Trust from the ground up

Highlights from NGI Assure 2020-2024

NGI ASSURE

Preface

This year, the internet has turned no less than half a century old. In its earliest days, of course it was only a handful of people that had access to the technology: internet only started reaching a global audience at scale far more recently, in the late nineties. At that time, for many people the internet was an unprecedented instrument of hope and universal progress. The great promise of the internet was that it was a user-driven decentralized "network of networks" — driven by "open standards". Most of us probably don't recognise this today, but back then this label ("open standard") was meant to distinguish from the top-down, classical standardisation efforts that dominated elsewhere — and which were often very political in nature. The more engineer-driven community approach of the internet did not rely on some central authority to coordinate it all, but on "rough consensus and running code". Anyone could join a mailing list and have their voice heard. Anyone can put up a web page. This open character allowed for explosive growth and rapid iteration of ideas and code. The end-to-end principle would ensure user choice and control, universal access and permissionless innovation.

When looking at the internet we have today, reality has caught up with utopia. The revelations of whistleblower Edward Snowden about large scale mass surveillance left the world speechless. We have witnessed a stark centralization and consolidation of power in a small number of profit-driven actors. Instead of communication and information technologies empowering citizens, we are often left feeling powerless, with technological innovation happening mostly to us rather than for us.

But of course the internet doesn't *have* to be that way. History may have pushed the internet in a certain direction, but there is no fundamental reason why it can not be an instrument of hope and universal progress again.

To reignite the vision of an internet as a digital commons, the European Commission conceived the **Next Generation Internet** initiative in 2016. While the initial ambition was still fairly modest, the initiative quickly picked up momentum and found its course. By investing in free and open source technologies, NGI served a global research and development community that was in dire need for such support.

NGI Assure was part of the second wave of NGI programmes. It was originally perceived to be much more narrow in technological scope, centered around the European Commission's erstwhile blockchain ambitions — which in the end didn't turn out to be that fertile a path in terms of societal outcome. Due to the resounding success of the first wave of NGI programmes, the new programme was allowed to pivot from this very limited solution space — "a solution looking for a problem" — and instead address the much more *comprehensive* and *practical* vision of the Next Generation Internet initiative.

The NGI Assure programme distilled the essence of why the European Commission was so enamoured with blockchain solutions: to **anchor** and **simplify** establishing trust, and to make the usage of remote resources on the internet inherently more **trustworthy** and **secure**. Or as it was succinctly phrased in the call texts: *"to design and engineer reusable building blocks for the Next Generation Internet as part of a complete, strong chain of assurances for all stakeholders regarding the source and integrity of identities, identifiers, data, cyber-physical systems, service components and processes"*. NGI Assure set out to address these problems with the complete set of potential solutions, and with a much higher ambition than probably most would expect: to create **reproducible** and **trustworthy** end-to-end solutions that not only can withstand the hostile battle grounds of the modern internet, but even make it **user-friendly**.

While originally the programme was supposed to last three years, due to both the disruptive effect of the SARS-CoV-2 pandemic and the large amount of projects, the project was prolonged for another year. Now, as NGI Assure is finally nearing its official end, we can look back to all that was achieved. We believe this is quite impressive. In total the programme harboured 152 projects, involving no less than 365 different teams from 42 countries.

A great diversity of technologies was tackled: from *post-quantum cryptography* to *productivity tools*, from securing the *software supply chain* to *digital signatures*, from *EDA* design tools to *E2EE instant messaging*, and from *videoconferencing* to advanced solutions built around *zero knowledge*, standardising *object capabilities* and *differential privacy*. The unifying factor: all of these projects worked on digital commons that improve our digital sovereignty, empower end users, increase systemic resilience and provide transparency, choice and self-determination. The result wasn't just software that people can now run to improve their lives, but also new standards, protocols and open hardware to use and reuse as building blocks.

Each of these efforts contributed to a **more trustworthy**, **resilient** and **sustainably open** internet in its own unique way. We are very grateful for the amazing work done by all these people. By helping redecentralise the internet, they have brought a better internet a few steps closer. To celebrate the vast effort made by all those involved in the NGI Assure program we have brought all of these projects together in a single booklet. And we even printed it on paper as a tangible homage, because no matter how cool the web is: paper may outlive it all.

Let's celebrate all those working on an internet for all, also in the wider free and open source community. And a special thanks to the team at DG Connect at the European Commission for their support. We are thankful as well to the amazing teams at *Funding Box*, *Innovation Engineering* and *NLnet foundation* that worked tirelessly to make it happen. Finally, our sincere gratitude to the members of the external review committee that kept a close watch on the eligibility of the projects. You all made a difference.

Team NGI Assure

Table of Contents

Aerogramme

Standards-compliant open-source IMAP server with server-side encryption

Description

Aerogramme is an open-source IMAP server targeted at distributed infrastructures and written in Rust. It is built on top of Garage, a (geographically) distributed object storage software. Aerogramme thus inherits Garage resiliency: its mailboxes are spread on multiple distant regions, regions can go offline while keeping mailboxes available, storage nodes can be added or removed on the fly, etc. Not only does it inherit its resiliency, but it also shares the burden of data management. Aerogramme can be seen as a proxy between the IMAP protocol and Garage protocols (S3 and K2V); it does not handle any data on its own and can be freely moved between machines. Multiple instances can also be run in parallel. As emails are very sensitive, Aerogramme encrypts users' mailboxes with their passwords. Data is decrypted in RAM upon user login: the Garage storage layer handles only encrypted blobs. Aerogramme is to our knowledge the first IMAP server to be designed from the ground up with object storage in mind. Thanks to this design, it is resilient and easy to scale.

Email **IMAP**

More info: https://nlnet.nl/project/Aerogramme

Website: aerogramme.deuxfleurs.fr
Repository: git.deuxfleurs.fr/Deuxfleurs/aerogramme

Ari

Purely functional programming language designed to 'type' binary files

Description

Ari is an early research project designed to make binary files more accessible. It's a purely functional programming language and library intended to act as foundation for building developer tools that can manipulate arbitrary binary files. It can be used as a basis for building a structural binary differ, or a tree-based editor for directly editing binary files.

It aims to reach this goal by tackling the biggest obstacle with binary data: the need for implicit format-specific knowledge to understand how binary files are structured. Over time, we'll build up a repository of file formats encoded in Ari (called " Ari types"), which can then be used to compile a " type radix tree" from any given set of Ari types. This " type radix tree" will be used as an efficient way to interpret a single file as multiple formats at once, while trimming out invalid interpretations along the way of parsing.

Ari fundamentally differs from existing approaches like Kaitai Struct, GNU poke, and even parser generator tools like Tree-sitter in that it's heavily based around the combination of algebraic type theory & set theory and sits in-between a data specification language that doesn't have support for functions, and a fully Turing complete language that has no guarantee of halting. The plan is to work together with these other projects as they each have their own unique approach that Ari isn't focused on, whereas Ari is more of a research project intended to explore what's possible.

BinaryAnalysis **StaticTyping**

More info: https://nlnet.nl/project/Ari

Website: gitlab.com/ari-lang/ari

Atomic Data

Typesafe handling of LinkedData

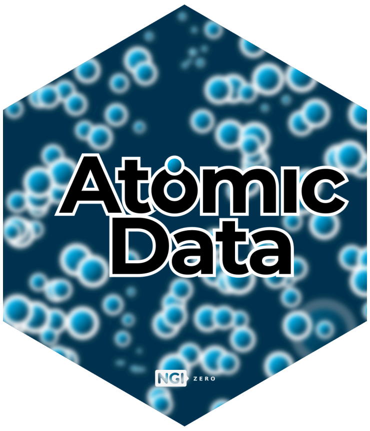
Description

Atomic Data is a modular specification for sharing, modifying and modeling graph data. It uses links to connect pieces of data, and therefore makes it easier to connect datasets to each other - even when these datasets exist on separate machines. Atomic Data is especially suitable for knowledge graphs, distributed datasets, semantic data, p2p applications, decentralized apps and linked open data. It is designed to be highly extensible, easy to use, and to make the process of domain specific standardization as simple as possible. It is type-safe linked data (a strict subset of RDF), which is also fully compatible with regular JSON. In this project, we'll work on the MIT licensed atomic-server and atomic-data-browser, which are a graph database server and a modular web-gui that enable users to model, share and edit atomic data. We'll add functionality, improve stability and testing, improve documentation and create materials that help developers to get started.

