a modular, repairable and entirely transparent laptop, from the software it runs down to the silicon.

Technical description

MNT Reform is a modular open hardware laptop, the first of its kind - designed and built in Europe. The project has high ambitions in terms of usability and user experience. A mechanical keyboard and an elaborate industrial design provide for professional ergonomics. MNT Reform uses RISC processors like ARM and has no built-in recording technology. It runs a free and open source software stack from the ground up. Third parties can easily contribute to the development of new modules. The modular approach does not only make the laptop more extensible but also improves sustainability, and supports the right to repair.

During the project, the team will develop two open hardware System-on-Modules. The first module is based on NXP LS1028A, and will increase RAM capacity to up to 16GB and make external GPUs usable.

The second open hardware SoM uses an FPGA (field programmable gate array) to support the validation of open silicon SoC projects in a real laptop. Modules like this make the development of embedded computers easier for open hardware engineers by pre-solving risky and expensive challenges. Finally, we will develop an optional camera module for MNT Reform as part of the project, which will allow the laptop to be used for remote learning and video conferencing.

MNT Research GmbH — **Visit https://NLnet.nl/project/MNT-Reform**

NGI0 PET **Laptop** **OpenHardware** **SystemOnModule**

Balthazar - One laptop for the new internet age.

When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component, the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

To break through this standstill, developer communities are working hard to deliver open, trustworthy and accessible alternative computer hardware that anyone can use, study, modify and distribute, just like they can with open source software. This project aims to combine open source software and open hardware into a sustainable, affordable and trustworthy laptop, to make privacy-friendly and transparent computing technology available and accessible to everyone.

Technical description

Project's ambition is to design and deliver an innovative and technically advanced open hardware (RISC-V/ISA) based, European made, inexpensive, FOSS laptop as a personal computing device, containing on board all desirable (FOSS compliant) hardware and software features and functionalities needed to prevent any 3rd party intrusion into the system. It adds physical safety features currently not available in the market such as hot-swappable CPU, hardwired switches for e.g. camera and audio devices, and a quickly removable encrypted hard drive and peripherals. A goal of Balthazar is to enable and educate end users to be private, safe and careful with their own data, and that of others. Another goal is to make

computing more sustainable and reach eco-friendly footprint, by empowering users to take up their 'right to repair', through a modular laptop that allows components to be easily exchanged and upgraded - up to the CPU itself. The goal is to lead by example and gently lead other hardware manufacturers to become fully open and transparent. And create an educational platform, as well as an advanced computing device where its users (including those with low income) to feel secure, safe and comfortable using it. For the children of all ages.

Balthazar E.V. — **Visit https://NLnet.nl/project/Balthazar-Prototype**

NGI0 PET **Laptop** **OpenHardware**

Distributed Trust for Web Servers

One of the oldest questions on the internet is: how do you adequately prove you are you? Or perhaps the reverse formulation offers a better mental model: how do you prevent others from succeeding in pretending they are you? Now lets flip this question around once more: how would you like to see this managed yourself, if you could? How heavy-weight or convenient do you want to be proven that you are you, to allow you to get into your own environment or have something done on your behalf? And what is it worth to you in terms of effort? Would you be willing to spend a minute to have some clever secure device you have in your pocket involved? Authenticate via your mobile phone? And what if you are in a rush, or on the go? Are you happy with some company like your email provider or a large social network having the ability to make that judgement, based on a user login a few hours ago? And what if that company is based in some other jurisdiction, and could be forced to let others in as well? Or would you rather choose your own identity, and formulate direct rules to have complete control at any given point?

As could be guessed, individual people have a need for different levels of confidence and security in different contexts. A security breach matters perhaps less if you just want to login to a music service to change a playlist. After all, the worst that can happen is that someone messes things up and you have to create a new one. It matters a great deal more if you want to do a significant financial transaction at work, or open the door of your house remotely to let the babysitter in while you are delayed in traffic. Perhaps you can think of scenarios where you want even more control.

So what proof to use as the basis of your trust, and the subsequent actions taken? Historically people rely on some authority they collectively trust. Such an authority has typically taken high tech countermeasures to make the channel through which that trust is conveyed hard to fraud. A passport or banknote are quite tricky to fabricate due to the use of special techniques. Online we have only a very limited amount of trust "anchors" of varying quality. The domain name system is such an anchor, digital certificates or customer relationships are another. Any central authority of course introduces a potential single point of failure. If a certificate authority or the digital proofs they provide for your online identity and were to be compromised, you can imagine the worldwide damage that can be done and data that could be stolen.

Instead of relying on single points of failure, this project wants to distribute this trust across two or more independent certificate authorities so no single attack can be successful. This can help provide more resilient security to the entire internet infrastructure and ward off the increasing threat of cyber criminality.

Technical description

The M-Pin protocol, and its implementation in the Milagro project currently incubating at Apache, provides cryptographic security using a distributed trust model. In place of the single point of failure (and high-value target for social engineering attacks) of today's Certificate Authorities (CAs), cryptographic verification is assembled from two or more mutually independent authorities, all of which would need to be subverted at once to break security. This project helps bring distributed trust to the Web, by implementing M-Pin support via Milagro's libraries in leading Open Source web servers. This will pave the way both to a distributed trust alternative to monolithic CAs and browser trust lists, and to a distributed trust alternative to protocols such as OpenID for user identification.

Visit https://NLnet.nl/project/M-PIN

NGI0 PET **DistributedTrustAuthority** **IdentityManagement** **Proxy** **Server**

End-To-End Encryption for Jitsi Meet

One of the things people enjoy the most about the internet, is that it enables them to talk to others remotely almost without limit. Internet allows anyone to keep closely connected with friends and family, and help their kids solve a math problem while they are at work. People collaborate with their colleagues from the couch of their living room, the cafe where they enjoy lunch or on their cell phone on the bus to the gym. Businesses can easily service their customers where this is most convenient to them, without having to travel themselves. This is so convenient, that some businesses have already moved entirely online. Internet communication has become the nerve center of whole neighbourhoods, where people watch over the possessions of their neighbours while these are away for work or leisure.

However, users have a hard time to understand how privacy is impacted if they use the wrong technology. Because internet works almost everywhere, the natural privacy protection of the walls of a house, a school or an office is gone. Unlike the traditional phone companies, many of the large technology providers run their business not on delivering an honest service but on secretly eavesdropping on their users and selling information to others. It is mostly not about what you say, so it is relatively easy for providers to allow some form of privacy by encrypting messages. The more interesting parts are who talks to whom, when, and where they are in the real world while they meet on the internet. if you want to be reachable across the internet, you have to constantly let the communication provider follow you wherever you go. This makes the private and professional lives of citizens an open book to companies that with the help of AI and other technologies make billions from selling 'hidden data' normal people are completely unaware of even exists. And of course in societies that are not so democratic, this type of information is critical to bring down opposition and stifle human rights.

Users assume the confidentiality and privacy when they communicate, and they are morally justified to do so. There is nothing natural or final about internet communication providers having access to all this very personal information - or going down the dark path of selling data about customers. The cost of this in terms of internet usage and computer power needed is actually negligible, and so all it takes it the availability of open alternatives that people can use. Jitsi Meet is a popular alternative to proprietary videoconferencing solutions. Users can host Jitsi Meet on their own system or use a host they can trust. This project will provide additional security and privacy by completely encrypting your video conferences from end to end, leaving no chance for possible data leaks or spying third parties between users.

Technical description

Jitsi Meet is an open-source video conferencing application that uses Jitsi Videobridge to provide high quality, secure and scalable video conferences. Traditionally, it used hop-by-hop encryption to secure the contents. The drawback of this is of course that the videobridge is able to view the unencrypted contents. With the advent of the WebRTC Insertable Streams API in Chrome it became possible to implement actual end-to-end encryption on top of WebRTC. This project will implement and verify a more complete solution that involve a key management system which establishes public keys, derives encryption keys and changes them depending on the state of the conference.

Visit https://NLnet.nl/project/JitsiMeet-E2E

NGI0 PET E2EE Videoconferencing WebRTC

PrivateRecSys

Search and discovery is one of the most important and essential use cases of the internet. When you are in school and need to give a presentation, when you are looking for a job, trying to promote your business or finding relevant commercial or public services you need, most of the time you will turn to the internet and more importantly the search bar in your browser to find answers. Searching information and making sure your name, company or idea can be discovered is crucial for users, but they actually have little control over this. Search engines set the terms for what results you see, how your website can be discovered and what information is logged about your searches. What terms are set remains obscure for users and they can only follow the rules laid out for them, instead of deciding on their own what, where and how to find the information they are looking for.

Online search basically is a black box: you enter your question and get an answer, or optimize your site to to end up in the top ten results, but no one has actual control over how it all works. Not only does this make us dependent on search providers, it can (and does) jeopardize your privacy, from the actual query itself to all sorts of sensitive metadata you might leak (other sites you visited, your IP address, other online accounts, etcetera).

So how do we regain control over how we search online? One way to do this is to build transparent, user-centric and privacy-friendly alternatives to popular search solutions. That is what this project aims to do for recommender systems like you find on the bottom part of most webshops (think of 'customers have

also bought'). These systems are very lucrative for online businesses, but usually take in a lot of personal data to provide accurate recommendations. This project will show that personalized search and discovery does not have to come at the cost of your privacy by making an open source toolkit for privacy-preserving recommender systems. This way websites and web shops do not have to choose privacy over functionality, but instead combine the two into a more user-friendly online space for everyone.

Technical description

The use of recommender systems has grown significantly in recent years, with users receiving personalised recommendations ranging from products to buy, news to read, movies to watch, people to follow. At the same time, recommender systems have become extremely effective revenue drivers for online business. However, producing personalised recommendations requires collecting of users' data, which makes conventional recommenders effective at the cost of users' privacy. The PrivacyRecSys project aims to develop an open-source toolkit for delivering accurate recommendations while respecting users' privacy. The toolkit will consist of novel privacy-preserving recommender approaches, which modify the state-of-the-art recommender approaches by applying the principles of differential privacy, homomorphic encryption and federated learning.

Visit https://NLnet.nl/project/PrivateRecSys

NGI0 Discovery **Recommending**

N o m i n a t i m

Everyone needs to find their way around the world, be it traveling for work, taking a vacation or going to the doctor, dentist, your local municipality and other important (public) services. How we move around and where we go is very personal information: imagine following someone for a week and what this can teach you about their life, their loved ones and what is important to them. Now think about the apps or devices you use for navigation and what they can and probably do log about you. Where does this information go, who has access to it, how does this feed into your data profile that is created and sold by tech platforms to businesses (and sometimes governments)?

Navigation shouldn't be yet another underhanded means for tracking and profiling, it should help you get to where you need to be and inform you about your travel, nothing else. OpenStreetMap is a collective effort to build a tool that brings geographic data and navigation into the public space, as an alternative to commercial services. Users help map areas, roads, buildings and other points of interest and keep this information up to date and enrich it. All data is open and free to use.

A map of the world of course is not enough: you will also need something to help you find your way

around. An important starting point is geographic search, like the search engine on the main website of OpenStreetMap that lets you find places and navigate routes. The technology behind this search engine and behind countless other geographical search tools can be improved to take different languages and addressing practices into account. This would open up the geographical treasure trove of OpenStreetMap to users and developers everywhere, providing them with the tools and data necessary to build more privacy-friendly and independent geographical technology.

Technical description

Nominatim is an open-source geographic search engine (geocoder). It makes use of the data from OpenStreetMap to built up a database and API that allows to search for any place on earth and lookup addresses for any given geographic location. It is used as the main search engine on the OpenStreetMap website where it serves millions of requests per day but it can also be installed locally. You can easily set it up for a small country on your laptop. Nominatim has always aimed to be usable world-wide for any place in any language. To that end it has used generic, language-agnostic algorithms that assume a uniform data model. This has served us especially well while the OpenStreetMap database was in its early stages of development and changing fast. Now that it has matured, it is time to further improve the search experience by taking into account the particularities of different languages and the different practises when it comes to geographic addressing. We aim to restructure the part of the software that parses the place names and search queries to make it more configurable and make it easier to take into account languages and regional peculiarities.

Visit https://NLnet.nl/project/Nominatim

NGI0 Discovery **Geocoding** **OpenStreetmap** **SearchEngine**

GNU social

A lot of the people we talk to, the media we watch and the services we search for are found in or through using social media. For users these platforms offer easy and usually free services to send public and private messages, stay updated on relevant news and promote your business or product.

But the services these social media offer do actually come at a personal and societal cost. The platforms are not neutral exchange platforms like the rest of the internet. They do not just deal with all messages they receive in the same way. Part of the corporate social network model is to give some messages preferential treatment over others, i.e. there is a noticeable bias towards those that pay. People only have so much attention they can spare every day, and the companies decide what you cannot skip based on what they get paid. This would be equivalent to you always seeing the newsletter from Coca Cola at the top of your email client, but only half of the emails from your father or local charity because they are automatically put in a folder out of sight. This "pay to play" creates a knockout race for attention fueled by commerce, not by arguments, emotions, ethics or societal considerations.

This exposure is worsened by the fact that the platforms monetize your data and behaviour. Social media

companies create fine-grained personal profiles, that even include attributed political, relational and other deeply personal matters. By clustering people, profiles becomes more crisp and valuable. But they tend to push people step by step to more extreme options. You liked marijuana. You like drugs. Maybe you like cocaine? You visited a site with conspiracy theories. Well, here is another one which is even more incredible. When these profiles are made available to advertisers at a premium price, psychometrics such as used by Cambridge Analytica (and others), these allow to influence subsets of the population in both subtle and crude ways.

These selfish business practices continuously raise fundamental societal questions: how do we feel about social media being used by foreign state actors to influence democratic elections through very personalized (and misguided) political campaigns? And how do we contain the algorithmic pressure towards global extremes, rather than brings people together as one would expect from a social network?

Another problematic issue to address is monoculture. Social networks do not allow to cross the boundary of their service in an easy way, leading to social lock in and a "winner takes all" scenario. This limits choice, but also exposes users to legal dangers. Confidential discussions through "private" messages for instance turn out to be not so private, such as the case where a United States got the social network Twitter to hand over the personal communication from European human rights activists and a member of the Icelandic parliament over a severe human rights violation by the USA military. The European Court of Human Rights would certainly not have allowed this, but it happened outside of our jurisdiction - even if all the actors never left Europe.

The federated universe, abbreviated to fediverse, wants to offer social media users a more transparent, ethical and decentralized environment to talk, find and connect. This is done through a plethora of completely independent servers hosted by organisations and individuals around the world. Each has their own policy, each has their own community and reputation. But they can all interoperate. If you don't like any of the existing options, or want to do something different or innovative, you download some open source software and start your own. If you feel some server is toxic, or misbehaves, it just takes one click to stop listening to what is being said. And there is no need to share data with anyone, if you want to. Every node can essentially be a complete social network in itself.

The fediverse is not confined to what a single company wants to do - in every way. That means a broader offering in terms of design, usability and user experience, in terms of technology, ethics and culture. Essentially every server is a full-fledged social network in itself, able to talk to other social networks when it wants. People can use the fediverse for traditional social networking, but they can also integrate it with other services such as online video sharing, all without the fear of having their data being monetized or their activity profiled. Switching from closed social networks to the fediverse contributes to privacy and trust, by enabling users to understand and control who sees their data. The fediverse as a network of social networks, is also more resilient than a single network could ever be.

GNU social is a federated social network built around privacy and trustworthy technology, as the GNU operating system and project are built on free software (free as in freedom, specifically to run, study, change and distribute your software as you please). GNU social is technology for communities to run and host their own social media. This project will add features to easily and flexibly create groups and tag, filter and connect so you can connect with anyone that shares your interest, all under your own control and without infringing your privacy.

Technical description

GNU social is a free social networking platform, easily self-hostable and highly accessible, that enables

both private and public decentralized communications. With NLnet NGI Zero's support, the project is undergoing a change of main focus from microblogging to groups and tags. With this, GNU social will be a space for communities where users can express their passions and explore new ones. Users will be able to immerse themselves in easily filterable content relevant to their interests, and to create and join communities. It's hard to pinpoint an existing alternative service that promotes the same level of functionality in terms of tagging, filtering and connecting with people that share common interests. Especially considering the available degree of accessibility, customization and expansion via plugins.

GNU — **Visit https://NLnet.nl/project/GNUSocial**

NGI0 Discovery **ActivityPub** **Server** **SocialMedia**

PixelDroid

After you take a picture of your brand new car, your smiling baby or the food you were just served, what do you do? You want to show it to everyone you know of course. But do you really know who you are actually sharing your private snapshots with when you post them online? With high grade cameras in nearly every mobile phone and numerous instant messaging apps and social media platforms available, sharing photos is just as easy (and perhaps more popular) than typing out what you want your friends and family to know about your life.

Social platforms and apps make us feel like we are only sharing our images with our own social circle and maybe some faraway friends we met online. But because many so-called 'free' social sharing tools like Instagram actually monetize your data and online activity to sell you personalized ads, your online picture book may not be so private at all. And where do those snapshots, that sometimes contain very personal information about where you live, what you are doing and who you know, actually end up after you clicked that upload button?

When you want to show someone your holiday pictures, you simply want to share those pictures, instead of also handing over a copy to the postal service to check where you went to and possibly send you a cheap flight deal for the coming holidays. Pixelfed is a platform that makes this possible on the internet. Users can choose to run and host the service themselves or choose someone they trust to store their pictures and private data with. No one will track what photos you share and which people you follow. The pictures your friends and family share pop up in your timeline one after the other, without ads or algorithms that decide what you can and cannot see.

Since smartphones have become the everyday computer for a lot of users (sometimes the only device they own), privacy-friendly social media need proper apps to offer the same kind of functionality less privacy-friendly networks offer. PixelDroid is an Android-app for Pixelfed that offers the same functionalities as the website does. This project will add features that fit with federated social networking, like using multiple accounts across different instances. This way Pixelfed is an all-round

competitor for anyone who would rather share their pictures and videos with friends and family in a privacy-friendly way.

Technical description

PixelDroid is an Android client for Pixelfed, the federated image sharing platform based on W3C ActivityPub. Our goal is to bring the Pixelfed platform to Android and provide a mobile user experience that excites. We aim to provide feature-parity with the Pixelfed web client as well as add additional features - like image and video editing, capturing and uploading directly from the app. During the project we will also make it easy to use multiple accounts, even across different instances. Additionally, we want to contribute to the Pixelfed API with testing and additional documentation.

Visit https://NLnet.nl/project/PixelDroid

NGI0 Discovery **Android** **ImageSharing** **MobileApp**

Folksonomy engine for the food ecosystem

When you go do your groceries, how do you decide what food you will buy? Most people rely on a mix of familiarity, habit and package texts to find out what products fit in their diet, what is missing in their cupboard or simply what they want to try out. But what about the millions of people with allergies or strict health-related diets? Doing your groceries becomes a more difficult when you need to be sure you are not buying anything that might trigger potentially very dangerous allergic reactions.

People turn to the internet for information about health and food, but are often confronted with either conflicting opinions or commercial apps and databases that are after their personal information. What you eat tells a lot about who you are, like a kosher or halal-diet clearly indicates your religion affiliation. Finding out what products you can and cannot buy, should not mean you have to disclose very personal information with all sorts of untrusted third parties. Unfortunately, this can happen when you log in to a website of a supermarket chain and filter their offer based on your personal diet.

Because our diet and our health is our own information, we deserve open and public information about food we can search through freely. Open Food Facts is an effort to collaboratively build such a database and currently contains open data on 1 million food products from around the world, independent from the food industry and commercial interests.

The Food Folksonomy Engine is search technology built on top of the Open Food Facts database that allows anyone to search, create and share new kinds of data on their own terms and for their own uses. This way we can share and look up information about the food we eat and create without relying on not so private platforms and middlemen.

Technical description

Everybody is interested in the food they eat, by many different aspects, ranging from taste, cost, ingredients and nutrition to its impact on health, the environment and society. We also happen to have many different names for the same food, the way we prepare it and other properties - sometimes only used very locally. That means it is not always easy for everyone to effectively search open data sets like OpenFoodFacts. Open Food Facts - sometimes referred to as the "wikipedia for food products" - is the biggest open food-database in the world.

The Folksonomy engine for the food ecosystem created within this project will unleash an ocean of new data and uses regarding food. Citizens, researchers, journalists, professionals, artists, communities, and innovators will be able to define and add new properties of their choice to food products on Open Food Facts for their own use or to enrich the shared knowledge. Open Food Facts already feeds hundreds of data reuses. Thousands more will become possible thanks to the new user defined properties.

Open Food Facts — **Visit https://NLnet.nl/project/FolksonomyEngine**

NGI0 Discovery **Discovery** **Food** **Foundation** **Ingredients** **OpenFoodFacts** **Tagging**

O ff e n

As you fire up your computer, laptop or smartphone and click your browser icon to connect to your favorite site, do you know what happens behind the scenes? Many websites actually have dozens of different trackers, and some of these have such a global presence that they can form a pretty clear picture of ones online behaviour. Some argue that privacy is and has been dead for quite some time. As long as users have a quick internet connection and can access the web, email, games and messages without a hitch, they won't complain. But if you question people about the importance of online privacy, usually the answer is that it is indeed important and should be better protected. What is happening here? Perhaps we misunderstand carelessness with unfamiliarity. The technology behind most of our devices, our connection to the internet and the virtual spaces we inhabit is complex, yes, but the solutions we use to access them have also kept actual control away from us under the guise of 'intuitiveness' and 'pick up and play'. Playing here means playing by the rules of the developer, not by your own. What users instead should have are tools that give them actual access to what their devices do, what choices are made, and decide for themselves whether they agree with them or not.

Privacy isn't dead, we just lack the tools to actually protect it. On the internet this would mean users

need tools that first give them behind the scenes access and show how they are tracked and profiled. Then they should be able to flip a switch and decide, no, I don't want some unknown company to follow me around and record everything I do.

This is what offen is creating: a tool that gives users just as much insight and control as a website owner has over data gathering and analysis - putting both on an equal footing. And the user remains in full control: before any data is actually collected, users can see precisely what would be collected about them, and who that data would be shared with. Since website owners needs to convince the visitor that they will respect her or his privacy in order to get their explicit consent, the rather than brutally grabbing any data she or he can get how that affects their privacy. Then they can either opt-in. This way, users have the tools and the access they need to make informed choices online and web site operators, who usually just want to know how many views their site gets and where their visitors are coming from, can respect the choices of their viewers.

After supporting development of offen for components like end-to-end encryption and proper accessibility, NGI Zero now funds work to improve performance, allow sharing of data across instances and more to make this privacy-friendly analytic solution more robust and reliable.

Technical description

Transparently handling data in the open creates mutual trust: Offen is a fair web analytics software that gives users insights into the data they are generating by giving them access to the same suite of analytics tools site operators themselves are using. One unique aspect of Offen is requiring user consent before collecting any data. Especially in countries that are governed by GDPR and its siblings this is a real world requirement for many websites. This is not only about collecting data, but also about embedding third party content or similar.

Usage metrics come with explanations about their meaning, relevance, usage and possible privacy implications, and also details which kind of data is not being collected. Users can expect full transparency and are encouraged to make autonomous and informed decisions regarding the use of their data, and operators are being enabled to collect needed usage statistics while fully respecting their users' privacy and data. No user data is being collected until the user has explicitly opted-in. All data can be deleted either selectively or in its entirety by the users.

Visit https://NLnet.nl/project/OffenOne

NGI0 Discovery **Analytics** **Opt-in** **WebApplication**

Peertube-Desktop

In the same year when the ARPAnet (the predecessor of the internet) was invented, people tuned into their tube televisions to watch a global live broadcast of astronauts first landing on the moon. If they missed that historical moment, that would be it. There was no ability for normal people to record television broadcasts, no ability to rewind or look back programmes from the online guide. At the turn of the millennium, three decades later, everyone was still watching traditional television: quite a few people may have had a video recorder, but this needed to be programmed in advance or you would still miss your favourite tv programme. And there had better not be two programmes you would want to record at the same time.

That has all changed in recent years. On demand video via the internet has meanwhile assumed an important, but also somewhat controversial role. A tiny set of dominant online video hosting platforms (most people would have trouble naming more than two) has emerged, these control how hundreds of millions of users spend many billions of hours of human lives every year. The platform's features and algorithms determine what you see, who can be discovered (whether this is called "trending", "recommended" or "autoplay"), who is banned and deleted, and who is just left out of the spotlight. Users can only follow the patterns laid out for them on screen. The platforms also determine what information is logged about your searches and binge viewing behaviour, and privately decide who they sell your interests and location to. That is a far cry from the privacy granted by traditional television and radio broadcasting, where literally noone outside of the room could know which programme you would pick from the ether. What data is tracked, and what filters and algorithms are used by these online video platforms, remains opaque for users. Contrary to traditional media, the platforms feel no responsibility for checking facts: they focus on commercial value to them, not social value.

To move away from these self-serving monopolies, we need alternative infrastructures to host and share our own videos with. Something like our own private television channel where we decide what to broadcast and tune in to, without advertisements, tracking and profiling. That is what PeerTube allows you to do: peer-to-peer open source technology that lets you set up a turnkey video platform for your own content (and with your own rules). Videos Videos are stored by each instance independently, and so there is no censorship or systemic bias.

There is a lively community of Peertube-instances and audiences sharing and enjoying content, from tv shows to lectures and music, and a host of clients and programs to watch it all on your phone or laptop. What Cuttlefish adds to this, is letting users support the PeerTube-instance they are watching simply by using the program. When you want a video on a PeerTube-instance, you are a peer sharing bandwidth with that instance to make sure the server can manage a lot of users streaming the same content at once. But when you are done watching and close the tab, all the downloaded video data is lost and you are no longer sharing with the peer-to-peer-network. Cuttlefish instead allows you to keep the files of watched videos as long as you want to, relieving pressure on small instances. This way the video player does not only provide a fun and seamless user experience, users are also contributing to a stable and available network for others to watch videos as well. This can help to make PeerTube more fault-resistant and attractive to new viewers and content creators who are fed up with the increasing control video platforms hold over their content.

Technical description

Cuttlefish is a client for PeerTube that will allow for searching and discovering new and interesting video's online with more privacy. PeerTube is a federated video hosting service based on the W3C ActivityPub standard. By using WebTorrent - a version of BitTorrent that runs in the browser - users help serve videos to other users. Cuttlefish is a desktop client for PeerTube, but will work on GNU/Linux-based

phones (like the Librem 5 or Pinephone) as well.

We want the experience of watching PeerTube videos and using PeerTube in general to be better, by making a native application that will become the best and most efficient way to hook into the federation of interconnected video hosting services. It will have improved search, and will allow people to continue sharing watched videos with other PeerTube users for longer periods of time, instead of discarding the video when done watching. It will also help bridge PeerTube's gap between the - now separated - BitTorrent and WebTorrent networks by speaking both of those protocols.

Visit https://NLnet.nl/project/PeerTubeDesktop

NGI0 Discovery DesktopClient P2P Video

S E A R X R

Virtual, augmented or 'enhanced' reality are all terms for a range of technologies that have been long in the making, but now seem to become more commonplace with the rise of smartphones and video screens. Combining digital experiences with the real world open up a lot of possibilities, from new gaming experiences (remember the Pokemon Go craze?) to industrial design, healthcare, education, the list goes on.

The problem is however, that like most consumer technology, the tools and devices you end up using are usually not transparently designed and essentially black boxes that give you a shiny user experience, but little control over how it actually works. This can raise issues of security and privacy which becomes especially problematic when this technology is used in for example healthcare.

Innovation should not come at the cost of privacy, safety or data governance. The same goes for augmented and virtual reality, which does not need to be a tool for surveillance to offer useful and imaginative experiences. This project wants to make online search on augmented reality devices like VR headsets more privacy- and user-friendly. As search is an important starting point for any online device, this can be a first step to building augmented and virtual reality experiences more reliable and trustworthy.