KnowledgeGraph **RDF** **SemanticWeb**

More info: https://nlnet.nl/project/AtomicData

Website: atomicdata.dev
Repository: github.com/atomicdata-dev/atomic-server



Authenticated DNSSEC bootstrapping

Secure in-band announcements of DNSSEC parameters

Description

Turning on DNSSEC for a domain involves (1) signing the domain's DNS zone content and (2) adding the signature public key to the chain of trust. The second step has long posed a problem, as it requires (often manual) transfer of information from the domain's operator to the parent (usually the top-level domain). It is largely due to this " DNSSEC bootstrapping problem" that only about 6% of the Top 1M domains are securely delegated (Tranco, 06/2022).

The project extends commonly used authoritative nameserver software with native support for authenticated DNSSEC bootstrapping (draft-ietf-dnsop-dnssec-bootstrapping,). This protocol, meanwhile published as RFC 9615 by IETF, allows DNSSEC parameters to be communicated automatically and securely, enabling DNS operators and parent registries to turn on DNSSEC automatically. To measure the protocol's impact on real-world DNSSEC deployment, measurements of protocol adoption over time will be made available.

Cryptography DNSSEC Deployment IETF Standardisation

More info: https://nlnet.nl/project/AuthenticatedDNSSECbootstrap

Website: datatracker.ietf.org/doc/draft-ietf-dnsop-dnssec-bootstrapping
Repository: github.com/desec-io/desec-ns

Heads-OpenPGP

OpenPGP Authenticated Heads and long-time awaited security improvements ●

Description

The work to be accomplished in this project will resolve Heads current missing accessibility, reproducibility and platforms locking improvements, including Heads missing authentication mechanisms prior of permitting recovery shell access or booting USB external media, possibly leading to data loss without evil-maid even having to unscrew anything. Also, a user currently losing his USB OpenPGP dongle would lose its private encryption subkey forever therefore losing access to all past encrypted content and lessening security until dongle replacement.

By considering Heads as a secure pre-boot " clean room" environment on initial flashing/reflashing of whole firmware, generating OpenPGP master key and subkeys in memory and implementing keys backup/restore mechanisms to/from/creating USB thumb drive encrypted storage, Heads will be able to rely further on OpenPGP (gnupg toolstack) and its detached-signing of content and signature verification against fused public (measured) key to authenticate the owner of the machine prior of letting him have access to the machine's persistent states. Having reproducible builds again will make auditability of the firmware easier, while locking the firmware prior of leaving Heads environment will prevent whole classes of SPI based persistent threats.

Bootloader **OpenPGP**

More info: https://nlnet.nl/project/AuthenticatedHeads

Website: osresearch.net
Repository: github.com/osresearch/heads

Heads-OpenPGP

What does this project mean to users?

Heads already strongly relies on OpenPGP to measure and verify /boot digest integrity content on top of a firmware measured attested state. It also already permit to generate OpenPGP keypairs inside of OpenPGP smartcards, so the private keys never leave the security dongle where they were generated. But if Heads authentication is enforced right now and that dongle is lost, the end user would be locked out of his machine, would need to operate it in a much unsafe way waiting for replacement dongle would require new keypair generation on received dongle. The end user would have to choose between booting an outdated Xen/Kernel/initrd detached signed/verified /boot digest, or, upgrade the OS system core binaries without having Heads able to detach-sign/verify /boot digest.

The funding will also permit to resolve the current reproducibility issues by replacing the current Make based buildsystem with linux-builder on top of Guix/Nix installable layer on any linux based distribution, which will pin all poked system binaries/libraries dependencies on top of any Linux distribution/docker image. Funding will also permit platform locking logic to prevent write access to SPI from the final kexec'ed OS, will finally make Heads work with different configurable keyboards language maps, will extend QEmu testing possibilities with software TPM (swtpm) and encrypted OpenPGP keypair thumb drive (raw disk), will include TPM2 toolchain, ease future boards integration, further Heads adoption from coreboot hardware distributors while revamping documentation so it reflects its evolved state.

Bertie

Formally verified TLS 1.3 implementation

Description

The security of the Web ecosystem relies crucially on Transport Layer Security (TLS) protocol, but despite years of study, cryptographic weaknesses and implementation bugs in TLS implementations continue to be found on a regular basis. Bertie is a high-assurance TLS 1.3 implementation written in a subset of Rust called hacspec. Bertie uses the formally verified HACL* cryptographic library and its protocol code can be verified using the F* framework. Hence, it offers strong guarantees from the crypto layer up to the protocol API. The funding from NLnet will be used to stabilise Bertie, add documentation and tests, improve its performance, maintain its proofs, and set it up as an open source project with best practices and long-term software support.

FormalVerification TLS

More info: https://nlnet.nl/project/Bertie

Website: cryspen.com
Repository: github.com/cryspen/bertie

Blink Qt Messaging

Add modern encryption to SIP softphone

Description

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. This project will extend its capability to support end-to-end asynchronous messaging and end-to-end encription that works both online (OTR) and offline (OpenPGP). Additional features to be developed include end-to-end delivery and read notifications, and a searchable history database.

E2EE InstantMessaging OTR OpenPGP SIP Softphone

More info: https://nlnet.nl/project/Blink-OTR-OpenPGP

Website: icanblink.com
Repository: http://devel.ag-projects.com/repositories/blink-qt

Briar Desktop

E2EE online and offline messaging and discussion

Description

Briar Desktop is a client for the peer to peer messenger Briar that runs on the typical desktop operating systems Windows, macOS and Linux. With the emergence of multiple Linux-based operating systems for phones, it will also become possible to adapt it to run on operating systems such as Manjaro, PureOS and postmarketOS. A basic version of Briar Desktop has just been implemented and released to the public, but its features are still limited to one-to-one communication. The main goal of this project is to implement the additional group-oriented modes of communication that Briar's Android client supports: groups, forums and blogs. While the first iteration of development focused on Linux, publishing for macOS and Windows are going to be stabilized from experimental to production stage within this project. To keep up with the development of the Android client, support for the upcoming Mailbox feature is also going to be implemented.

Cryptography **InstantMessaging**

More info: https://nlnet.nl/project/Briar-beyond-Android

Website: briarproject.org
Repository: code.briarproject.org/briar/briar-desktop

Cable

A new wire protocol for cabal (and beyond)

Description

Distributed systems development is hard. Doubly so when you have adopted a complicated technological stack in order to achieve the goals of a peer-to-peer group chat like Cabal. Some problems inherent in an approach can only be seen in hindsight, and repaired with foresight. Enter Cable, a new lightweight binary communication protocol originally specified to be the upcoming backbone of the peer-to-peer group chat Cabal.

The Cable protocol is pull-based, with message authenticity through cryptographic hashes, where peers receive messages by sending queries into the network: " give me the most recent week of chat messages in channel main" . Peer-to-peer query-forwarding is built into the design to enable message retrieval outside any given peer's direct connections. Its logless approach enables message deletion and allows the many devices owned by a single person to use the same cryptographic identity in communication. The binary specification combined with the pull-based design minimizes system resources in transport and storage alike. Cable's goals as a protocol: to be compact over the wire, easy to implement from scratch with libsodium bindings as the only dependency, to enable bridging across any network transport, and to be agnostic with regard to how data is stored. In addition to unlocking new capabilities in Cabal's future, we also hope to pave the way for a multitude of other protocols to be hosted on Cable's agnostic wire format.

InstantMessaging **P2P** **Protocol**

More info: https://nlnet.nl/project/Cable

Website: cabal.chat
Repository: github.com/cabal-club/cable

Libre-SOC Cavatools: Power ISA Simulator

Power ISA Simulator

Description

Cavatools is a high performance ISA simulator, similar to qemu. However unlike qemu, cavatools is designed with two goals in mind: to provide accurate guidance on instruction effectiveness, and to run at close to real-time performance on multi-core host systems.

The only hardware that cavatools currently supports is cycle-accurate emulation of RISC-V: this Grant is intended to add not only the Power ISA but also add the Draft SVP64 Cray-style Vector Extensions being developed by Libre-SOC (and sponsored by NLnet). Other work includes being able to verify and compare multiple independent implementations, running the same program, to check interoperability, whether in emulators, hardware simulations, simulators or actual ASICs.