Technical description

SearXR brings a beautiful, privacy-respecting search to 2D and 3D devices. Why? Because searching on alternative devices (VR headsets, conference-presentation) is not always easy nor private. SearXR aims to provide alternative search interfaces which are more appropriate for VR, AR and big screens. SearXR aims to progressively enhance these search experiences: better screen-layout, privacy, and WebXR compatibility. All features are based on user preferences and available hardware. Built upon SearX and W3C's WebXR technology, it will enable everybody to search, or add XR-features to their SearX instance. Whether it be state of the art headsets, or a 65” screen: pointing the browser to an SearXR-instance will immediately launch a wonderful, privacy-respecting search experience.

2WA — **Visit https://NLnet.nl/project/SearXR**

NGI0 Discovery Annotation EnhancedReality SearX SearchEngine UX WebApplication WebXR

Keyoxide

Our identities are scattered across the internet and the web: we have accounts on all kinds of services, from social media to knitting communities, from music websites to video portals, instant messaging services and code hosting. And every time a new service is launched or we join some new community, we run the chance that someone has beaten us to it and has picked up 'our' name. Many of our given names are chosen for traditional reasons, for instance by naming a child after a grandparent or uncle. Other given names happen to be in fashion. And there are just so many family names, meaning that names are not so unique as we need them to be. So how are people to know which Bob Jones, Maria Bernard or Jing Lǐ they are looking at? And with billions of people making up names simultaneously, even fantasy names are not safe. Not to mention that some people take existing names on purpose for malicious reasons ... and all they have to do for people to believe them is to copy a photo and a bit of text.

Keyoxide is a solution to this pressing issue. It functions a bit like a notary, by allowing you to prove in an automated way it is you behind these different accounts and names (only the ones you feel comfortable combining, of course) And you can leave additional secrets there for people to automatically pick up, like for instance a public key to encrypt messages to you or a private messaging channel. That way, people that want to get in touch with you can contact you confidentially without hassle or additional setup for either side.

You can of course run Keyoxide on your own domain, but you can also use a Keyoxide instance you trust - an educational institution, an NGO or a friend. That way there is an independent authoritative source of information about you that people can verify, providing additional resilience against identity theft.

Technical description

How do you discover which other online accounts across different services and service providers actually belong to the same person? Keyoxide is a secure, privacy-friendly and decentralized platform to manage online identities, uncompromisingly driven by what the user herself wants to share.

Keyoxide is a new type of service to allow proving linked account ownership on a variety of platforms.

Keyoxide levers existing and battle-tested cryptographic primitives. The goal is to give users more control over their online presence, independent from dominant internet actors - without in fact having to depend on any centralised services or third parties. The project will improve the usability of the current Keyoxide, and its emerging underlying technology (Decentralized OpenPGP Identity Proofs). More service providers will be added and additional tools to provide proofs will be developed, to create a smooth and easy onboarding process for less tech-savvy people.

Visit https://NLnet.nl/project/Keyoxide

NGI0 Discovery **ClientSideEncryption** **KeyExchange** **KeyManagement** **MessageEncryption** **OpenPGP**

---

f w u p d

Most users rely on antivirus programs to keep their system and important data safe and private. Visited sites, downloaded files, email coming in and out, everything should pass through a digital border control that keeps malware and spyware out. Perform a complete system scan every other month and most users will be reassured: I am safe.

The truth is that there is more than one way into your system and not every backdoor is properly protected. Attackers can also target the most fundamental software on your device, which is also known as firmware. A common example is the BIOS or Basic Input/Output System that every computer has to boot up and load the operating system. Accessing the BIOS and installing malicious software on such a fundamental level gives attackers far-reaching control over a system (which is why it is used for ransomware) and the user usually does not even realize it. And updating their BIOS probably is not something they do (if they are even aware of it at all). That is unfortunate, because a number of hardware vendors do put out updates for their firmware that you can update your computer with.

To make firmware updating more commonplace, you should simply get a notification that you need to get the latest update. That is what this project aims to do for a widely used firmware update effort for Linux-based operating systems. This way users outside of the more experienced small clique of hardware geeks can also be sure their device is trustworthy, from the software they actually run to the programs that start everything up and keep their system going. As Linux-based systems are used everywhere and sometimes perform vital functions to local and wider area networks, a straightforward project like this can actually contribute to a more resilient and reliable global internet.

Security should not be a black box. Instead, users should be able to choose from plug & play solutions that work together nicely and cover most if not all exits in their systems. Or they should have a one-stop-shop solution, a big green button they can press for total security.

Technical description

Security holes in the equipment we run are discovered all the time, and firmware is continuously upgraded as a result. But how do users discover what they need to upgrade to protect themselves? The goal of the "fwupd/LFVS integration in the BSD distributions" is to reuse the effort done by the fwupd/LVFS project and make it available in the BSD-based systems as well. The fwupd is available on Linux-based systems since 2015. It is an open-source daemon for managing the installation of firmware updates from LVFS. The LVFS (Linux Vendor Firmware Service) is a secure portal which allows hardware vendors to upload firmware updates. Over the years, some major hardware vendors (e.g. Dell, HP, Intel, Lenovo) have been uploading their firmware images to the LVFS so they can be later installed on the Linux-based systems. The integration of the fwupd in the BSD-based systems would allow reusing the well-established infrastructure so more users can take advantage of it.

3mdeb Embedded Systems Consulting — **Visit https://NLnet.nl/project/fwdup-BSD**

NGI0 Discovery **Firmware** **SecurityUpdates**

A E A P

One of the things people enjoy the most about the internet, is that it enables them to talk to others remotely almost without limit. Internet allows anyone to keep closely connected with friends and family, and help their kids solve a math problem while they are at work. People collaborate with their colleagues from the couch of their living room, the cafe where they enjoy lunch or on their cell phone on the bus to the gym. Businesses can easily service their customers where this is most convenient to them, without having to travel themselves. This is so convenient, that some businesses have already moved entirely online. Internet communication has become the nerve center of whole neighbourhoods, where people watch over the possessions of their neighbours while these are away for work or leisure.

Email to this day is among the most popular online communication services and is used by governments, companies and organizations to talk to clients and share files. Even though email was designed without privacy or security in mind. When you send an email, anyone that can gain access to your mail server or the mail server of the recipient can read your mail, from top to bottom. And copy it, for later usage. Or modify it. It is often compared to sending a post card, and of course in many cases there may be little harm in others reading what the weather is like in Athens. But what if you want to use email to send something confidential, something you do not want to share with others? Like a love letter, a political rant or an important contract? And what if you can't actually trust the mail man, for instance because the other party is using a free email service known to search through everything? Or what if you live in a

country that has an unhealthy interest in bringing down certain political voices, or are part of a cultural minority that is at risk?

Users could try to host their own email server, or if they are not so technically inclined switch from one of the 'free' email providers (that are usually after your (meta)data and are known to read your messages) to hopefully more trustworthy independent parties that simply charge a monthly fee and in exchange, keep your email safe, private and abuse-free (no spam coming from your address, for example). Switching email providers, unfortunately, is a rather painful experience, which is kind of the point: one way to keep you engaged is to offer an easy to set up account that once you are hooked is hard to get out of with all your data intact.

With the GDPR in mind, data portability should be nothing less than a minimal requirement for any online service. After all, it is your data. Because tech monopolies are yet to offer proper data porting, this project is an effort to offer alternative tools to switch from one email provider to another without losing mail or contact information. That is what this project aims to do for Delta Chat, an end-to-end encryption mail solution that allows you to chat and email privately and securely. Users will not need any information or consent from their existing email provider to have their email and contacts securely transfered to a new, hopefully more reliable provider. And using Delta Chat users would also gain a handy instant messaging app that allows them to communicate encrypted end-to-end to their friends and family.

Technical description

There is no search for email addresses, like there was in the days long gone of the phone book. Once an old contact disappears (e.g. moves jobs, changes provider), even hough you may have exchanged many emails with that person you can not discover which new email address(es) go(es) with that old contact.

The Automated E-mail Address Porting project (AEAP) wants to allows you to find the new email addresses of these existing email contacts. The project will research and develop the porting of an e-mail address to a new provider. We will implement, document, user-test and release a porting mechanism for Delta Chat, a leading end-to-end encryption mail client. Users can decide they want to use a new provider by entering credentials for a new e-mail address. The outcome of the AEAP project will be Delta Chat Desktop, Android and iOS releases to all app stores, providing seamless porting of e-mail addresses. Changing an e-mail provider will not depend on the consent of the existing one. GMail and various other "free e-mail" provider lock-in strategies will be weakened, also through the e2e-encryption that our AEAP effort spearheads.

merlinux GmbH — **Visit https://NLnet.nl/project/EmailPorting**

NGI0 Discovery **Cross-Device** **DataPortability** **E2EE** **Email** **InstantMessaging**

In the 'real world', you instinctively know what information you should keep behind locked doors and what is safe to share. Your bank statements are stored in a folder somewhere in the attic instead of leaving them laying around on your kitchen table. You do not tell random people on the street what your phone number is, or where your children go to school. In the virtual world, this type of common sense can work differently. Users are quicker to trust service providers to keep their personal data safe from theft and prying eyes, and do not always see the dangers of storing passwords in an online text file, or sharing sensitive financial documents via email. The dangers are unmistakably there, but until someone close to you suffers the consequences of a hack or a privacy breach, the risks of online data storage are vague and its convenience is too tempting to pass up. People are accustomed to easy, accessible and convenient online tools and services. More private and secure open-source alternatives should not exclude users because of an overly technical setup or incompatibility with existing proprietary solutions.

Solid (or Social Linked Data) is a new approach to protecting personal data initiated by Tim Berners-Lee, the inventor of the world wide web and developed in collaboration with the Massachusetts Institute of Technology (MIT). The project aims to give users back full control over their personal data, which they can store in personal online data stores (or pods) and then give applications that run on the Solid platform access rights as they see fit. Users always retain ownership over their data, decide for themselves where it is stored and can change the permissions of any application that can access the data. Eventually the Solid ecosystem should offer decentralized and user-centric alternatives to centralized social media like Facebook, Twitter, LinkedIn etcetera.

Convincing people to switch to Solid will take more than just telling them privacy horror stories. You cannot (and should not) scare someone into using your product, no matter how good your intentions might be. The alternative should be as good or even better than the original and switching should be easy and painless. Search and discovery is an important if not vital part of this: you need to be able to easily look for and find your data on a whim when you need it. Solid-Search will lay the groundwork for intuitive search that matches the unique set up of Solid. New search tools and interfaces can then be easily made and put on top of Solid Pods, making this the preferred way of storing and managing your data securely and accessibly.

Technical description

Solid-Search aims to provide an open source module that adds full-text search functionality to Solid pods. Solid is an emergent specification initiated by the inventor of the World Wide Web, sir Tim Berners-Lee. Solid aims to decentralize the web by decoupling applications from databases by introducing Solid Pods (personal online datastores that are in full control of the data owner). Having a way to search through your personal data on your Solid Pod is a must-have for the project to become truly successful. However, this requires technology that does not exist yet: a full-text search interface that works with

schema-less RDF data. In order to maximize adoption and retain a modular, open approach, we will standardize the way in which data changes are described. By doing so, it will be relatively easy to introduce new search / query systems (such as search by location). The project will will create the open source search back-end, improve linked data synchronisation specs, link the module to two solid implementations, create a front-end for end-users, and write a tutorial for adding data sources.

Ontola — **Visit https://NLnet.nl/project/Solid-Search**

NGI0 Discovery **Back-end** **Decentralisation** **FullTextSearch** **LinkedData** **Ontology** **RDF**

Namecoin: ZeroNet and Packaging

If you want to look something up online, send an email to a friend or read the morning news, your computer panics and starts asking for help. How does it know where to retrieve or send anything? Luckily, it is connected to the domain name system. This naming system has been translating names users can remember (like ngi.eu or NLnet.nl) into numbers (or with a fancy word: addresses). Your computer has such a unique number itself, but it needs the numbers of the other computers you want to interact with to connect. You probably use domain names every day, whether you type in the address of a website, listen to a podcast or send an email.

It is called a domain name system for a reason, because it comprises more than just a naming convention. Getting a domain name involves talking to a lot of different computers. Your computer or phone basically doesn't know much about the world. One thing it does know, is how to ask that question to other, specialised computers. These computers actually also probably don't know themselves, unless they have recently answered the same question for another user. Names can change really fast for good reasons, so you would need to refresh this data a lot - otherwise users could end up on the wrong computer. The computers you sent your question to, thus pass the question on to other computers - and so forth. After just a few steps, some of the computers that were consulted get parts of the answer we were looking for. And at some point in time, the domain name system will have the entire answer. The magic happens so fast, most people are not even aware how complex this is. For them it "just works". One disadvantage: many other computers have learned something about us, about who we interact with and about our interests - in an neatly labeled way. Someone is connecting to derspiegel.de or globaleaks.com. The more unique your question, the deeper the digging inside the DNS - and the more it stands out.

Domain names are at present a critical component for users, and so also a critical point of failure and a choke point. Without functioning DNS, most people will have a hard time finding basically anything on the network of networks. There have been cases where for instance a Spanish company got their domain name taken away, even though what they did inside Europe for European citizens was legitimate here. But not in the USA. And since the organisations that handle the .org, .com and .net domain names are based in the USA, these could be forced to remove these names from the DNS.

When DNS was designed, neither security nor resilience was that much of a concern for most users. The internet in its early days was not yet 'open to the public'. This of course has changed dramatically. The massive use of the internet and thereby our dependency on DNS has highlighted very important privacy

and security issues with the design of DNS. At present, it is is not always capable of preventing misleading users nor can it prevent some leakage of what users do, who they talk to and where they go.

To solve these recurring issues with domain names, we could switch from the trust-based setup of stakeholders and decision making organizations to a trustless space. No central points of authority that get to decide who does what with a domain name and whether they are, by their standards, trustworthy enough to be reachable. Instead Namecoin uses blockchain technology to provide a decentralized DNS which already offers a decentralized top-level domain (.bit) that is resistant to hijacking and censorship. This project will use these unique properties for ZeroNet, network of peer-to-peer users that are not identified by a public IP address, but by their public key, specifically a bitcoin address. As these addresses are not readable, just like Tor onion services are not readable (think of www.expyuzz4wqqyqhjn.onion for torproject.org) this raises issues of accessibility and misuse. Namecoin can add a human-readable layer to make these anonymity networks more usable and protecting against phishing attacks.