Emulation **OpenHardware** **OpenPower**

More info: https://nlnet.nl/project/Cavatools

Website: libre-soc.org
Repository: git.libre-soc.org

Choreographic Programming: From Theory To Practice

Generating a standard library of core distributed algorithms with formal proofs

Description

To safely leverage the next-generation internet for mission-critical apps, it is crucial to assure that communications among distributed processes are deadlock-free (i.e., processes never get stuck waiting for a message that will never be sent) and behaviourally-compliant (i.e., processes never send messages that violate the intended application-level protocols). Choreographic programming is a promising new method to build distributed systems that assures the absence of deadlock and compliant behaviour by construction (vs. testing, which is notoriously difficult in the presence of concurrency and distribution). The aim of this project is to take advantage of recent scientific progress in programming language theory for distributed systems, and develop a new choreographic programming language (Klor) as an embedded DSL in Clojure, including a standard library of core distributed algorithms.

ChoreographicProgramming DSL FormalProof

More info: https://nlnet.nl/project/ChoreographicProgramming

Website: github.com/lovrosdu/klor

Coko Docs

A modern, open source replacement for Google Docs and Drive

Description

Coko Docs is an open source solution for storing and editing documents using Coko's publishing technologies. It is the first part of an Open Suite, which will be integrated with professional Open Publishing products. Coko Docs will have a modern collaborative environment for creating, sharing and hosting files in various formats. We aim to build inclusive tools as powerful as Google Drive and Docs, our initial target audience ranges from individuals to small organisations. Our primary goal is an Open Source product with strong Privacy and Security protocols and elegant accessible design. We will utilize the NLnet funding for the first phase of development where we are adding collaborative editing to the integrated document editor, with offline support (for low-bandwidth scenario's).

CollaborativeEditing **Office** **Productivity**

More info: https://nlnet.nl/project/CokoDocs

Website: coko.foundation/community
Repository: gitlab.coko.foundation/cokoapps/cokodocs

Conversations 3.0

Secure and standards-compliant XMPP client for Android

Description

Conversations – a popular XMPP instant messaging client for Android – has been around since 2014. Since then not only have Android development best practices changed but also user requirements on the app have shifted dramatically. Features like emoji reactions, quotations (references), edit history or simply multiple images per message weren't on the developers mind in 2014 and are difficult or impossible to implement with the current software architecture. Conversations 3.0 is an architecture overhaul that adapts Conversations to a modern Android development style (namely Android Jetpack) and also redesigns the database to accommodate the aforementioned features. The well-functioning XMPP layer will remain intact during this refactoring in order to keep all existing features and not re-introduce bugs that have been fixed ages ago.

InstantMessaging **MobileApp** **XMPP**

More info: https://nlnet.nl/project/Conversations-3.0

Website: conversations.im
Repository: github.com/siacs/Conversations

CryptoLyzer

Cryptographic settings analyzer library

Description

CryptoLyzer is a cybersecurity tool that can analyze the cryptography-related settings of clients and servers in the case of several different protocols. The tool's primary purpose is to support end users as well as system administrators, security engineers, auditors, etc., in their work by telling them the details of the currently applied setting and informing them about the potential weaknesses and vulnerabilities.

Unlike many other notable free software projects that focus on just one protocol family, CryptoLyzer wants to be as comprehensive as possible. On the one hand, users can analyze several cryptographic mechanisms (e.g., SSH, HTTP security headers, JA3 tag, and later OpenVPN), not just the most popular TLS protocol. On the other hand, it is possible to test both the standard and special or corner cases. Latter means the tool can test hardly supported, experimental, obsoleted, or even deprecated mechanisms or algorithms, which may carry significant risks. The project intends to learn from the existing projects and integrate their solutions to lower the barrier to good cryptographic settings making communication on private and public networks more secure.

ProtocolCheck

More info: **https://nlnet.nl/project/CryptoLyzer**

Website: gitlab.com/coroner/cryptolyzer

Reinstatement of crypto.signText()

Cryptographic signatures brought back to the browser

Description

Since the 1990s Netscape and Firefox supported the ability to sign an arbitrary piece of text with a digital certificate, and have that signature returned to the webserver. The texts being signed have historically ranged from transaction records, financial declarations, and court documents. This project implements a set of Native Browser Web Extensions that bring the digital signing of text to all modern browsers that support the NMBE standard. The process of choosing the certificates and generating the signatures is performed outside of the browser, using APIs native to each operating system. Web pages communicate with the extensions using the Javascript crypto.signText() function, and the signed documents are returned packaged as a PKCS7 response. The project aims to make digital signing accessible, while being browser agnostic.

More info: https://nlnet.nl/project/crypto.signText

Website: redwax.eu/rst
Repository: source.redwax.eu/projects/RST

CryptPad Auth

Implement external identity mechanisms to E2EE collaborative editor

Description

CryptPad is a real-time collaboration environment that encrypts all user-generated content in users' browsers, making it illegible to the host of the service. In this project we'll develop optional extensions to the platform to provide additional layers of protection for such data by pursuing two broad strategies in parallel. For the first, we'll take a top-down approach to security through integration with identity provider services like LDAP or SSO, allowing organizations to apply centrally managed access control policies. For the second, more bottom-up approach, we'll offer tighter control of user accounts through various secondary authentication methods like app-based TOTP or email " magic-links" . These new features will provide more choices for the protection of data stored in CryptPad, while also making the platform more approachable for conventional organizations by leveraging their existing points of trusted infrastructure.

CollaborativeEditing MultiFactorAuthentication

More info: https://nlnet.nl/project/CryptPad-Auth

Website: cryptpad.org
Repository: github.com/xwiki-labs/cryptpad

CryptPad Auth Improvements

Better user management, 2FA and SSO for CryptPad

Description

CryptPad is a secure and encrypted open-source collaboration suite, allowing people to work together in real-time on presentations, texts and spreadsheets as well as conduct polls or gather data through forms. And unlike traditional cloud offerings, the server does not get to learn what its users are working on: all the data is encrypted on the devices of the users, before it is sent to the server. The project already offers advanced features like 2FA and Single Sign On (OIDC and SAML), making it easy to smoothly integrate the tool into corporate environments.

The goal of this project is to perform user interface improvements to the 2FA and SSO system. It will also build a User Directory that will allow to manage users and also list users according to information that would be available about them in case of login through SSO or Invitation. It will also build towards enabling advanced usage scenario's without SSO, instead offering for instance the possibility to send registration invitations to users in a way that doesn't break the security model of Cryptpad.

2FA **CollaborativeEditing** **SSO**

More info: **https://nlnet.nl/project/Cryptpad-Directory**

Website: cryptpad.org
Repository: github.com/xwiki-labs/cryptpad

CryptPad Quality Test Suite

Continuous testing of critical CryptPad functionality

Description

Cryptpad is an open-source, end-to-end encrypted online collaboration platform featuring a number of different services like a code editor, spreadsheet, polls and Kanban boards. Unlike with other office suites, the server learns nothing about the contents of what is being collaborated on.

As the project continues to gain traction with users and developers, and various integrations with the platform are taking place, there is an obvious need to make sure development in one place doesn't inadvertently break something somewhere for others.

With the software now widely deployed and in active use by many people and organisations, a more structured approach to testing core accessibility of the platform through CI is necessary. This will ensure that Cryptpad remains available to serve users as long as they need it.

CollaborativeEditing **Testing**

More info: https://nlnet.nl/project/CryptPad-QA

Website: cryptpad.org
Repository: github.com/xwiki-labs/cryptpad

CryptPad WCAG

Accessibility improvements to CryptPad suite

Description

CryptPad is an end-to-end encrypted collaboration suite that is fully open-source. It is used by people around the world to work together on shared documents and spreadsheets in real-time, to conduct private polls, and many other use cases.

A significant effort has always been made to make sure that the software is fully usable with assistive technologies. As a very active project which is continuously in development, this is of course a moving target. The goal of this specific project is to remove the last remaining hurdles that prevent people with disabilities from using the entire feature set of Cryptpad. The ultimate ambition of Cryptpad is to become officially W3C WCAG certified, and serve the widest possible community of users.

CollaborativeEditing **WCAG**

More info: https://nlnet.nl/project/CryptPad-WCAG

Website: cryptpad.org
Repository: github.com/xwiki-labs/cryptpad

Converged Security Suite Improvements

Open source tooling for BIOS configuration

Description

The Converged Security Suite has been developed as an open-source tool to provision and test systems where proprietary (and closed) Intel Security Technologies - such as " Trusted Execution Environment" , " BootGuard" , and " Converged BootGuard and TXT" (CBnT) - are enabled. Since this is a security-critical operation, transparent open-source tooling is needed to securely provision and test the configuration of your system within the limitations of a closed system.

However, current configuration tools are not available for technical scrutiny and only available under NDA. The same applies to test suites that validate the system and its configuration.The Converged Security Suite tries to change this by implementing an open alternative for those tools. Within this project, the team will implement Bootguard (provisioning and test suite) and add CBnT test suite support.

Firmware

More info: https://nlnet.nl/project/CSS

Website: github.com/9elements/converged-security-suite

DATALISP

Universal data interchange format using canonical S-expressions ●

Description

As society moves digital the need for thorough fundamentals becomes more prominent. Datalisp is a laboratory for decentralized collaboration built on a few well understood ideas which imply a certain architecture. The central thesis of datalisp is: " If we agree to use a theoretically sound data interchange format then we will be able to efficiently express increasingly complicated coordination problems" , but in order to move the web to a different encoding we will need incentives on our side. A substantial improvement in user experience is needed and we aim to provide it. Ultimately our goal is to give peers the tools they need to protect themselves, and others, by collaboratively measuring the legitimacy of information and locally; by assessing whether data can be trusted as code or whether it requires user attention. Datalisp is the convergence point for all these tools (none of which is named " datalisp") rather than a language, join us in figuring out how to reach it!

Canonicalisation **DataPortability**

More info: https://nlnet.nl/project/DataLisp

Website: datalisp.is
Repository: sr.ht/~ilmu/tala.saman

Securing Internet protocols with DIDs

Bridge Decentralized Identifiers with standardised authorisation mechanisms ●

Description

Many Internet protocols require authentication, e.g. when we check our email account with a username and password, when we authenticate to SSH hosts with public keys, or when we log in to websites using OpenID Connect.

Decentralized Identifiers (DIDs) are a new type of identifier that have associated private keys and can be used for authentication purposes. DIDs are in practice mostly used for exchanging Verifiable Credentials (VCs) between Issuers, Holders, and Verifiers. However, on a more basic level, DIDs can also simply be used as a replacement for usernames/passwords or static public keys, to authenticate by proving control over one's DID. Unlike other identifiers such as usernames or domain names, DIDs do not require a central authority for creating and using them.

In this project, we will work on integrating DIDs with existing Internet protocols that require authentication by developing a new SASL mechanism. The idea is that for example you could log in to your SSH host, email account, IRC server, XMPP server, etc. using your DID, which can improve both usability and security.

DID IETF Library SASL SSI W3C

More info: https://nlnet.nl/project/DID-auth

Website: github.com/peacekeeper/did-based-sasl

Distributed Mechanism Learning

Privacy preserving ways of distributed data usage

Description

Mechanism design is a field concerned with finding rules for economic processes which incentivize self-interested agents to behave in a way, such that a common goal is reached. This project aims to build robust infrastructure for mechanism design via machine learning, to make theoretical results more applicable to practical networked deployments. We plan to do this by finding solutions for the following two problems and making them accessible to developers, while keeping the required domain knowledge to a minimum:

On the one hand, a trusted third party is often assumed to exist, which is supposed to learn and execute the mechanism. In practice, finding neutral trusted parties who do not stand to gain anything from cheating can be hard. To solve this problem, we distribute the computation of the trusted party over multiple computers, ideally controlled by different entities, using multiparty computation. This way, we get a more robust trust base with better alignment of incentives.

On the other hand, current models often assume prior knowledge about preference distributions of agents to learn optimal mechanisms. In practice, this knowledge is not always available. We exchange finding optimal solutions using prior information with finding approximate solutions using no prior information, by way of differentially private learning. This results in more general applicability, especially in settings with sparse information.

Auction **DifferentialPrivacy** **MultiPartyComputation**

More info: https://nlnet.nl/project/dist-mech-learn

Website: github.com/degregat/dist-mech-learn
Repository: github.com/dpsa-project

Python supply-chain with dream2nix

Towards a secure, extensible #sym.amp reproducible Python supply-chain with dream2nix

Description

We aim to improve the software supply chain of Python with Nix by extending Dream2nix. While the Nix build system offers great reproducibility and auditability features, the effort required to manual write build expressions for all transitive dependencies has lead to the creation of various " lang2nix" tools. Dream2nix is a collection of such tools and a library handling shared concerns, with existing implementations for NodeJS, Rust and Haskell. This project is going to implement first class Python support in dream2nix. Packagers and developers will be able to build standards-compliant projects with nix automatically, while still being able to transparently apply patches where necessary.

Nix **Python** **SoftwareSupplyChain**

More info: https://nlnet.nl/project/Dream2nix-Python

Website: nix-community.github.io/dream2nix
Repository: github.com/nix-community/dream2nix

dream2nix

Automate reproducible packaging for various language ecosystems

Description

Dream2nix is part of the overal effort to create more technical assurances, transparency and robustness within the software supply chain. Dream2nix as a framework allows more open source projects to achieve reproducible builds easier, and helps to create an auditable toolchain across different technical dependencies. The ability to reproduce software builds is of major importance when it comes to verifying if a given binary is the product of a given source code. Reproducibility also increases the maintainability and reliability of small and large software deployments. The nix build system allows for such reproducibility even for complex software systems. dream2nix integrates existing well known programming language specific package managers like npm, yarn or cargo with the nix build system, which will allow many open source projects to benefit from nix' unique properties.

Packaging **Reproducibility**

More info: https://nlnet.nl/project/Dream2nix

Website: nix-community.github.io/dream2nix
Repository: github.com/nix-community/dream2nix

Earthstar

P2P protocol and APIs for collaborative and social applications

Description

Your data is stuff you care about. But a lot of the time, you only get to interact with it in places owned by corporations. It's a bit like living in someone else's house. One consequence is that you don't get to choose who can see your stuff: malicious actors can follow your activities and harass you, and the owners of the space can record what you do and sell that information on. And because the space isn't yours, you don't get any say over how anything works: features you like can disappear overnight, and your data can be changed or deleted without your consent.

What if you and the people you care about could band together and have your own place for your data to live? Where the only people who see your stuff are people you trust, and no-one is selling your privacy? And where you decide how things works and when it should change?

Earthstar is a pocket-sized toolkit to help users build a place of their own. Easily create user-owned infrastructure that holds the data you care about, in formats which suit your needs, and write your own applications to interact with it — or use ones from the community!

CRDT P2P

More info: https://nlnet.nl/project/Earthstar

Website: earthstar-project.org
Repository: github.com/earthstar-project/earthstar

Earthstar (Encryption, Safety, and Local Sync)

Improve security, encryption and sync capabilities in Earthstar CRDT ●

Description

Storing and collaborating digital data is an essential part of every day computing, from photo-sharing amongst family members, to document co-authoring between colleagues. Earthstar is a tool for building undiscoverable, offline-first shared data storage. Users decide which devices their data are stored on, what the infrastructure of their network looks like, the shape of their data, and how they can interact with it. The proposed project adds a number of useful features, notably end-to-end encryption (including metadata), P2P discovery in local networks and efficient data synchronisation.

CRDT

More info: https://nlnet.nl/project/EarthstarEncryption

Website: earthstar-project.org
Repository: github.com/earthstar-project/earthstar

Earthstar (Encryption, Safety, and Local Sync)

What does this project mean to users?

Storing and collaborating digital data is an essential part of every day computing, from photo-sharing amongst family members, to document co-authoring between colleagues.

Almost all systems we use require us to trust corporations who pry into our data, sell on our personal details and add or remove functionality on a whim -- not always in the interest of users. In extreme cases, these companies can go out of business or be acquired and closed down, causing total data loss.

It is clear that this situation is untenable and must change. Many proposed solutions include pushing data onto blockchains. While blockchains might add a degree of data resilience, they also have significant implications. On a blockchain, data can never be edited or deleted and is visible to everyone. The nature of blockchains themselves reduce users to powerless nodes in a network without trust.

Earthstar is a tool for building undiscoverable, offline-first shared data storage. Users decide which devices their data are stored on, what the infrastructure of their network looks like, the shape of their data, and how they can interact with it.

Earthstar is powerful and flexible. It can be used by friends building a chatroom, researchers without internet access in the field, archivists seeking a resilient way to store data, and anyone else whose data and collaboration needs can no longer be served by the existing status quo.