Technical description

Namecoin provides a decentralized naming system and trust anchor. Its flagship use-case is a decentralized top-level domain (TLD) which is the cornerstone of a domain name system that is resistant to hijacking and censorship. Among other things, this provides a decentralized trust anchor for Public Key Infrastructure that does not require third party trust. It operates independent from the DNSSEC root trust chain, and can thus offer additional security under some circumstances. ZeroNet is a decentralized web-like network of peer-to-peer users, which provides an alternative to TOR hidden services. In the project, Zeronet will be adapted to support a local Namecoin client, and provide additional assurances such as a Host Header-like mechanism to protect users from spoofing. Namecoin will be used as a human-readable naming layer for Tor onion services and ZeroNet sites. This eliminates the user problem of pseudorandom, unmemorable website addresses for onion services and ZeroNet sites, which can facilitate phishing attacks.

The Namecoin Project — **Visit https://NLnet.nl/project/Namecoin-ZeroNet**

NGI0 Discovery DNS DistributedLedger Naming

---

eduVPN on Apple part II

Technical description

eduVPN is a program under the Commons Conservancy, a non-for-profit foundation focusing on free and

open source projects. The goal of the project is to provide a comprehensive and reliable, open source VPN solution for all platforms. The project is plagued by some nasty bugs that have been found hard to fix by the community. This particular project aims to deliver a new and more user-friendly user interface for the macOS and iOS-app, as well as implement a new server discovery mechanism in these apps.

Visit https://NLnet.nl/project/eduVPN-apple-II

VPN Fund EduVPN GUI MacOSX TheCommonsConservancy VPN iOS

JavaScript Restrictor

As you fire up your computer, laptop or smartphone and click your browser icon to connect to your favorite site, do you know what happens behind the scenes? Modern websites offer their users a ton of functionalities, but it is becoming increasingly difficult to know just how all these slick graphics, popups and interactive elements actually work, and what they do precisely. This is very true for most users, but even those more technically inclined may not be entirely sure what happens on their browsers exactly. Not because they lack the knowledge or tools, but because a lot of these little bits of software that come with visiting particular websites are not transparent.

Simply put, you open a site, your browser is sent some programs that immediately run on your computer and you do not and cannot know what is going on. This poses many problems, not just for user agency and freedom, but also for privacy and security when we have some unrecognizable piece of software from some unknown source run on our system, that might hold sensitive personal data or run vital services. Your browser may know how to protect you from harm, but would it not be better to go straight to the source and decide for yourself what you want to run?

JavaScript Restrictor is a browser extension that helps you regain control by doing exactly what its title suggests: restricting unnecessary and potentially harmful programs from running. When a website tries to find out some information about you or fingerprint your browser or computer, JavaScript Restrictor will make sure only false or if possible, no data at all is sent. The best thing is that the tool will try to do this without breaking the website you visit. That way you can continue to browse as you are used to while being sure you are not tracked or profiled.

Technical description

A JavaScript-enabled web page can access any of the APIs that a web browser provides. The user has only a limited control, and some APIs cannot be restricted by the user easily. JavaScript Restrictor aims to improve the user control of the web browser. Similarly to a firewall that controls the network traffic, JavaScript Restrictor controls the APIs provided by the browser. This project has several goals: (1) the

analysis of fingerprinting scripts deployed on the web; based on the study, we want to improve the anti-fingerprinting techniques deployed in the JavaScript Restrictor, (2) improvements in the integration, functional, and unit testing, (3) usability and documentation.

Brno University of Technology — **Visit https://NLnet.nl/project/JSRestrictor**

NGI0 PET **BrowserExtension** **Fingerprinting** **JavaScriptBlocking** **Tracking**

Betrusted Storage

As our lives get more digital every day, we use the internet to have important conversations - both personal and professionally. We also store and share more and more sensitive personal data on devices. On the internet you cannot just close the door to talk privately. So we need digital safe spaces and digital locks and vaults that are just as reliable and easy to use to store our secrets and mediate our communication.

Recently manufacturers have started to build so-called hardware enclaves or secure elements into their devices that function like a digital safe: even if someone is able to get some software installed into your computer, phone or laptop, they should not be able to immediately access what is in the safe.

But of course, creating a secure space or making a digital safe in an environment you don't really control or understand is really hard. All the technical protection no longer matters when someone can invisibly take control or peer over your shoulder. Especially since you as a user can't see yourself what is happening on the inside of your digital house. A safe and a rogue application can and will look completely identical to a user, and there is simply no way to distinguish among them based on their appearance. Users install many unknown games and applications all the time ("install our app to start getting discounts now!"), and forget that this is actually letting more or less random entities run unknown software on the phone that holds some of their most important information. And what if the operating system of your computer or phone itself has an unhealthy interest in your data or metadata, or is weakly protected to that others can just enter - similar to how unsafe it would feel if your landlord or the janitor is a peeping tom or a thief?

Betrusted is a dedicated open hardware project that is pioneering a new class of hardened communication device. It has the goal to create safe and more easily protected private channels for your communication. You can have a frivolous phone to play games, and do all the other things you can use your phone for. A Betrusted device is a complementary device that restricts itself to protecting the things that matter most - like your conversations and phone calls. It will also be able to hold passwords, digital versions of your passport (and other digital credentials and attributes), and whatever sensitive digital information you need to keep completely secure.

The first device spawned by the Betrusted project is aptly called Precursor. Think of it as a Raspberry Pi crossbred with a traditional Blackberry phone form factor, but with strong security features you can verify yourself from top to bottom. Precursor will allow you to be among the first in the world to experience the unique ideas behind the Betrusted project. However, because of its unique form factor,

Precursor is more than 'just' a Betrusted device: it is a framework for DIY fans and developers to build upon. It will also diligently serve your own projects as an ultrasecure 2FA device, a portable HSM, an encrypted team pager, a scientific calculator, a mobile VPN hotspot that tunnels your traffic safely across the internet - or whatever else your creativity may come up with.

After NGI Zero funded the initial work on the Betrusted hardware and software design, this project will further develop a number of core components to ultimately create an easy-to-use and thoroughly trustworthy vault for everything you like to keep safe.

Technical description

Betrusted aims to be a secure communications device that is suitable for everyday use by non-technical users of diverse backgrounds. We believe users shouldn't have to be experts in supply chain or cryptography to gain access to our ultimate goal: privacy and security one can count on. Today's "private key only" secure enclave chips are vulnerable to I/O manipulation. This means there is no essential correlation between what a user is told, and what is actually going on. Betrusted will build a full technology stack, including silicon, device, OS, and UX that is open for inspection and verification. We've passed the first hurdle of creating an FPGA-based device, which we have spun out into a development platform we call Precursor. We are now advancing deeper into the technology stack to improve FPGA, drivers, OS, and UX elements, all driving toward the common goal of making Betrusted a simple, secure, and strong device that aims to advance Internet freedom.

Visit https://NLnet.nl/project/BetrustedStorage

NGI0 PET **Cryptography** **DataVault** **OpenHardware** **RiskModel** **SecureEnclave**

L i b e r a f o r m s

Liberaforms

Technology can make things easier. Instead of having to travel and meet somewhere, you arrange a phone or video call. Instead of sending documents around, you collaborate with your colleagues in an online document. The same thing can be said for online forms, which saves us the hassle of ticking boxes on a page (and prevent a tree getting cut down as well).

Technological solutions, however, can also introduce new problems. In the case of online forms, you can imagine the privacy and security issues when you fill out information about your personal health in a Google- or Microsoft-form, which stores this sensitive data on an unnamed server somewhere you can only hope is managed and updated properly. Unfortunately this is what happens a lot, even for forms you need to fill out to access public services or contact organizations providing vital utilities. This is a quite peculiar blind spot for organizations and businesses that are required to keep your private information safe, but nevertheless use third party technology that handles very sensitive data outside of either your or the organization's control.

This project will offer a transparent alternative for online forms you can create and host on your own, where only you (or a host you trust) manages the information filled in. Because not everyone knows how to host your own technology, Liberaforms aims to make self-hosting forms accessible even to first-timers,

so everyone from a major organization to your local sports club can send out a form and be assured that no data will get lost or forgotten somewhere, but remain completely under your control.

Technical description

Cloud services that offer handling of online forms are widely used, for questionnaires but also for gathering data within schools, associations, volunteer organisations, civil society and even families. While these cloud services (such as Google Forms and Microsoft Forms) can be quite convenient to create forms with, for the constituency which has to fill out these forms such practices can actually be very invasive to their privacy - as many forms not only include personal details such as their name, address, gender or age, but also a lot more intimate questions - up to medical details, political information and life style background. In many situations there is a power asymmetry between the people creating the form and the users that have to supply the data through that form. Often there is significant time pressure. No wonder that users feel socially coerced to comply and hand over their data, even though they might be perfectly aware that their own data might be used against them.

This project will produce a free and libre software solution to create online forms, and to manage the outcomes. The goal is to make something for regular humans: user-friendly, non-intrusive and light-weight. The project aims to make self-hosted form management easy even for novice users, so data can be kept safely on-premise or with a hosting company you can trust. Something that can be used by our neighbours, friends, colleagues and anyone else who respects privacy and understands the moral obligation of the creator of a form to protect the privacy of the people that are supposed to share data with them.

Visit https://NLnet.nl/project/Liberaforms

NGI0 PET **FormServer**

R o b o t n i x

Consumers that go shopping for a new cell phone or tablet these days, at the surface have quite a choice. But the choice is far more limited when it comes to the software that runs on those phones. Pretty much every phone manufacturer (with one notable exception brand of luxury phones) puts Google's Android on it, and while nominally the source code of Android is published under an open source license - in practical terms vendors are very much restricted by contracts with Google and the soft lock-in of the app ecosystem that seeks compatibility with Google's version only to make any significant changes.

The open source community is not tied to the same rules as phone vendors. They have not signed any contracts, and can just pursue what they feel is right - and what users need. As a result a number of 'Altdroids' exist, such as Lineage, Replicant, CalyxOS and CopperheadOS. These are paving the way for more consumer choice, more privacy, more control and configurability and more innovation - with the user's best interest at heart. To set up the infrastructure to build such operating systems is far from trivial

though, and requires a variety of tools for fetching source code and executing builds.

This is significant barrier to entry and an inefficient use of the time of the contributors to these alternative platforms. If we raise the bar from a security point of view, we also want to do more than just build the software. We want to be fully transparent about each adjustment we make, and make it so that we can reuse the work by others - and have others easily reuse our work too. And we want 'reproducible' builds - so that we can verify by building the software independently on different systems that the software we run is actually the software we intended to build. Not many people are aware that build infrastructure from major actors is often heavily attacked by both state actors and criminals, because it is a relatively cheap way to compromise and get access to many end user devices.

This is where Robotnix fits in: it makes it easier for the community to automatically build reliable and reproducible Android and Altdroids. Every package can be followed from the source code, and every patch is visible and reusable. Robotnix is built on the declarative Nix package manager, a powerful tool that can create reliable and reproducible software regardless of the system you are using. Its unique capabilities will make the whole build process much easier and transparent, where instead of switching between a bunch of tools and tricks, you only add a few lines of text to indicate your tweaks. The rest happens automatically: Nix takes your instructions and builds precisely what you want. And if all is well, bit by bit identical, every time over and over.

The operating system is of critical importance, as it forms the basis of everything running on top. The benefits from Robotnix however stretch beyond the operating systems it can build. There are orders of magnitude more people working on apps that run on top of the operating system layer, and these can also use Robotnix - once the OS is there, with the same convenience and assurances it can build any apps that one needs to have built along. Robotnix therefore also fits into a so called continuous integration pipeline, something that makes sure that new features do not break older parts of applications. With the same convenience, developers can support different versions of the OS and different version of their application to support older versions of Android or their app too. So people with an older phone model will benefit from longer and better support too, thanks to Robotnix.

Technical description

Robotnix enables a user to easily build Android (AOSP) images using the Nix package manager. AOSP projects often contain long and complicated build instructions requiring a variety of tools for fetching source code and executing the build. This applies not only to Android itself, but also to projects which are to be included in the Android build, such as the Linux kernel, Chromium webview, and others. Robotnix orchestrates the diverse build tools across these multiple projects using Nix, inheriting its reliability and reproducibility benefits, and consequently making the build and signing process very simple for an end-user.

Visit https://NLnet.nl/project/Robotnix

NGI0 PET **Android** **MobileApps** **Reproducibility** **SoftwareDevelopment** **Transparency**

One of the things people enjoy the most about the internet, is that it enables them to talk to others remotely almost without limit. Internet allows anyone to keep closely connected with friends and family, and help their kids solve a math problem while they are at work. People collaborate with their colleagues from the couch of their living room, the cafe where they enjoy lunch or on their cell phone on the bus to the gym. Businesses can easily service their customers where this is most convenient to them, without having to travel themselves. This is so convenient, that some businesses have already moved entirely online. Internet communication has become the nerve center of whole neighbourhoods, where people watch over the possessions of their neighbours while these are away for work or leisure.

However, users have a hard time to understand how privacy is impacted if they use the wrong technology. Because internet works almost everywhere, the natural privacy protection of the walls of a house, a school or an office is gone. Unlike the traditional phone companies, many of the large technology providers run their business not on delivering an honest service but on secretly eavesdropping on their users and selling information to others. It is mostly not about what you say, so it is relatively easy for providers to allow some form of privacy by encrypting messages. The more interesting parts are who talks to whom, when, and where they are in the real world while they meet on the internet. if you want to be reachable across the internet, you have to constantly let the communication provider follow you wherever you go. This makes the private and professional lives of citizens an open book to companies that with the help of AI and other technologies make billions from selling 'hidden data' normal people are completely unaware of even exists. And of course in societies that are not so democratic, this type of information is critical to bring down opposition and stifle human rights.

Users assume the confidentiality and privacy when they communicate, and they are morally justified to do so. There is nothing natural or final about internet communication providers having access to all this very personal information - or going down the dark path of selling data about customers. The cost of this in terms of internet usage and computer power needed is actually negligible, and so all it takes it the availability of open alternatives that people can use.

Matrix is one of those open alternatives that provides safe and private real-time communication you can set up yourself. The lively community around Matrix offers everything you need to run your own decentralized network in your school, business, with your family or friends. An added bonus is that recently all your chats and talks can be end-to-end encrypted, meaning no one can snoop in (if they are not there in the room with you).

The goal of this project is to add end-to-end encryption support to Fractal, a client or program you can use to communicate through Matrix that focuses on being accessible and intuitive to use. This way you can setup a Matrix-network that properly protects how you communicate with your friends, family and colleagues while not scaring them away with too technical or complicated programs to chat.

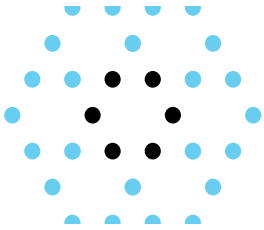
Technical description

Fractal is an Open Source (GPLv3) Matrix client written in Rust. It uses the GTK graphical interface toolkit and is part of the GNOME project. It was created with a big focus on usability and interface design. The objective of this project is to add end-to-end encryption support to Fractal. Fractal has two major parts: A backend part, which communicates with the Matrix server, and a part that contains the GUI and data handling. This will be achieved by first replacing the current backend with the matrix-rust-sdk that was created recently and has several advantages to the current backend, including an abstraction for handling end-to-end encryption for Matrix. Once the backend pieces are in place, Fractal's UI needs to be updated to allow users to actually use end-to-end encryption, which involves a number of non-trivial new user flows (e.g. device verification, cross-signing, key backup).

GNOME — **Visit https://NLnet.nl/project/Fractal**

NGI0 PET DesktopClient GroupMessaging Matrix Messaging

LibreCellular



For many people, especially those living in urban areas, mobile internet and mobile telephony literally just fall out of the sky. It's there, invisibly. But of course, wireless communication is not really wireless: somewhere near, often discretely hidden, there is a powerful antenna on top of some specialised radio equipment. On one end that equipment establishes a radio link with our phone or tablet through electromagnetic waves, and on the other end it connects to the internet and the telephony network - through a fiber optic link or a copper cable.

After that explanation, we might think we understand a bit more how wireless communication works. But we run into new unknowns: how does that equipment actually work? Can we really trust it? How can we tell? Meanwhile, we heard enough warnings about the possibility of mobile network equipment being backdoored, and used to spy on people. And who decides which equipment is put where, what it costs to get access to the network and how our communication is protected? Now compare that to how you use your wifi at home: you also connect to a device over a radio link, but you control that device yourself. You can change the security settings yourself, add any device you want without having to ask for permission, and even take the whole device apart to see what it does. Wifi gives a lot more ownership.

If you consider how much wifi and mobile telephony resemble each other at a technical level, how odd is it that one technology is so close within our reach - and the other is really almost a black box? If someone would tell you that downloading a video your mom posted online would cost you 5 euro to download on your home wifi, you'd politely tell them to go away. If someone told you that they were very sorry but that your wifi is always going to be crappy because you live in a pocket of the network, you'd pick it up and put it somewhere else; or add another antenna, or change the equipment altogether. And if

someone told you they would use your wifi signal strenght to inform some third party whenever you move from one room to the other, you'd be more than right to be angry. So why can't mobile telephony be more like wifi?

This project is about building an open hardware mobile telephony and data network. It allows anyone - of course within the bounds of the applicable legislation - to learn about and experiment with mobile telephony. The goal is to produce a fully functional 4G network, with everything you would need to actually use it as a daily driver - and a lot of documentation to help you on your way. Everything can be inspected and modified from top to bottom. That way, you can make sure your own telecom network is fully trustworthy and private.

Technical description

Free and open source solutions now exist for every component that is required to create a 4G cellular (LTE) network, all the way from the radio access network (RAN) and core, to services which are used for integrated voice (VoLTE). Creating a fully functional mobile network is the next logical step, but this requires overcoming the final remaining technical hurdles. This project will provide end-to-end integration of a FOSS technology stack for 4G networks, via a validated hardware and software configuration that is subjected to appropriate testing. Together with additional tooling and documentation for repeatable deployment, the project will making it far easier to create a self-contained 4G network than ever before. This is particularly timely given the availability of low cost software-defined radio (SDR) hardware, coupled with the efforts of wireless regulators to provide increased access to spectrum for private and community LTE networks.

AB Open Ltd — **Visit https://NLnet.nl/project/4G_FOSS**

NGI0 PET MobilePhone OpenHardware Radio Telephony

SpinalHDL, VexRiscv, SaxonSoc

Consumers and businesses overpay for computer hardware, because the market is not working well. When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component, the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new

setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

Fortunately there are efforts underway to make hardware that, like open source software, is free to be reimagined and reassembled without restriction and that is transparently created, from the design down to the silicone. As these projects grow and connect, they can lay the foundations for a technological commons of trustworthy hardware that is accessible for everyone to learn from and build upon. This project is one of these efforts and will contribute a open source system-on-chip using a publicly available development and design stack. It will be able to run the widely used Linux-system and be deployed on affordable chips and development boards, allowing anyone to quickly create a device that does precisely what you want it to and that you can trust with your information or online service.

Technical description

The goal of SaxonSoc is to design a fully open source SoC, based on RISC-V, capable of running linux and optimized for FPGA to allow its efficient deployment on cheap and already purchasable chips and development boards. This would provide a very accessible platform for individuals and industrials to use directly or to extend with their own specific hardware/software requirements, while providing an answer to hardware trust.

Its hardware technology stack is based on 3 projects. SpinalHDL (which provides an advanced hardware description language), VexRiscv (providing the CPU design) and SaxonSoC (providing the facilities to assemble the SoC).

In this project, we will extend SpinalHDL, VexRiscv and SaxonSoc with USB, I2S audio, AES and Floating point hardware capabilities to extend the SoC applications to new horizons while keeping the hardware and software stack open.

SpinalHDL — **Visit https://NLnet.nl/project/SaxonSoc**

NGI0 PET HardwareDesign OpenHardware RISC-V System-on-Chip

S y l k c h a t

One of the things people enjoy the most about the internet, is that it enables them to talk to others

remotely almost without limit. Internet allows anyone to keep closely connected with friends and family, and help their kids solve a math problem while they are at work. People collaborate with their colleagues from the couch of their living room, the cafe where they enjoy lunch or on their cell phone on the bus to the gym. Businesses can easily service their customers where this is most convenient to them, without having to travel themselves. And sometimes, like when there is a large global pandemic requiring everyone to stay home as much as possible, there is no alternative to moving entirely online.

It still sometimes feels magical to hear the voices and see the faces of the people we talk to across the internet. However, not every way we connect is equally clean and honest under the hood. Users have a hard time to understand how for instance privacy and security is impacted if they use the wrong technology. Because internet works almost everywhere, the natural privacy protection of the walls of a house, a school or an office is gone. Unlike the traditional phone companies, many of the large technology providers run their business not on delivering a fair and transparent service but on secretly extracting data from their users and selling that (together with other derived information) to others.

Ever wonder why advertising companies have paid tens of billions of dollars for buying messaging apps they give away for free, without ads? A lot of interesting behavioral information can be learned without knowing what you say, so the problem is not solved by just encrypting messages. The app itself is the issue: if you want to be reachable across the internet, you have to constantly let the communication provider follow you wherever you go. Other interesting parts are who talks to whom, when, and where they are in the real world while they meet on the internet. Exposure of this data through centralised messaging networks makes the private and professional lives of citizens an open book to companies that with the help of AI and other technologies make billions from selling 'hidden data' normal people are completely unaware of even exists. And of course in societies that are not so democratic, this type of information is critical to bring down opposition and stifle human rights.

Users assume confidentiality and privacy when they communicate, and who can blame them? There is nothing natural or final about internet communication providers having access to all this very personal information - or going down the dark path of selling data about customers. The cost of this in terms of internet usage and computer power needed is actually negligible, and so all it takes it the availability of open alternatives that people can use.

Sylk is one such alternative, supported by NGI Zero: it is a mature private and secure system for video and audio calling that grew from an open source videoconferencing tool. You can install Sylk on your own infrastructure, completely free of charge - and with the ability to make it do whatever you need it to do, on your own terms. Businesses like the internet provider or the IT company around the corner can run it for their customers, or you can run it for your family yourself. You can use Sylk in the browser, or download one of the open source apps for mobile phones, tablets or desktop computers.

In this project Sylk will add privacy-friendly group chat features, that will allow it to become a full-fledged alternative to proprietary solutions like WeChat, Whatsapp and Telegram.

Technical description

Internet communications privacy is important to users, and there is a limited set of encrypted multiparty audio and videoconferencing solutions available to consumers and businesses today. The market, predominantly occupied by proprietary services that often require risky plugins, lack introspection and transparency, proved to expose users to significant security and privacy issues. This trend must be counteracted by better open source equivalents. Sylk provides a multi-party video encrypted

conferencing solution meant to run on an end user computer or a mobile device. It is based on the WebRTC standard, and has a focus on user privacy and easy of use. This project will add one-to-one and group chat capabilities, allowing users to for example have end-to-end encryption or maintain long term group chats like other messaging apps do.

AG Projects B.V. — **Visit https://NLnet.nl/project/SylkChat**

NGI0 PET **AudioCall** **InstantMessaging** **MobileApp** **Videocalling** **Videoconferencing** **WebRTC**

NoScript Contextual Policies & LAN protection (ABE Quantum)

As you fire up your computer, laptop or smartphone and click your browser icon to connect to your favorite site, do you know what happens behind the scenes? Modern websites offer their users a ton of functionalities, but it is becoming increasingly difficult to know just how all these slick graphics, popups and interactive elements actually work, and what they do precisely. This is very true for most users, but even those more technically inclined may not be entirely sure what happens on their browsers exactly. Not because they lack the knowledge or tools, but because a lot of these little bits of software that come with visiting particular websites are not transparent.

Simply put, you open a site, your browser is sent some programs that immediately run on your computer and you do not and cannot know what is going on. This poses many problems, not just for user agency and freedom, but also for privacy and security when we have some unrecognizable piece of software from some unknown source run on our system, that might hold sensitive personal data or run vital services. Your browser may know how to protect you from harm, but would it not be better to go straight to the source and make sure we can actually trust what we run?

To make sure we can browse the web more privately and securely we need control over what our browser is actually doing and what programs it allows to run for which purpose. NoScript is a popular open source browser extension that does precisely this: it gives the user control over what content and programs can and cannot do, while protecting against widely exploited website vulnerabilities. These capabilities are why the privacy-friendly Tor Browser comes with NoScript turned on.

Extensions like NoScript give users back some control over their online experience, protecting them against harmful exploits. This project aims to further strengthen this protection and mitigate other prevalent browser-based attacks while making NoScript easier and more intuitive to use, so that this privacy-friendly technology is not only future-proof, it can also give everyone the safe online experience they deserve.

Technical description

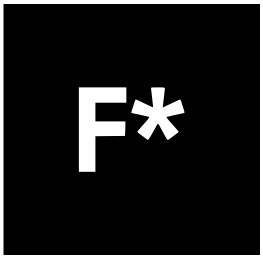
NoScript is a FOSS browser extension for Firefox, Chromium and its derivatives. It can be used on desktop and mobile browsers, and enhances security by providing control over JavaScript and other active content. It is the first and still most effective XSS filter. NoScript is an integral part of the Tor Browser, as the back-end of its "Security Level" settings.

ABE-Quantum is the next generation of the Application Boundary Enforcer (ABE), a NoScript module that provided protection against several cross-site and cross-network attacks. When Mozilla abandoned the legacy Firefox add-ons platform in 2017, ABE did not survive the painful transition to the new cross-browser (but backward incompatible) WebExtensions API. The ABE-Quantum project aims to bring the main ABE features to WebExtension-capable browsers, and specifically: 1) contextual content blocking policies depending both on the origin and the destination of the request, e.g. "Block facebook.net scripts everywhere unless the parent site is facebook.com"; 2) protecting LAN endpoints (i.e. routers or other internal applications) against browser-based attacks from the WAN using the web layer to work-around traditional firewalls. These features will be integrated in NoScript's user interface - rather than leveraging a firewall-inspired policy definition language like in the original ABE - in order to provide a simpler, more accessible and more intuitive user experience.

Visit https://NLnet.nl/project/NoScriptABE-Quantum

NGI0 PET **BrowserExtension** **JavaScriptBlocking**

FemtoStar Project



A lot of users do not really have to consider what it takes to have an internet connection. It Just Works and if it does not, you call up your internet service provider or net maintainer and they send someone around to fix some wires, push some buttons and voila, you are online again. Simplifying complexity has its benefits, as you only need to select a service provider, pay a monthly fee and all the technical complexity is handled for you, behind the screens.

Of course there are also downsides to simple plug-and-play technology. Perhaps the most important thing is that abstracting away complexity can also take away any control or governance over technology you may want to have. In the case of your internet connection, you might want to know who has access to your data, how your traffic can be monitored or influenced, which information can be logged about what you do online. You may think that proper legal protection and consumer rights take care of this. But this is not the case for everyone, especially for internet users living under repressive governments, vulnerable communities like minorities and activists and journalists working in unstable regions.

As an alternative to available connectivity services, FemtoStar is developing a satellite network for

transparent, fundamentally anonymous and censorship-resistant communication you can manage yourself. All you need is a user terminal that connects to the network of FemtoStar-satellites and you are online. No identification or geolocation is necessary to have a working connection, ensuring an actual anonymous internet connectivity for anyone who needs to fully trust the cable they plug into their device before going online.