Encoding for Robust Immutable Storage (ERIS)

Encrypted and content-addressable data blocks

Description

The Encoding for Robust Immutable Storage (ERIS) is an encoding of content into a set of uniformly sized, encrypted and content-addressed blocks as well as a short identifier (a URN). The content can be reassembled from the encrypted blocks only with this identifier (the read capability). ERIS is a form of content-addressing. The identifier of some encoded content depends on the content itself and is independent of the physical location of where the content is stored (unlike content addressed by URLs). This enables content to be replicated and cached, making systems relying on the content more robust.

Unlike other forms of content-addressing (e.g. IPFS), ERIS encrypts content into uniformly sized blocks for storage and transport. This allows peers without access to the read capability to transport and cache content without being able to read the content. ERIS is defined independent of any specific protocol or application and decouples content from transport and storage layers.

The project will release version 1.0.0 after handling feedback from security audit, provide implementations in popular languages to facilitate wider usage (e.g. C library, JS library on NPM), perform a number of core integrations into various transport and storage layers (e.g. GNUNet, HTTP, CoAP, S3), and deliver Block Storage Management (quotas, garbage collection and synchronization for caching peers).

ContentAddressing Cryptography Encoding

More info: https://nlnet.nl/project/ERIS

Website: eris.codeberg.page
Repository: codeberg.org/eris

Friendly Forge Format (F3)

Proposed Standard for secure communication between software forges

Description

The Friendly Forge Format (abbreviated F3) is an Open File Format for storing the information from a forge such as issues, pull/merge requests, milestones, release assets, etc. as well as the associated VCS (Git, Mercurial, etc.). F3 is designed to exchange the state of a software project between GitHub, GitLab, Gitea, etc. for backup, mirroring or federation. F3 is essential for a forge to provide key requirements. (i) Portability: the entire state of a software project can be dumped and restored at a later time, on a different development environment (ii) Versatility: when published and updated as a F3 archive, a software project effectively is Open Data on which an unlimited range of applications can rely, even outside of the forge domain (iii) Consistency: it provides a common language to use when talking about the forge related domains (iv) Trust: cryptographic signatures on each F3 dump guard against malicious or unintentional tampering that could compromise the integrity of a software project.

Forge

More info: https://nlnet.nl/project/F3-FriendlyForgeFormat

Website: f3.forgefriends.org
Repository: lab.forgefriends.org/friendlyforgeformat

Federated Timesheets

Interoperable machine-readable time tracking

Description

This project brings together developers from WikiSuite, m-ld.io, Muze and Ponder Source in a collaboration to deliberately research how federated machine-readable data can work between independent software projects on the user-operated internet. We want to showcase how our vision of Federated Bookkeeping can make internet users " connected but sovereign" .

Each project's timesheet system that tracks billable hours will be extended with time tracker apps (locally or on a self-hosted server) to expose machine-readable timesheet data through a query endpoint (reader pull) or through a webhook (writer push).

Furthermore a W3C interest group "federated timesheets" was started that will contain and maintain a repository of time tracker schemas and extend this continuously in an orderly fashion to enable developers to import recipients' schemas as well as add their own to the repository.

TimeTracking

More info: https://nlnet.nl/project/FederatedTimesheets

Website: federatedbookkeeping.org
Repository: github.com/federatedbookkeeping/timesheets

Federated Task-Tracking with Live Data

Track tasks and issues in a federated way

Description

Applications and data are tightly coupled: the format, structure, and meaning of data are almost inseparable from the application generating and using them, hindering the data's portability. Sharing data between applications entails mastering complex and proprietary APIs or export formats, and transforming output data into the necessary structure and meaning for use elsewhere, time-consuming and error-prone activities. Federation is a way of linking different systems together so users can share data by being 'connected, but sovereign'. The precursor Federated Timesheets project successfully pioneered this approach for time-tracking data, bringing together WikiSuite, timeld, and Prejournal such that timesheet data entered into one are easily disseminated to the others. Federated Task-Tracking builds ambitiously on that foundation, with a more complex data model applicable to a broader range of real-world scenarios, introduces live collaborative editing of latency-critical data shared between participating systems.

DataPortability Federation TimeTracking

More info: https://nlnet.nl/project/FederatedTaskTracking

Website: github.com/federatedbookkeeping/task-tracking

Fobnail

Remote attestation delivered locally

Description

The Fobnail Token is a tiny open-source hardware USB device that provides a means for a user/administrator/enterprise to determine the integrity of a system. To make this determination, Fobnail functions as an attestor capable of validating attestation assertions made by the system. As an independent device, Fobnail provides a high degree of assurance that an infected system cannot influence Fobnail as it inspects the attestations made by the system. Fobnail software is an open-source implementation of the iTurtle security architecture concept presented at HotSec07; in addition, it will leverage industry standards like TCG D-RTM trusted execution environment and IEFT RATS. The Fobnail project aims to provide a reference architecture for building offline integrity measurement servers on the USB device and clients running in Dynamically Launched Measured Environments (DLME). It allows the Fobnail owner to verify the trustworthiness of the running system before performing any sensitive operation. Fobnail does not need an Internet connection what makes it immune to the network stack and remote infrastructure attacks. It brings the power of solid system integrity validation to the individual in a privacy-preserving solution.

Attestation OpenHardware

More info: https://nlnet.nl/project/Fobnail

Website: fobnail.3mdeb.com
Repository: github.com/fobnail

ScanCode

FOSS Code Supply Chain Assurance

Mitigate attacks through software dependencies

Description

It is of the utmost importance to ensure that FOSS packages from public repositories have not been tampered with by malicious actors. This type of compromise is described as an open source " supply chain attack" and these have been increasing significantly. This project is building a new system (which is FOSS itself) to help verify the integrity of deployed code packages and validate their origin with external data sources, with the potential to mitigate attacks on open source packages supply chains such as: detecting if a package in use is matching verified code by matching source and binaries exactly and approximately. Or detecting abnormal code changes that may be signs of malicious modifications and possible attacks on a package.

The key components of this open code and data solution are a Package and File Fingerprints Database, a Code Similarity and Changes Detection Engine, utilities to detect possibly malicious changes in upstream projects, and integration in build system(s). While existing approaches may require a tight control of the whole code supply chain, the approach of this project is designed for practical usage with limited changes to a build and CI/CD pipeline.

Fingerprinting **Repology** **Scanning** **SoftwareHeritage** **SupplyChain** **SupplyTransparency**

More info: **https://nlnet.nl/project/FOSS-supplychain**

Website: AboutCode.org
Repository: github.com/nexB/purldb

Gash

Port Gash to GNU Mes for auditable bootstrap

Description

For several years, the GNU Guix project has been reducing the amount of unauditable binary blobs used in bootstrapping its operating system, through efforts such as GNU Mes. This is needed to avoid " reproducibly malicious" behaviour within the software toolchain.

Gash is a POSIX-compatible shell written in Guile Scheme. Gash provides both the traditional shell interface, as well as a Guile library for parsing shell scripts. Once this project is completed, Guix (and other operating systems) can be bootstrapped from legible source, without depending on already compiled compilers or C standard libraries. This will allow to move step by step from a minimal Scheme interpreter to full-blown modern scheme dialects to subsequently much more advanced features and optimisations required during the bootstrap.

Bootstrap **Guix**

More info: https://nlnet.nl/project/Gash

Website: git.savannah.nongnu.org/cgit/gash

Full-source GNU Mes on ARM and RISC-V

Expand full-source bootstrap to other CPU platforms

Description

GNU Mes was created to address the security concerns that arise from bootstrapping an operating system using large binary blobs of several 100s of megabytes, which (incredibly so!) is common practice for the software supply chains in use today. While these days users can reproducibly build software with modern functional package managers like Guix and Nix, the presence of potentially toxic code in these unauditable blobs or the propagation into binaries cannot be excluded. Users have no technical assurance that the executable they use corresponds with the source code - or whether the tool chain which compiled the source code introduce weaknesses or undefined behaviour. By making the toolchain 'bootstrappable' (as per bootstrappable.org), users can verify themselves for every step what happens - in the case of GNU Mes from one tiny (and orders of magnitude more easily verifiable) 357-byte file upwards. The final goal is to help create a " full source" bootstrap for any interested UNIX-like operating system and any type of architectures. In this project the project will add ARM and RISC-V, with other architectures on the roadmap.

ArchitecturePortability Bootstrap

More info: https://nlnet.nl/project/GNUMes-ARM_RISC-V

Website: www.gnu.org/software/mes
Repository: http://savannah.gnu.org/projects/mes

GNU Mes RISC-V

Bringing the trustworthy bootstrap to RISC-V

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 RISC-V platform, an instruction set architecture (ISA) that is provided under open licenses. Combining GNU Mes with an open ISA will provide an extra level of security and trust by extending the auditability of the system from the software to also the hardware.