Technical description

The FemtoStar Project is developing a low-cost communications satellite, intended for use as part of a scalable, decentralized network enabling verifiably anonymous, geolocation-resistant communications on a global scale. While many anonymizer services are currently available to users of existing communications systems, these serve simply to separate knowledge of identity (which still lies with the communications service provider) from knowledge of activity (which lies at the exit of the anonymizer service). All current wide-area communications networks are fundamentally identifying (users and their hardware are, at minimum, pseudonymous to the network) and no two-way communications system offers any meaningful degree of resistance to geolocation of the user. The FemtoStar Project intends to use a constellation of FemtoStar satellites to provide global, space-based open communications infrastructure linking users to services (which can be operated by anyone, and require no special ground station installation beyond a regular FemtoStar user terminal) or directly to other users, and requiring no identification or geolocation of user terminals. We are seeking funding for the development of a prototype satellite and user terminal, implementation and testing of the FemtoStar protocol on this hardware, and, dependent on funding amount and regulatory approval, the licensing and launch of one FemtoStar satellite to low earth orbit for system testing and, possibly, for use in a limited open beta service. With prototype hardware and, ideally, with one production satellite in orbit, the FemtoStar Project will be able to validate the FemtoStar system and move towards our goal of operating a scalable constellation for global, verifiably-private communications service - a world-first in privacy technology.

FemtoStar Project — **Visit https://NLnet.nl/project/FemtoStar**

NGI0 PET **NetworkingArchitecture** **OpenHardware** **RealtimeCommunication**

R N P C o n f i u m

When you mention encryption or encoding, some are quick to think of exciting, sensational and sometimes shady things: spies exchanging secret messages and handshakes, criminals dealing drugs on the Dark Web, black hat hackers hiding in anonimity. But actually, encryption could not be more commonplace. Every time you call someone on your phone, you fire up your browser, send a chat message to friends, do some online banking, you rely on some complex mathematics behind the screen that makes sure you can talk, bank and browse securely and privately. The internet, practically all modern communication technology, could not exist without encryption we can trust to keep our data, our money, our lives, safe.

Encryption, however, will never guarantee complete, 100 percent, total security. Or to put it more

precisely, the encryption schemes and implementations we use today may not be a match for the computers and use cases of tomorrow. That is why this industry is always looking for the next future-proof scheme and solution to essentially prevent a global hack of all communication: a much discussed recent example would be a quantum computer breaking perhaps the most widely used cryptosystem for secure data transmission.

This project aims to advance encryption through something called threshold cryptography, which means you can only prove you are who you are when you have a certain amount of secrets (reach a particular threshold). Think of multi-factor authentication, where you need for example two out of three items like a user password, a one-time password generated by your phone and your fingerprint. Even if one of these items gets lost or stolen, you are still not at risk since you have the other two. This project will provide the tools and architecture to make threshold encryption the standard for secure, private communication, so we can be sure our internet technology is safe against future threats.

Technical description

Confium is an open-source distributed trust store framework that enables usage of the new paradigm of threshold encryption, powering new modes such as cryptographic secure multi-factor authentication. It aims to provide a generalized API and an extensible architecture for the usage of trust stores and future cryptographic families, to support standardization efforts of threshold cryptography, and to bridge cryptographers with the practical usage of cryptography. The current project enables implementation of the Confium framework with a 2-out-of-3 threshold RSA signature scheme.

Ribose Limited — **Visit https://NLnet.nl/project/RNPConfium**

NGI0 PET **Cryptography** **OpenGPG**

WikiRate Insights

WikiRate

Technical description

For too long actionable data about the behavior of companies has been hidden behind the paywalls of commercial data providers. As a result only those with sufficient resources were able to advocate and shape improvements in corporate practice. Since launching in 2016, WikiRate.org has become the world's largest open source registry of ESG (Environmental, Social, and Governance) data with nearly 1 million data points for over 55,000 companies. Through the open data platform anyone can systematically gather, analyze and discuss publicly available information on company practices, joining current debates on corporate responsibility and accountability.

By bringing this information together in one place, and making it accessible, comparable and free for all, we aim to provide society with the tools and evidence it needs to spur corporations to respond to the world's social and environmental challenges. Homing in on the usability of the platform, this project will

tackle some of the most crucial barriers for users when it comes to gathering and extracting the data, whilst boosting reuse of the open source platform for other purposes.

WikiRate — **Visit https://NLnet.nl/project/WikirateInsights**

NGI0 Discovery **Crowdsourcing** **EthicalFilter** **NGO** **OpenData** **SDG** **ServerApplication**

R e d a s h

Technical description

Dasher is an alternative text entry system that searches for suggestions without the discrete input through a keyboard. The software is invaluable to people with disabilities who use it to type or speak and who can't control a regular physical or on-screen keyboard. Dasher is instead driven by continuous gesture using a dynamic predictive display, a concept originally developed by the University of Cambridge Inference group.

The dasher project aims to help all individuals with disabilities who use similar assistive technology by developing a modular word and letter prediction engine that is allows for a range of language models to be used - and new ones be trialed out, including potentially integration with context sensitive search prediction provided by search engine providers. The new dasher will provide a fresh codebase matchin the features that current users require - whilst improving on the user experience for new users. Thanks to a permissive open source software license anyone will be able to develop additional innovations on top of dasher, including commercial entities that produce bespoke systems. This will help increase the ability for employers to hire people that depend on this type of input mechanisms.

Ace Centre — **Visit https://NLnet.nl/project/Dasher**

NGI0 Discovery **Accessibility** **ContextualSearch** **Entry** **NGO** **PredictiveText** **Search**

XMPP-ActivityPub gateway

Technical description

XMPP (aka Jabber) is the vendor-netural internet standard for instant messaging. ActivityPub is a web standard for federated social networking, used in software like Mastodon, Pleroma, PeerTube, Pixelfed and Funkwhale. The project consists of two components: an ActivityPub-XMPP gateway, which will be a component bridging these protocols - enabling ActivityPub users to access XMPP blogs, comments and other features, and vice versa. And adding state of the art end-to-end encryption (E2EE) for PubSub and filesharing, which entails proposing a new XMPP standard which can provide a secure way to publish, retrieve and subscribe to all sorts of data over XMPP.

The project is built on Libervia (previously known as "Salut à Toi"), a communication ecosystem based on XMPP. Libervia offers several interfaces (web, desktop, mobile, command line, text UI) and explores the XMPP protocol beyond instant messaging. Libervia features chat, blogging, file sharing, photo albums, events, forums, etc. Libervia's goal is to develop an all-in-one, easy to use "familial and personal social network", i.e. a tool to communicate with the people close to you securely - and that lets your personal data stay within your control (as it should be).

Salut à Toi — **Visit https://NLnet.nl/project/Libervia**

NGI0 Discovery **ActivityPub** **Gateway** **PubSub** **XMPP**

OCCRP Aleph disambiguation

Technical description

To support journalists to uncover corruption and hold power to account, OCCRP develops Aleph. Aleph is an investigative data platform that searches and cross-references global databases with leaks to find

evidence of corruption and trace criminal connections. The project will improve the way that Aleph connects data across different data sources and how it ranks recommendations and searches for reporters. Our goal is to establish a feedback loop where users train a machine learning system that will predict if results showing a person or company refer to the same person or company. If successful this means journalists can conduct more efficient research and investigations, finding key information more quickly and wasting less timetrawling through irrelevant documents and datasets.

Organized Crime and Corruption Reporting Project — **Visit https://NLnet.nl/project/Aleph**

NGI0 Discovery CrossReferencing NGO NamedEntityRecognition Search

Certbot ECDSA support

Ministerie van Economische Zaken en Klimaat

Technical description

Certbot, part of EFF's larger effort to encrypt the entire Internet, is a free, open source software tool used to encrypt traffic to tens of millions of websites. By automatically generating and configuring Let's Encrypt certificates on webservers to enable HTTPS, Certbot improves the privacy and security of hundreds of millions of users worldwide. The project strives to provide the highest standard of security, which is why we are keen to implement Elliptic Curve Digital Signature Algorithm (ECDSA) support. ECDSA support in Certbot will improve privacy, performance, and trust for Internet users via improved authentication and security.

Electronic Frontier Foundation (EFF) — **Visit https://NLnet.nl/project/Certbot-ECDSA**

Internet Hardening Fund Certificate Cryptography Foundation TLS X509

GNU Guix - Cuirass

CUIRASS

Technical description

GNU Guix is a universal functional package manager and operating system which respects the freedom of computer users. The number of supported packages, almost 15.000 on 5 different architectures, is constantly increasing. With the recent efforts adding support for the GNU Hurd operating system, and the ongoing work to easily provide Guix System images for various boards, the need for a strong continuous integration system is critical.

This project aims to improve Cuirass, the GNU Guix continuous integration software to provide binary substitutes for every package or system image within the shortest time. This way, the user won't have to allocate important time and computation power resources into package building. The plan is to add to Cuirass an efficient offloading and work-balancing mechanism between build machines, an improved web interface allowing to monitor machine loads and other build related metrics. A user account section to setup customized monitoring dashboards and subscribe to build failures notifications will also be developed.

Visit https://NLnet.nl/project/Cuirass

NGI0 PET **ContinuousIntegration** **ReproducibleBuilds**

E E Z D I B

Technical description

The aim of the EEZ DIB project is to enable the creating and management of modular open hardware T&M (Test & Measurement) solutions. Born out of frustration that solutions from reputable manufacturers are feature rich but closed in design and with expensive software licenses, an attempt have been made to fill the gap between such solutions and DIY/hobbyists solutions which although often open in design lack structure, documentation and completeness that could ensure further growth, development and support.

The hardware part of the project is EEZ BB3, an open source DIB chassis in a compact format that can accommodate up to 3 peripheral T&M modules which can be monitored locally via touchscreen display with responsive and attractive user interface or remotely via USB or Ethernet using Telnet, MQTT, JS and Node-RED. Additional autonomy and programmability has been achieved by adding support for MicroPython scripting.

The software part of the project is EEZ Studio, a free and open source cross-platform application that has two functions: a) visual editor that simplify and accelerate touchscreen GUI development and b) management of multiple EEZ BB3 and 3rd party T&M devices for the purpose of simple communication and acquisition, search and presentation of measurement data.

Envox d.o.o. — **Visit https://NLnet.nl/project/EEZ-DIB**

NGI0 PET **Measurement** **OpenHardware** **Testing**

Movim

Technical description

Movim is a web platform that delivers social and IM features on top of the mature XMPP standard (aka Jabber). Unlike other chat apps, with XMPP you have a choice of both servers and clients - and the ability to add any features you want, and restrict your trust to those that deserve it. Movim is a user-friendly communication platform aimed at small and medium structures (up to a hundred simultaneous users), and sports a number of unique social features beyond instant messaging. And because it sits on XMPP, Movim users can explore the whole global instant messaging network from a single account.

In this project, Movim will add end-to-end encryption to its chat interface, in this case the OMEMO XEP. Since Movim is browser based, the implementation will be have to put the encryption layer client-side - or in other words, inside the browser. Because users can connect simultaneously on the same XMPP account using different browsers with Movim, each browsers will be seen as a different "device". Decrypted messages will be saved in a browser database, using IndexedDB. The web server will just take care of handling public keys to the XMPP network and store the encrypted messages, same as the user's XMPP server does when using archiving methods. The project will deal with both the one-to-one chat implementation and the Multi-User Chat part of Movim. This is part of a concerted effort to create reliable end-to-end encryption for XMPP based real time communications. At present growth of the wider network is hampered by lack of interoperability.

Visit https://NLnet.nl/project/Movim-OMEMO

NGI0 PET OMEMO WebApp XMPP

Structuring the System Layer with Dataspaces

Technical description

The system layer is an essential but often-ignored part of an operating system, mediating between user-

facing programs and the kernel. Despite its importance, the concept has only been recently recognised and has not received a great deal of attention. The novel Dataspace Model of concurrency and communication combines a small number of concepts to yield succinct expression of ubiquitous system-layer features such as service naming, presence, discovery and activation; security mechanism and policy; subsystem isolation; and robust handling of partial failure. This project will evaluate the hypothesis that the Dataspace Model provides a suitable theoretical and practical foundation for system layers, since a well-founded system layer is a necessary part of any vision of secure, securable, resilient networked personal computing.

Visit https://NLnet.nl/project/Dataspaces

NGI0 PET **OperatingSystem** **SystemsProgramming**

B B B s e c u r e C h a t

Technical description

BigBlueButton is a video conferencing framework built on open source components. It is being used worldwide for education, events and training, and gained a lot of usage during the Covid-19 pandemic. Whilst audio and video are being handled by scalable components (notably Freeswitch and Kurento), the chat currently integrated in BBB is a single node.js thread for all conferences. This causes performance problems if used heavily in conferences, and lacks features such as E2EE and emoji support. In this project we will be trying to create an alternative chat service component based on mature open source solutions which have a richer feature set and offer end-to-end encryption. Some of the challenges are: respecting privacy in recordings, allowing chats 1:1 and in break-out rooms, automatic exchange of encryption keys, authentication, SingleSignOn and handling file exchange among chat users. We will be testing the enhanced chat with selected BBB users and will offer the result to the BBB developer and user community.

fairkom — **Visit https://NLnet.nl/project/BBBsecureChat**

NGI0 PET **InstantMessaging** **Videoconferencing** **WebRTC**

M o b i l e A t l a s

Technical description

MobileAtlas is an international measurement platform for cellular networks that takes roaming measurements to the next level. Although mobile cellular networks have become a major Internet access technology, mobile data traffic is surging, and data roaming has become widely used, well-established measurement platforms (e.g., RIPE Atlas) are not well-suited for measurements in the mobile network ecosystem. This includes measurements of metered connections and consideration of roaming status and zero-rating offers.

MobileAtlas implements a promising approach by geographically decoupling SIM card and modem, which boosts the scalability and flexibility of the measurement platform. It offers versatile capabilities and a controlled environment that makes a good foundation for qualitative measurements. We want to establish the framework with at least twenty open hardware probes, and create a platform for shared usage among scientists and Internet activists.

SBA Research — **Visit https://NLnet.nl/project/MobileAtlas**

NGI0 PET **Measurement** **MobileInfrastructure** **OpenHardware**

Timing-Driven Place-and-Route (TDPR)

Technical description

The lack of an open-source timing-driven place-and-route tool is one of the major barriers to creating technically fully transparent digital integrated circuits such as microprocessors. The most popular open-source place-and-route tools available today are not timing-driven, hence the generated layouts are generally not guaranteed to satisfy the timing constraints. This requires tedious and time-consuming manual interventions. This project will combine published algorithms with existing open-source projects to fill this gap. The tool will be released with the free/libre AGPLv3 licence together with extensive documentation and tutorials.

Free Silicon Foundation (f-si.org) — **Visit https://NLnet.nl/project/TDPR**

NGI0 PET **Foundation** **HardwareDesign** **OpenHardware**

GPG Lacre project

Technical description

This project is the continuation of the work on providing open source, GnuPG based email encryption for emails at rest. All incoming emails are automatically encrypted with user's public key before they are saved on the server. It is a server side encryption solution while the control of the encryption keys are

fully at the hands of the end-user and private keys are never stored on the server.

The scope of the project is to improve on the already existing code, provide easy to use key upload system (standalone as well as Roundcube plugin) and key discoverability. Beside providing a solution that is easy to use we will also provide easy to digest material about encryption, how it works and how to make use of it in situations other the just mailbox encryption. Understanding how encryption works is the key to self-determination and is therefore an important part of the project.

GPG Mailgate will be battle tested on the email infrastructure of Disroot.org (an ethical non-profit service provider).

Stichting Disroot.org (https://disroot.org) — **Visit https://NLnet.nl/project/GPGLacre**

NGI0 PET **Email** **Encryption** **Foundation** **KeyExchange** **OpenPGP** **Server**

Waasabi Framework

Technical description

Waasabi is a highly customizable platform for self-hosted video streaming (live broadcast) events. It is provided as a flexible open source web framework that anyone can host and integrate directly into their existing website. By focusing on quick setup, ease of use and customizability Waasabi aims to lower the barrier of entry for hosting custom live streaming events on one's own website, side-stepping the cost, compromises and limitations stemming from using various "batteries-included" offerings, but also removing the hassle of having to build everything from scratch. Active research into the creation of a peer-to-peer streaming backend seeks to advance the project's long-term goal of promoting the adoption of owned experiences through the use of decentralized technology. By further cutting down on dependencies, cost and infrastructure complexity this effort aims to enable broadcasts to scale as the audience size grows, which in turn will support Waasabi's continued adoption.

MTÜ Bay Area Tech Club — **Visit https://NLnet.nl/project/Waasabi**

NGI0 PET **IPFS** **P2P** **Videostreaming**

Solid Control

Technical description

Solid-Control aims to enhance Tim Berners-Lee's Social Linked Data Project (Solid) with Attribute-Based Access Control. By extending the Linked Data Platform (LDP) with WebID based authentication and Access Control Lists (ACL), Solid has enabled the emergence of new forms of Hyper-Apps. These apps can follow data from server to server, authenticate when needed and write to the user's Personal Online Data storage (Pod), creating a decentralised social web.

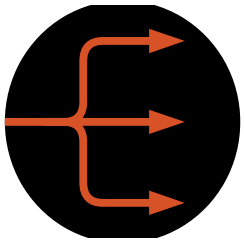
With relation-based access control (friend of a friend, business network, etc.), Solid can be a full alternative to centralised social networks. We also want to allow authentication based on Verifiable Claims such as age. Solid-Control will work on developing the needed logic, verify protocols, write prototype implementations and contribute to the Solid Auth Community groups, which are developing specs for standardisation.

Cooperating Systems UG (haftungsbeschränkt) — **Visit https://NLnet.nl/project/SolidControl**

NGI0 PET AccessControl Client FormalProof Library Server Solid StandardSetting

---

Edalize ASIC backend



Technical description

Affordable Open Source ASIC development and custom silicon has been a long-standing goal in the community. This will unlock innovation that has previously only been possible for the largest tech companies, allowing for the creation of deployable, trusted Open Source based hardware.

Step by step, this goal has come closer in the last few years as individuals, companies and academic

institutions have filled in the missing pieces. Today we have a fully open source end-to-end flow for building open source ASIC - but the effort of on-boarding existing designs remains high. This project aims to provide an easy way to onboard existing gateware and full designs to an open source ASIC flow by creating a FuseSoC backend that targets this toolchain. This will enable a smoother transition from projects already running on FPGAs to also be targeting ASIC flows. It will also allow easier switching between different open source ASIC flows at the point when there are several alternatives to choose from.

In addition to the backend itself, a reference design containing SERV, the world's smallest RISC-V CPU, will be run through the flow and committed to actual silicon. This will provide a way to guarantee a working flow and provide a simple but usable reference for everyone else looking to onboard their designs. Enabling and demonstrating this path will allow a fully trustworthy path for the fabrication of system-on-a-chip ICs, with no proprietary or closed tools as part of the flow and hence completely inspectable at all stages. This paves the road for other more complex FuseSoC-based open source silicon projects such as OpenTitan and SweRVolf.

FOSSi Foundation — **Visit https://NLnet.nl/project/Edalize-ASIC**

NGI0 PET CPU HardwareDesign OpenHardware RISC-V System-on-Chip

---

Libre/OpenCores FuseSoc backend

Open collaboration (like for example on open source software) is based on the premise that together, we know more than we do alone. For open source software development, there is a long history of tools and infrastructure that you can easily setup and maintain for your project, so you can involve as many viewpoints and contributions as you can to make your program versatile, secure, user-friendly, creative, and so on. What's more, the community created around a project can keep software going long after an initial creator has left, updating and expanding it as needed.

For open source hardware, development is often more restricted, where businesses and researchers are forced to reinvent the wheel again and again. This does not only make bottom-up innovation unnecessarily expensive and uncompetitive, it also wastes precious resources. This project makes it easier for innovators to reuse chip designs and focus on how existing components can be combined to do exciting new things, speeding up innovation and research and development.

Technical description

Chip (FPGA/ASIC) development is normally done in a very hierarchical manner where gateware is used to build up subsystems which are combined to a full chip design. On paper, this leans very well for reusing parts in many different chip designs, but the actual amount of reuse has always been hampered by the lack of tooling to manage and combine gateware. Compare this to the software world where languages such as JavaScript, Python or Rust have a rich ecosystem of user-created reusable parts that can be used as a base to quickly build new applications. This project aims to provide a similar ecosystem for chip development where users can publish their cores, find the cores they need and build upon these to rapidly create new designs.

LibreCores — **Visit https://NLnet.nl/project/FuseSoc-Cores**

NGI0 Discovery ASIC FPGA Foundation Repository

B a b e l i a

Technical description

Babelia is a privacy friendly, decentralized, open source, and accessible search engine. Search has been an essential part of knowledge acquisition from the dawn of time, whether it is antique lexicographically ordered filing cabinets or nowadays computer-based wonders such as Google or Bing. From casual search to help achieve common tasks such as cooking, keeping up with the news, a regular dose of cat memes or professional search such as science research. Search is, and will remain, an essential daily-use tool, and steers human progress forward.

Babelia aims to replace the use of privateer search engines with a search engine that is open, hence under the control of the commons. Babelia wants to be an easy to install, easy to use, easy to maintain, no-code, personal search engine that can scale to billions of documents, beyond a terabyte of text data, for under €100 a month per Babelia instance.

Visit https://NLnet.nl/project/Babelia

NGI0 Discovery Crawler PersonalSearch

P e e r T u b e

Technical description

PeerTube is a free, libre and federated video platform. Video is a very popular class of content and

meanwhile accounts for a signicant share of internet traffic, but the choice of hosting has a lot of implications - if you send your viewers to some proprietary platform because you want to avoid cost, what happens after they watch your video? And who watches them watch? PeerTube allows for a federation of interconnected hosts (so more choice of videos wherever you go to see them) while containing the risk of exposing users to profiling, algorithmic pressure that favors extreme content, censorship and other negative aspects of centralised services like YouTube or Vimeo. PeerTube implements the ActivityPub standard and works with peer-to-peer distribution - and therefore viewing. This means no slowing down when a video suddenly goes viral, and much lower distribution costs thanks to shared bandwidth. PeerTube aims to make it easier to host videos on the server side, while remaining practical, ethical and fun on the Internet user side. In this project, Framasoft will work on PeerTube 4.0 with interesting new features such as better search, live streaming, channel customisation and improved accessibility.

Framasoft — **Visit https://NLnet.nl/project/PeerTube**

NGI0 Discovery **Federation** **P2Pdistribution** **Server** **Videostreaming** **Webtorrent**

---

x q e r l

Technical description

The xqerl project is an open-source XQuery 3.1 implementation. It attempts to combine the simplicity of the W3C XQuery 3.1 language for querying and building XML and JSON, with the powers of the Erlang language for building massively concurrent, fault-tolerant, distributed applications. Many optional language features have already been added to xqerl, including the RESTXQ specification for building REST endpoints directly from code annotations. To further enhance user experience and the feature-set of xqerl, the "Schema Aware" and "Typed Data Features" will be added. These features will allow for XML Schema documents to be directly referenced from queries and the query statically analyzed at compile time using the schema to either build better query plans or return errors back to the user before running time consuming queries.

Visit https://NLnet.nl/project/xqerl

NGI0 Discovery **Schema** **Validation** **XML**

Irdest

Technical description

How can you search for wireless devices near you to interact with, without other infrastructure present? The Irdest project allows devices such as laptops and smartphones to create wireless mesh networks over Bluetooth and direct WiFi connections, rather than relying on internet access via mobile networks, and traditional internet service providers. It decentralises the routing and peering mechanisms used to connect people together, to allow users to have more control over their digital lives. In addition to this, direct circuits in a Irdest network are end-to-end encrypted, meaning that data privacy is built into the protocol at a fundamental level.

Visit https://NLnet.nl/project/Irdest

NGI0 Discovery **DelayTolerantNetworking** **MeshNetworking**

Corteza Discovery

Corteza

Technical description

Corteza Discovery will render Corteza as a search-oriented architecture. Corteza is an open source Low Code Application Development solution for building records-based management systems. It can be used in a wide array of applications, from Urban Data Platform for smart city management to business applications and CRM. Corteza is capable of many-to-many data federation and WCAG2.0 accessibility is an objective across all components of the solution.

Advanced, permissioned search will be implemented locally, within federations and between federations. Standards-oriented geolocation and mapping will be supported across the platform. The ultimate goal is to create a compelling, modern and friendly UX for users/citizens - yet based on federated, high-utility Low Code applications which have been specifically designed for purposes of data collection, organisation and portability. Search features such as tokenisation, lemmitisation and "more like this" functionality will enrichen user interaction.

From any point of user interaction with any search, to developers building new applications to be searched, Corteza aims to set a standard for inclusive design.

Crust Technology Ltd. — **Visit https://NLnet.nl/project/CortezaDiscovery**

NGI0 Discovery **Federation** **Search**

Bonfire Search & Discovery

Technical description

Bonfire is a modular ecosystem for federated networks. The project creates interoperable toolkits that people can use to easily build their own apps to meet their specific needs. Users are then free to interact with multiple people and groups using these apps hosted on their own device, regardless of what federated software these other people use. Federated topics within the Bonfire ecosystem can consist of a hashtag, a category in a taxonomy, a location, etc. This enables users to find a topic they are interested in, see everything that was tagged with that (publicly or in their network), and follow it to receive any new tagged content. This will be interoperable with existing fediverse apps like Mastodon without requiring extra development on their end, and will create a decentralised graph of topics that can help relevant information flow from instance to instance.

All content on a Bonfire instance (including remote content coming in via follows or federated topics) will also be aggregated in a local search index with which the user can search their own data, information from people or groups they follow, as well as content from topics or locations they are interested in from around the fediverse. This search will happen locally on their device (which is a plus for privacy), with results appearing instantly while typing a query, and being able to filter the results (e.g., by object or activity type, hashtags, topics, or language). Every line of Bonfire's code is available to be used or forked, in a collection of libraries that can be assembled and re-assembled to create all kinds of full-featured apps. One example is Bonfire's mutual aid extension where users can post and search for requests and offers across different instances according to topic and/or geographical location.

Visit https://NLnet.nl/project/Bonfire

NGI0 Discovery **Federation** **Hardware**

The PeARS app

Technical description

It is widely believed that Web search engines require immense resources to operate, making it impossible for individuals to explore alternatives to the dominant information retrieval paradigms. The PeARS project aims at changing this view by providing search tools that can be used by anyone to index and share Web content on specific topics. The focus is specifically on designing algorithms that will run on entry-level hardware, producing compact but semantically rich representations of Web documents. In this project, we will use a cognitively-inspired algorithm to produce queryable representations of Web pages in a highly efficient and transparent manner. The proposed algorithm is a hashing function inspired by the olfactory system of the fruit fly, which has already been used in other computer science applications and is recognised for its simplicity and high efficiency. We will implement and evaluate the algorithm on the task of document retrieval. It will then be integrated into a Web application aimed at supporting the growing practice of 'digital gardening', allowing users to research and categorise Web content related to their interests, without requiring access to centralised search engines.

University of Trento — **Visit** **https://NLnet.nl/project/PeARS**

NGI0 Discovery **Crawling** **Indexing**

WordPress ActivityPub

Technical description

WordPress ActivityPub is a plugin that allows your site users to interact with other users in the fediverse. Currently the plugin supports Follows by remote users, sending out pubilc posts to followers, and receiving remote users public Comments on local posts. This project will develop features allowing for a more rich and typical social experience with Direct messages, Followers only posts, and Threaded comments to and from the fediverse. Moderation tools will be included and user privacy features will also be developed.

Visit **https://NLnet.nl/project/Wordpress-ActivityPub**

NGI0 Discovery **ActivityPub** **Plugin** **Wordpress**

Collabora Online and LibreOffice

Technical description

Today it's usually easier to use a search engine for information than find it locally, which is not optimal from a digital sovereignty point of view. Part of the problem is that we lack good open source tools to provide context and graphical search of local documents. These tools present plain-text lists for search results, which means people with good graphical memory find information slower. We think it's a huge opportunity to show the context of search hits in a graphical form to find information faster. Technically, this will mean taking an existing file synchronization and sharing (FSS) solution, hosting your documents on-site. Then improving LibreOffice to index content in documents with their context. We will build a secure REST API on top of this in Collabora Online which provides good performance. Finally we will integrate with a search engine, e.g. Apache Solr to create a proof-of-concept search page that allows searching in all documents hosted in a FSS solution. This will serve as an example how to integrate our solution to other projects like Nextcloud.