RISC-V is a relatively new architecture so this effort requires the backport of many tools that were already available for GNU Mes in other architectures. Also the modular nature of RISC-V makes it an specially complex bootstrap target, because it needs to support all the possible RISC-V implementations. This project aims to overcome the current limitations to prepare GNU Mes and all the associated projects for a full RISC-V port.

Bootstrap **RISC-V**

More info: https://nlnet.nl/project/GNUMes-RISCV

Website: ekaitz.elenq.tech/tag/bootstrapping-gcc-in-risc-v.html
Repository: github.com/ekaitz-zarraga

RISC-V bootstrapping effort via GNU Mes

Allow bootstrapping Guix on RISC-V via GNU Mes

Description

This project is a continuation of several previous modest effort that each made good steps in bringing the GNU Mes project to the quickly growing ecosystem of RISC-V. RISC-V is a relatively new instruction set architecture (ISA) for computer chips, and because it obviously has its own variant of the very lowest level of instructions, adopting this new hardware platform for practical use cases requires porting of some software and tools that were already available in other architectures. Such " chip agility" makes the overall technology ecosystem more robust, creating more diversity and consumer choice.

One aspect of working towards chip agility in a trustworthy manner is aiming for a " full source bootstrap" , as pioneered by GNU Mes and others on other architectures. This addresses the security concerns associated with bootstrapping an operating system using large, unauditable binary blobs, which until recently was 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 goal of this project is to complete the port of Mes to RISC-V, and achieve the first full source bootstrap - which is then available to use for any interested UNIX-like operating system. As a first major step towards universal adoption, the project will subsequently package the whole process and include it in Guix's commencement module.

Bootstrap RISC-V

More info: https://nlnet.nl/project/GNUMes-RISCV-bootstrap

Website: ekaitz.elenq.tech/tag/bootstrapping-gcc-in-risc-v.html

Layer-2-Overlay

Generalising the GNUnet Layer-2 Overlay for broader usage

Description

Layer-2-Overlay is a P2P connectivity layer that allows decentralized applications to establish communication with peers. The current Internet architecture is strongly biased in favor of client-server applications. To regain data sovereignty from tech oligopoly, citizens must be able to communicate directly without a few gatekeepers. Therefore decentralized applications need to overcome network obstacles of the existing Internet infrastructure without the need to setup a costly alternative infrastructure. An additional benefit is the effective usage of existing resource, to lower the environmental damage big centralized systems are doing to our planetary ecosystem. The Layer-2-Overlay will achieve this goal by utilizing a variety of existing protocols and infrastructure (Ethernet/WLAN, TCP/UDP, QUIC, Satellite) and an effective flow- and congestion-control to distribute traffic through different channels. After reconnecting the edges (e.g. PCs at home or mobiles) of the existing Internet among each other again, traffic can be forwarded directly to known peers and existing infrastructure will be preserved. The API of Layer-2-Overlay will be usable by all kinds of decentralized application use cases. For a first showcase Layer-2-Overlay will be integrated into GNUnet, an alternative network stack for building secure, decentralized and privacy-preserving distributed applications.

CongestionControl Discovery FlowControl GNUnet Overlay P2P PluggableTransports

More info: https://nlnet.nl/project/GNUnet-L2

Website: www.gnunet.org/en/l2o
Repository: git.gnunet.org/gnunet.git

GNUnet Messenger API

API for decentralized instant messaging using CADET

Description

Communication is one of the most valuable goods, but it requires confidentiality, integrity and availability to trust it. The GNUnet Messenger API implements an encrypted translation layer based on Confidential Ad-hoc Decentralized End-to-End Transport (CADET). Through CADET the API will allow any kind of application to set up a fully decentralized form of secure and private communication between groups of users. The service uses e2e-encryption and does not require any personal information from you to be used.

You are able to send text messages, share files, invite contacts to a group or delete prior messages with a custom delay. Messages and files will both be stored decentralized being only available for others in the group. GNUnet provides the possibility to use this service without relying on the typical internet structures, with a turnkey optional DHT for sharing resources.

Unlike many other messengers out there the GNUnet Messenger service focuses on privacy. You decide who can contact you and who does not. You decide which information gets shared with others and which stays a secret. The whole service and its API is free and open by design to be used by many different applications without trusting any third party.

DesktopApp Encryption InstantMessaging MobileApp

More info: https://nlnet.nl/project/GNUnet-Messenger

Website: gnunet.org/en
Repository: git.gnunet.org/messenger-gtk.git

GNU Taler KYC

Know-Your-Customer support for GNU Taler

Description

This work is about adding proper Know-Your-Customer (KYC) support to GNU Taler to satisfy regulatory requirements to operate the Taler payment service. However, we will not implement our own KYC solution but instead provide a generic way to interface with existing KYC providers and implement several concrete adapters. By supporting multiple providers we will ensure that our KYC abstraction is reasonably generic.

The KYC integration will be configurable to adjust the deployment to the legal requirements of different countries. Finally, we will support attestation of collected KYC information to third parties. This will allow the payment system to assure consumers receiving a bill about the identity of the invoicing business.

Banking Compliance ElectronicPayment KYC

More info: https://nlnet.nl/project/GNUTaler-KYC

Website: taler.net

Gosling

Generic Onions Services Library Project

Description

One of the internet's core infrastructural flaws is a lack of anonymity - yet anonymity is a form of privacy that many users would prefer to have. Building products which preserve this user privacy while also being featureful and easy to use is difficult. Part of this difficulty has to do with the fact that developers need to be aware of and actively counter the myriad ways users can be de-anonymised (e.g. fingerprinting, side-channels). This requires knowing many intricate details at all levels of the software stack.Project parent Blueprint for Free Speech's goal is to gradually increase the portion of the internet that offers anonymity. By creating a "generic onions services library" (Gosling), we can help developers create secure and anonymous p2p applications without having to delve too deeply into protocol design or the Tor spec, and to do so with more security assurance.

Library **OnionRouting**

More info: https://nlnet.nl/project/Gosling

Website: github.com/blueprint-freespeech/gosling

Porting Guix to Riscv64

Port Guix software collection to Riscv64 architecture

Description

This project will work on bringing the Rust support of GNU Guix on Riscv64 up to fully supported, with the bootstrap chain from source. It will also bring Riscv64 in Guix up to the full level of support that is expected of commonly used architectures, ready to be used in all the applications where GNU Guix is already found. Riscv64, being an Open Architecture, freely available to anyone who wants to implement processors, goes a long way towards ensuring that our future computing platforms are free of hidden backdoors. GNU Guix, being a true Free Software Operating System and compiled from source from a small bootstrap binary, with reproducibility guarantees, is as close as the computing community has come to a fully auditable software chain that makes sure all the software we run on our computers is what we intend, and nothing more. By combining the Riscv64 architecture and GNU Guix for software we can reach toward a fully secure and auditable computing platform that we might consider trusting.

RISC-V

More info: https://nlnet.nl/project/Guix-Riscv64

Website: guix.gnu.org

TPM 2.0 for HEADS

TPM 2.0 support for open source BIOS replacement firmware

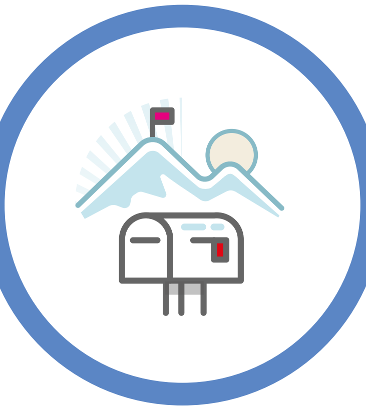
Description

HEADS is an open source custom firmware for laptops that aims to provide slightly better physical security and protection for data on the system. HEADS combines physical hardening of specific hardware platforms and flash security features with custom coreboot firmware and a Linux boot loader in ROM. This moves the root of trust into the write-protected region of the SPI flash and prevents further software modifications to the bootup code. HEADS allows to verify that laptop hardware has not been tampered with in transit or in your absence (so-called evil maid attack). Until now HEADS is mostly used with older Thinkpad X230 and T430 laptops. As part of this funded project we will develop HEADS to support state of the art hardware.