Collabora Productivity — **Visit https://NLnet.nl/project/LO-Online**

NGI0 Discovery **Format** **ODF** **Office** **OpenDocument** **Search** **Thumbnail**

Castopod Mobile

Technical description

Castopod Mobile is a free and open-source mobile podcast player application (GPL v3). It is intended to be installed on your mobile phone (iOS, Google Android, /e/…). You can install it from F-Droid, from your usual app store or you may compile it yourself for your own needs. Castopod Mobile is a two-in-one application: a podcast player and a Fediverse client. It serves several purposes: to provide a mobile application that takes advantages of ActivityPub features for podcasts (the ones that Castopod Server provides for instance). Secondly, to reduce the complexity of the Fediverse ecosystem during onboarding: account creation currently prevents many users into joining the Fediverse because it is difficult to guess where to begin. And thirdly: to provide a podcast application template for communities who want to build and manage their ecosystem from beginning (with your own private Castopod Server)

to end (with your own Castopod Mobile based application).

Ad Aures — **Visit https://NLnet.nl/project/CastopodMobile**

NGI0 Discovery **MobileApp** **Podcasting**

Namecoin: TLS

Technical description

Namecoin is a blockchain project that provides a decentralized naming system and trust anchor. Our flagship use-case is a decentralized top-level domain (TLD) which is the cornerstone of a domain name system that is resistant to hijacking and censorship. Namecoin can be used as a decentralized method of authenticating TLS certificates, without relying on public certificate authorities. This eliminates the risk of compromised certificate authorities facilitating MITM attacks, as well as the risk of authorities refusing to issue certificates for specific websites in order to censor them. This project aims to improve the security, usability, and code quality of the TLS use case of Namecoin.

The Namecoin Project — **Visit https://NLnet.nl/project/Namecoin-TLS**

Internet Hardening Fund

Ripple

NGI ZERO

When you start up your computer, you will probably think twice before you download some random piece of software from the internet and run it. You know that doing so could allow unwelcome guests to your computer and your data. Your computer might even end up in a bot net. So when you see some nice piece of software, you will ask yourself the question: can I really trust the software? Perhaps you will check the origin it comes from. Better safe than sorry.

Did you miss checking something, though? What about the software that is already on your computer before you started? A computer is not of much use without an operating system. While most computers are sold with an operating system, actually you have the choice to remove that and install something different. Have you thought about the trustworthiness of that fundamental piece of software - your most fundamental travel companion on the wild west of the internet? Trustworthiness is essential. When an operating system has a so called 'back door' (either intentionally or not), someone could extract whatever user data - like personal pictures or home movies - from your computer. And the worse thing: without you ever finding out. The operating system guards all the other software, and warns you when you install software from the internet. But itself, it doesn't have to ask for permission. Ever. It doesn't just have "access all areas": in fact, it runs the whole show.

With commercial software like Microsoft Windows or Mac OS X that you get delivered when you buy a computer, trust in what their closed operating system does will of course always be a leap of faith: as a user you essentially are given no choice. In proprietary systems you do not have the freedom to study the source code, or to control what really happens. So you either trust the vendor, or you'd better not use it. For an increasing amount of people, after the revelations from whistleblowers like Edward Snowden, that "leap of faith" is not so obvious anymore. They prefer to use free and open source operating systems like GNU Linux, FreeBSD and OpenBSD. These are technology commons: the people that wrote the software allow you to inspect the source code. Even more so, they give you the source code to do anything with it that you like. So you don't just blindly have to take their word for it and trust them, you can take matters into your own hands.

One step beyond transparent source code is transparent running code. After all, most software is distributed pre-compiled with no method to confirm whether the binary code you have installed on your system is actually identical to the thoroughly vetted source code. To promote such reproducible code, Ripple helps developers and users transparently and incrementally build programs, without relying on any particular tool or ecosystem.

Technical description

As it stands, reproducible builds are not accessible to the average developer. Existing projects tackling this problem come with significant caveats: some rebuild packages from scratch, making them practically useless for interactive development, while discouraging users from hacking on the core parts of their system due to cascading rebuilds; others are drastically more efficient, but come with fewer correctness guarantees, and require build scripts to be re-implemented in custom DSLs, making them costly to adopt. This is further exacerbated by frustrating, flaky tooling, and the proliferation of compatibility issues arising from inherent constraints of these solutions. Ripple is a hermetic, incremental, meta build system. It provides stronger purity guarantees and improved efficiency over existing solutions, while being completely ecosystem-agnostic. In effect, Ripple can memoize arbitrary programs. This lets users migrate gradually, opting into ecosystem-specific optimizations and abstractions at their own pace, and opens up a huge number of creative possibilities. Ripple aims to make reproducible builds not only easy, but fun — encouraging mainstream adoption, so we might together put to rest the ghost of bygone builds.

Visit https://NLnet.nl/project/Ripple

NGI0 PET **Packaging** **Reproducibility**

U L X 3 M

NGI ZERO

Consumers and businesses overpay for computer hardware, because the market is not working well. When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component,

the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

Fortunately there are efforts underway to make hardware that, like open source software, is free to be reimagined and reassembled without restriction. Hardware that is transparently created, from the design up to the actual physical creation. As these projects grow and connect, they can lay the foundations for a technological commons of trustworthy hardware that is accessible for everyone to learn from and build upon.

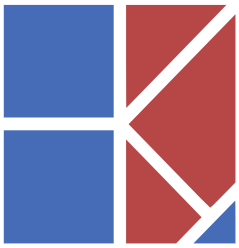
This project develops development tools to make it easier to work with field-programmable gate arrays, or FPGA's. FPGA's are chips that can be customized for a specific task ('programmable in the field') for image processing in digital cameras, portable electronics in smartphones and tablets, networking in 'harsh' industrial environments. Unlike a generic chip, an FPGA chip can be restricted in what it does - meaning that it can be made more secure while still using less energy. In this project, the ULX3S team will develop an affordable "vendor neutral" development board using open source tools, to stimulate bottom-up innovation, research and study of this flexible type of computing. Through a modular approach the same board can be used with different FPGA vendors daughter boards, lowering cost and making better use of natural resources.

Technical description

Embedded systems are everywhere, including in trusted environments. But what is really inside them? ULX3M is a modular version of the popular open hardware project ULX3S. ULX3M delivers a versatile programmable (FPGA) modular mainboard that can be used a wide choice of peripherals. The main board is "vendor neutral" and can be used with different FPGA vendors daughter boards. As the community continues to grow, lots of FPGA modules are written, and one goal of our boards would be that we can easily switch and check other vendor chips, and work more on vendor neutral code where possible. The project also improves SERDES availability. Some cheaper FPGA chips do not have lots of SERDES lines and when someone makes a board it needs to choose what peripheral will be using those SERDES lines. A daughter board that can be rotated in any position will allow more flexible usage. In that way, cheaper FPGA could be used to write all the code. With an open source design, users are not dependent on anyone to make boards and can run independent production.

RadionaOrg — **Visit https://NLnet.nl/project/ULX3M**

NGI0 PET DevelopmentBoard FPGA Foundation OpenHardware



When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component, the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

To break through this standstill, developer communities are working hard to deliver open, trustworthy and accessible alternative computer hardware that anyone can use, study, modify and distribute, just like they can with open source software. This project adds an essential tool to the tool kit for open hardware development, one of the first a developer would use: design and simulation of a chip mosaic. Using the latest technologies and an accessible user interface, Mosaic solves yet another part of the puzzle of open hardware design.

Technical description

Today, the chip design industry is deeply proprietary with NDAs at every level, which means it is not possible to share design files at all, which in turn stifles innovation and transparency in chip design. In order to create a chip design industry that can be trusted with our digital lives, and is accessible to educational institutions and small business, it is essential to develop powerful open source tools for chip design, which can be used by anyone and allows unhindered collaboration. Mosaic is a tool that attacks the first design phase of an analog chip, or analog peripherals for a digital one: design and simulation of the schematic. It will also interact with other phases of the design as needed. Unlike existing open source solutions it will be catered towards chip design, based on modern technologies, and extensive UX design.

Wishful Coding — **Visit https://NLnet.nl/project/Mosaic**

NGI0 PET **CircuitDesign** **Simulation**

When you go to a store to buy a laptop or mobile phone, you may see different brands on the outside but choice in terms of what is inside the box (in particular the most expensive component, the processor technology) is pretty much limited to the same core technologies and large vendors that have been in the market for decades. This has a much bigger effect on the users than just the hefty price tag of the hardware, because the technologies at that level impact all other technologies and insecurity at that level break security across the board.

In the field of software, open source has already become the default option in the market for any new setup. In hardware, the situation is different. Users - even very big users such as governments - have very little control over the actual hardware security of the technology they critically depend on every day. Security experts continue to uncover major security issues, and users are rightly concerned about the security of their private data as well as the continuity of their operations. But in a locked-down market there is little anyone can do, because the lack of alternatives. European companies are locked out of the possibility to contribute solutions and start new businesses that can change the status quo.

To break through this standstill, developer communities are working hard to deliver open, trustworthy and accessible alternative computer hardware that anyone can use, study, modify and distribute, just like they can with open source software. Unfortunately restraining and expensive design tools prevent these communities from solving important problems in their designs. This project aims to provide an open source alternative tool for placing and routing, an essential stage in making chips and boards that can make or break its performance, sustainability and usefulness. This freely available and transparent alternative will work seamlessly with other tools, which all together create a toolkit that can advance innovation on equal footing and ultimately improve the security, reliability and trustworthiness of the many devices we use everyday.

Technical description

Making a custom chip (ASIC) requires a vast ecosystem of expensive commercial tools, limiting the application of ASICs to large companies; this greatly hampers innovation. Project Luna aims to mitigate this situation by providing a robust open-source automated place & route tool, which forms an important but mostly missing part of the ASIC design flow. This way, universities, makers, small companies and start-ups can get access to ASIC design tools. Luna targets ASIC processes larger than 100nm, which makes it ideal for designing mixed-signal (analogue + digital) chips used in sensors and IOT devices. It integrates well with existing open-source tools, such as YosysHQ's Yosys (a logic synthesis tool) and KLayout (a manual ASIC layout tool), and commercial tools via industry standard file formats. In addition to the affordability issue, Luna allows a full-circle chain-of-trust to be established between designer and chip manufacturer because of its fully open-source nature. During its development, Luna will be used to manufacture designs via our industrial partners in order to verify the correctness and usability of the software. The goal is to present a minimal viable product consisting of a GUI, working place & route and timing verification.

Moseley Instruments — **Visit https://NLnet.nl/project/Luna**

NGI0 PET ASIC PnR Synthesis

l i b s p n g

Computer security for many people is a matter of trust, blind faith even. As we use the internet for basically everything and our devices and networks become increasingly complex, it takes more time and effort to understand and verify each layer of technology (even more so for devices that are glued together and software that is hidden behind restrictive licenses). And because new solutions are built on top of existing legacy systems, we continue to rely on technology that does not always meet today's needs for security and privacy any longer.

Building a future-proof internet does not only require totally new and outrageous ideas, but also fixing persistent problems and outdated parts: you can only build a fancy new house on a strong foundation. This project aims to provide an alternative for a widely used component that handles one of the most common (open) image formats, PNG. To prevent errors in handling images and security vulnerabilities, an alternative component will be delivered that can easily be tested and verified for correctness. This helps website technology and applications used all over the world function a little bit safer.

Technical description

libspng is a platform-independent C library for handling IETF's Portable Network Graphics (PNG) images. The goal of this project is to provide a robust and fast library with an easy to use API. It is designed to be a modern alternative to the reference implementation, written from scratch using secure coding standards. It comes with an extensive test suite and is fuzz tested, it is also fastest decoder overall. The NGI Zero grant will be used to develop complete PNG write support, architecture-specific performance optimizations, including improvements to testing, decoding and documentation.

Visit https://NLnet.nl/project/libspng

NGI0 PET ArchitecturePortability FuzzTesting Library PNG

Annex 2: Presentations, contributions and initiatives in 2020

NLnet and its employees actively participate in various fora and projects regarding the open and free internet, cybersecurity, and the implementation of open standards and open source. Due to the pandemic most of these events were limited to digital gatherings. A selection of the most prominent contributions:

- FOSDEM, February 1-2 2020
- Session Digitalisering van de Samenleving, April 2 2020
- NGI Forum, May 18-19 2020
- 2nd International Open Search Symposium , May 26 2020
- EuroDIG, June 10-12 2020
- Secure Messaging Summit, 3-4 August 2020
- NGI Policy Summit, 28-29 September 2020
- ICT Verticals and Horizontals: Fintech, Digital Assets and Smart Grids, 11 November 2020
- Guix Conference, 22 November 2020
- International Conference on Computer Operating Systems and Technologies (Iccost), December 16-17 2020

'Radically Open Security' (ROS) is a company around ethical hacking and security founded in 2014 by dr. Melanie Rieback. ROS has been donating over 90% of its proceeds to NLnet foundation, and this year reached the magical threshold of half a million euro of cumulative donations. The company takes a principled approach which puts transparency, open source, responsible disclosure and ethics first – which together with its idealistic and non-hierarchical model has attracted a talent pool of ethical hackers. In 2020 the portfolio and public profile continued to grow, for example with ROS **uncovering privacy and security issues in the Dutch contact tracing app CoronaMelder**. ROS is also part of Reviewfacility.eu (see elsewhere in this document) and the NGI Zero coalition.

In 2020 NLnet prolonged its membership of Digital Infrastructure Netherlands. DINL is a group of seven institutes, associations and foundations (SIDN, DHPA, DDA, AMS-IX, ISPConnect, Surfnet, Nederland-ICT, VVR and NLnet) that collectively works on important topics in the Netherlands Digital Infrastructure community: promotion, education, cybersecurity, and laws & policy.

NLnet supports the Open Invention Network. Organisations and non-formal organisations like FLOSS communities benefit from the defensive patent pool and from the collective legal support to shield themselves and their users against patent offenses. Open Invention Network has made several donations to NLnet in recognition of its contribution to this initiative.