BIOS TPM

More info: https://nlnet.nl/project/HEADS-TPM2.0

Website: github.com/osresearch/heads



Himalaya

End-to-end encryption capable scriptable email

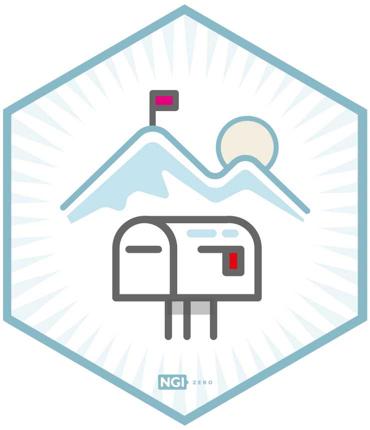
Description

Himalaya is a cross platform and open source toolsuite for managing emails. Its aim is to extract the email business logic into a safe and secure Rust library, so it can be consumed by any compatible client. This architecture makes the tool very flexible and versatile: move batch of emails from the command-line input, automatically sign or decrypt emails levering OpenPGP's web of trust, view HTML version of emails from the terminal, write emails with your favourite text editor, set up a new message notifier in a systemd daemon, view emails from a graphical user interface alla Thunderbird… possibilities are endless! The funding from NLnet will be used to release the first production-ready version of the library and to release few compatible clients like a CLI, a TUI, a GUI, a Vim plugin and an Emacs plugin. Himalaya also plans to extend the concept to other email-related domains, like contact management, events/calendar management, tasks management etc.

Email IMAP OpenPGP POP

More info: https://nlnet.nl/project/Himalaya

Website: pimalaya.org
Repository: github.com/soywod/himalaya



Hyper Hyper Space

Cryptographically secure append-only distributed data layer

Description

The Hyper Hyper Space project aims to make distributed applications easy to build and usable by anyone. It introduces “spaces”, shared information objects that are stored locally (on personal computers or phones) and can be easily replicated over the network to any number of participants and kept synchronized. Spaces have formats (just like files): blogs, discussion forums, e-commerce stores, etc. can be represented as space-types. Instead of filenames or URLs, spaces can be universally looked up by entering a 3-word code into the application. This code is used to find devices hosting the space, and then to fetch and validate it.

Application designers can build upon a library of building blocks supplied by Hyper Hyper Space (e.g. cryptographic identities, CRDT-inspired datatypes, etc.) that work over append-only DAGs. Once a space is defined this way, its synchronization can be handled by Hyper Hyper Space transparently, simplifying application development. Finally, to make spaces universally available, the Hyper Hyper Space runtime works inside an unmodified web browser (as a JavaScript library: IndexedDB is used for in-browser storage, WebRTC as transport - no extensions are needed). Thus a distributed application can be deployed as a static website that fetches its contents from a browser-to-browser mesh.

Ultimately, the Hyper Hyper Space project’s goal is to encourage open information formats and software interoperability, helping make open source, non-for profit and public interest application development sustainable.

CRDT DAG P2P RealTimeCollaboration

More info: https://nlnet.nl/project/HyperHyperSpace

Website: www.hyperhyperspace.org
Repository: github.com/hyperhyperspace

imap-codec library

Release version 1.0 of the imap-codec library

Description

With an expected volume of 333 billion messages per day in 2022, email is one of today's most common methods to exchange information on the Internet. For better or worse, email is unlikely to go away soon, meaning that even the latest software needs to support it in a trustworthy and resilient way. imap-codec is a misuse-resistant IMAP parsing and serialization library focusing on correctness and security. It should pave the way for a new generation of email clients, servers, and utilities written in Rust and become a reusable building block for the Next Generation Internet. To archive that, it is essential to stabilize the API, improve testing, provide excellent documentation, and establish a welcoming and sustainable open-source environment for imap-codec.

Email **IMAP**

More info: https://nlnet.nl/project/imap-codec

Website: github.com/duesee/imap-codec

Interpeer SDKs

Secure and efficient peer-to-peer networking stack

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. In order to make the Interpeer technology stack accessible to software developers, the goal is to provide SDKs for a desktop and a mobile platform, complete with examples. These SDKs should enable seamless cross-platform data exchange and live editing capabilities by multiple authors.

P2P

More info: https://nlnet.nl/project/Interpeer-SDK

Website: interpeer.io
Repository: codeberg.org/interpeer

IPDL

Equational Proofs for Distributed Cryptographic Protocols

Description

In cryptography, interactive, distributed cryptographic protocols are most often proved secure using the simulation paradigm, wherein the protocol of interest is proved (approximately) equivalent to an idealization. The simulation paradigm is extremely powerful, as it allows a wide range of security properties to be captured under one definition. On the other hand, while expressive, the simulation paradigm presents extra complications for formally verifying security proofs. Proving equivalences between distributed protocols in general requires heavyweight techniques based on manually constructing so-called bisimulations (suitable relational invariants), which creates a barrier to entry for formal methods. We lower this barrier to entry with IPDL, or Interactive Probabilistic Dependency Logic, a new process calculus for cryptographic protocols. IPDL includes an approximate equational logic that allows computationally sound reasoning about protocols in a manner both close to the simulation paradigm and amenable for formal verification. Using IPDL, we deliver short, simulation-based proofs of variety of cryptographic protocols. Our most complex and very general case study verifies the n-party GMW protocol for secure function evaluation.

FormalVerification MultiPartyComputation

More info: https://nlnet.nl/project/IPDL

Website: github.com/kristinas/IPDL-Maude

json-joy

JSON data structure as a CRDT

Description

Conflict-Free Replicated Data Types (CRDTs) are specialized data structures that enable the merging of changes in two or more data replicas without conflicts. Despite their immense potential, CRDTs remain a relatively new area of research and development, and much can be improved in existing open source CRDT libraries. The objective of the json-joy project is to implement a full JSON CRDT library that reflects the current state of the art, while simultaneously ensuring optimal performance through the use of custom-designed data structures and the latest advancements in Replicated Growable Array (RGA) literature. In addition, the project aims to establish specifications for critical components of the library, including the data types employed, serialization protocols, and patch format protocols, thereby facilitating the portability of the open source code to other programming languages and promoting educational initiatives.

CRDT

More info: https://nlnet.nl/project/JSON-Joy

Website: jsonjoy.com
Repository: github.com/streamich/json-joy

Kaidan

Encrypted A/V calls, group chat messaging

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.

AudioCall OMEMO Videocalling XMPP

More info: https://nlnet.nl/project/Kaidan-Groups

Website: kaidan.im
Repository: invent.kde.org/network/kaidan

Kaidan

Kaidan Mediasharing

Media sharing and improved contacts for Kaidan XMPP

Description

Kaidan is a user-friendly and modern chat app for every device. It uses the open communication protocol XMPP (aka Jabber). Kaidan is a convergent app, capable of supporting different device dimensions. It runs on a variety of mobile and desktop systems including Android, FreeBSD, Linux, macOS, Plasma Mobile, Ubuntu Touch and Windows. Kaidan uses the open communication standard XMPP, which is built around federation. That way, users can individually pick from a variety of apps, servers and service providers - or even run their choice of software themselves so that they are not dependent on any service provider or company.

In this project, the team will work in particular on improving media sharing, including smoothening the overall user experience. In addition, a number of useful XEPs will be implemented, such as XEP-0368 (" SRV records for XMPP over TLS for Direct TLS support") and XEP-0484 (" Fast Authentication Streamlining Tokens") which speed up and strengthen transport layer security. Within this project the team will also refactor and fix presence subscription handling, enabling the user to accept presence subscription requests at any time. Where possible, features are upstreamed to the cross-platform C++ XMPP client and server library Qxmpp.

IM OMEMO TLS XEP XMPP

More info: https://nlnet.nl/project/Kaidan-Mediasharing

Website: kaidan.im
Repository: invent.kde.org/network/kaidan

Kaidan
NGI ZERO

Katzen

Meta-data resistant instant messaging over the Katzenpost mixnet

Description

Katzen is a new private instant messaging application built using the Katzenpost mixnet project, which is an overlay network that is able to hide communication patterns of individual users from passive network observers. This means that attackers cannot link sending and receiving of messages on the network with any of the participants. Messages between conversation parties are delivered to and read from message queues operated by the mixnet service operators. The legacy simple design maintains a per client queue and is able to see when a client is receiving a message, how often clients receive messages, and when the client is online and checking for their messages. The purpose of this project is to replace the legacy ephemeral message storage system used by Katzen with a replacement that does not link messages with a specific user or conversation, To do this, clients will include a csprng seed as part of the contact creation process that will be used to generate a deterministic sequence of message identifiers between conversation participants; these identifiers will be used by each client to query the ephemeral storage provider for the next message in the conversation. Because polling the storage service adds latency, and this design must check for new messages from each conversation partner, mechanisms to reduce the number of round trips - such as using SURBs as an asynchronous callback upon message delivery on the storage provider will be explored as a means to build a mixnet ‘push’ service to decrease the total round trip delay in receiving a new message.

InstantMessaging **Mixnet**

More info: https://nlnet.nl/project/Katzen

Website: katzenpost.mixnetworks.org
Repository: github.com/katzenpost

KDE Connect

KDE Connect discovery and transport protocol improvements

Description

KDE Connect allows devices on a local network to discover each other and, after an initial pairing process, exchange data over an encrypted connection. Leveraging this abstraction, the KDE Connect desktop and phone apps provide cross-device syncing features like sharing files, notifications, input devices, multimedia controls and more.

There are multiple independent implementations of the KDE Connect protocol written in C++, Java, Swift, Javascript, and more; as well as various applications using the protocol targeting different operating systems.

The aim of this project is to reimplement KDE Connect's discovery process and transport protocol, which were shaped by the limitations of the smartphones of 10 years ago, using multicast and modern TLS.

CrossDevice **mDNS**

More info: **https://nlnet.nl/project/KDE-Connect**

Website: kdeconnect.kde.org
Repository: invent.kde.org/network/kdeconnect-kde

Standardizing KEMTLS

Post-quantum TLS without handshake signatures

Description

KEMTLS is a recent academic proposal for an alternative way of adding authentication to the Transport Layer Security (TLS) protocol. The project is motivated by the need to migrate public key cryptography to new algorithms that resist attacks by quantum computers. Compared to traditional cryptography, post-quantum signature schemes generally have larger public keys and/or signatures, and need more computational effort. KEMTLS, published at the ACM Computer and Communications Security Conference in 2020, replaces signature-based authentication for web servers with a post-quantum key exchange (called a KEM) in a way that saves communication and computation.

In this project we aim to prepare KEMTLS for standardization by the Internet Engineering Task Force (IETF). To that end we will implement KEMTLS in a few different open source TLS software libraries and demonstrate the viability and interoperability of these implementations. This software will assist later implementers of KEMTLS by allowing to validate their implementations against our reference. We will also investigate optimizations for using KEMTLS in specialized environments like IoT, and will investigate issues involving certification of KEM keys.

IETF PostQuantumCrypto TLS

More info: https://nlnet.nl/project/KEMTLS

Website: kemtls.org
Repository: github.com/kemtls

Private Key Operations for Keyoxide

Implement Private Key Store design in Keyoxide

Description

Keyoxide is one of the open-source success stories when it comes to providing an alternative to the proprietary product (Keybase). The UI is straightforward so that the interaction with the site is available to all kinds of users. Unfortunately there is one critical part that differentiates Keyoxide from Keybase - no support for private key operations. Adding proofs requires a complex maze of command line invocations. This project will implement best of both worlds: simple, UI centric way of interaction without technical knowledge required and the strong security of Keyoxide.

DigitalSignature OpenPGP

More info: https://nlnet.nl/project/Keyoxide-PKO

Website: keyoxide.org
Repository: codeberg.org/keyoxide/web

Keyoxide Mobile

Mobile client for identity magement tool Keyoxide

Description

The Keyoxide Mobile app is an open source keyoxide client for Android that lets you verify and manage decentralized cryptographic identities while being on the go. To verify somenone else's decentralized identity: simply enter their identifier or scan their qr-code to see the verification result generated by the app. With the funding from NLnet, the app will be able to create new Keyoxide profiles and additional features will be added such as iOS support, a design update, being able to save multiple profiles, text encryption/decryption, custom instance support, accessibility features like localization, color themes and contrast.

IdentityManagement OpenPGP

More info: https://nlnet.nl/project/Keyoxide-Mobile

Website: mobile.keyoxide.org
Repository: codeberg.org/Berker/keyoxide-flutter

Keyoxide v2

Add cryptographic signature based to Keyoxide

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 build on top of the existing OpenPGP Identity Proofs to add other types of profiles based on various cryptographic signature mechanisms from a variety of new tools. To maintain linkable profiles, a new signature-hosting infrastructure needs to be designed and developed. Other improvements are aimed at safeguarding privacy and achieving plausible deniability.

DigitalSignature **OpenPGP** **ServerApp**

More info: https://nlnet.nl/project/Keyoxide-signatures

Website: keyoxide.org
Repository: codeberg.org/keyoxide/web

K-Gen

From datasets in DCAT catalogs to knowledge graphs

Description

Data Catalogs are an important building block for a knowledge graph. Most available open-source data cataloging solutions, however, are tailored either to the needs of dataset publishers or to bigger companies with existing data warehouses or data lakes. Open data communities or smaller-sized companies do have not many options to choose from when it comes to lightweight solutions to catalog their existing data assets or collect existing metadata about relevant datasets for their needs. K-Gen will be such a lightweight data catalog solution. It will be based on DCAT, the W3C standard for data catalogs, which has been widely adopted in the public sector for the publishing of open datasets.

In the first development phase, the milestone of a basic data catalog to collect metadata about datasets of a user and a general data processing pipeline to import existing metadata about datasets from various sources and various formats, including ways to keep them in sync with the original source should be developed. Further development should then provide tools to build a knowledge graph over the content of the datasets of the data catalog.

DCAT Elixir OpenData SPARQL W3C

More info: https://nlnet.nl/project/K-Gen

Website: github.com/ontogen/ontogen

Kintex-nextpnr

Open toolchain for high performance FPGAs

Description

FPGAs are reconfigurable chips capable of handling many electronic signals in parallel. They are used in network equipment like backbone switches, firewalls, video devices like surveillance cameras and radio equipment like mobile-phone base stations and radar systems and satellites to process high volumes of data with very low latency. FPGAs are also used to test digital circuit designs before they are manufactured as chips.

The functionality of FPGAs is determined by a configuration file which is loaded into the FPGA at power-on. The configuration file is usually generated from a design file by a proprietary closed source tool provided by the manufacturer of the FPGA. nextpnr-Kintex will provide a complete set of open source tools to generate a configuration file for the widely used family of Kintex7 FPGAs from manufacturer Xilinx/AMD without having to use any proprietary tools. This will empower digital design engineers to have the guarantee that no backdoor is implemented on FPGA based devices by the proprietary design tool provided by the vendor. The availability of the source code of the FPGA design tool will also allow innovators to come up with new use cases for FPGAs currently not possible with proprietary tools. Overall, the project will help to increase the security of FPGA based wired and wireless network infrastructure in Europe.

FPGA **Toolchain**

More info: **https://nlnet.nl/project/Kintex-nextpnr**

Website: github.com/openXC7

Let's Connect! Client-Server to P2P

Add P2P features to Let's Connect!

Description

Let's Connect! provides an open-source VPN solution allowing ISPs, hosting providers and businesses to easily set up a secure VPN service. Currently Let's Connect! has been engineered in a traditional client-server VPN model. Basically connecting the client with VPN technology into the organization where the VPN server is deployed.

Let's Connect! is also used in the educational and research community under the name eduVPN. Roughly 140 organisations, and estimated 300K users, around the globe are using eduVPN.

The current client-server model of Let's Connect! doesn't facilitate directly connecting devices located in various places, like IoT devices at home or services offered in various datacenters or (public) cloud environments.

This project focusses on engineering a P2P solution integrated with Let's Connect! VPN, which empowers users to connect safely to all their devices, anywhere on the internet.

P2P VPN

More info: https://nlnet.nl/project/LetsConnect-P2P

Website: www.letsconnect-vpn.org
Repository: github.com/eduvpn

LiberaForms

LiberaForms

End tot End Encrypted Forms

Description

Cloud services that offer handling of online forms are widely used by schools, associations, volunteer organisations, civil society, and even families to publish questionnaires and collect the results. 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 because forms may not only include personal details such as their name, address, gender or age, but also more intimate questions including 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.

LiberaForms is a transparent alternative for proprietary online forms that you can easily host yourself. In this project, LIberaForms will add end-to-end encryption with OpenPGP, meaning that the data is encrypted on the client device and only the final recipient of the form data can read it (and not just anyone with access to a server). Also, the team will add real-time collaboration on forms, in case users need to fill out forms together.

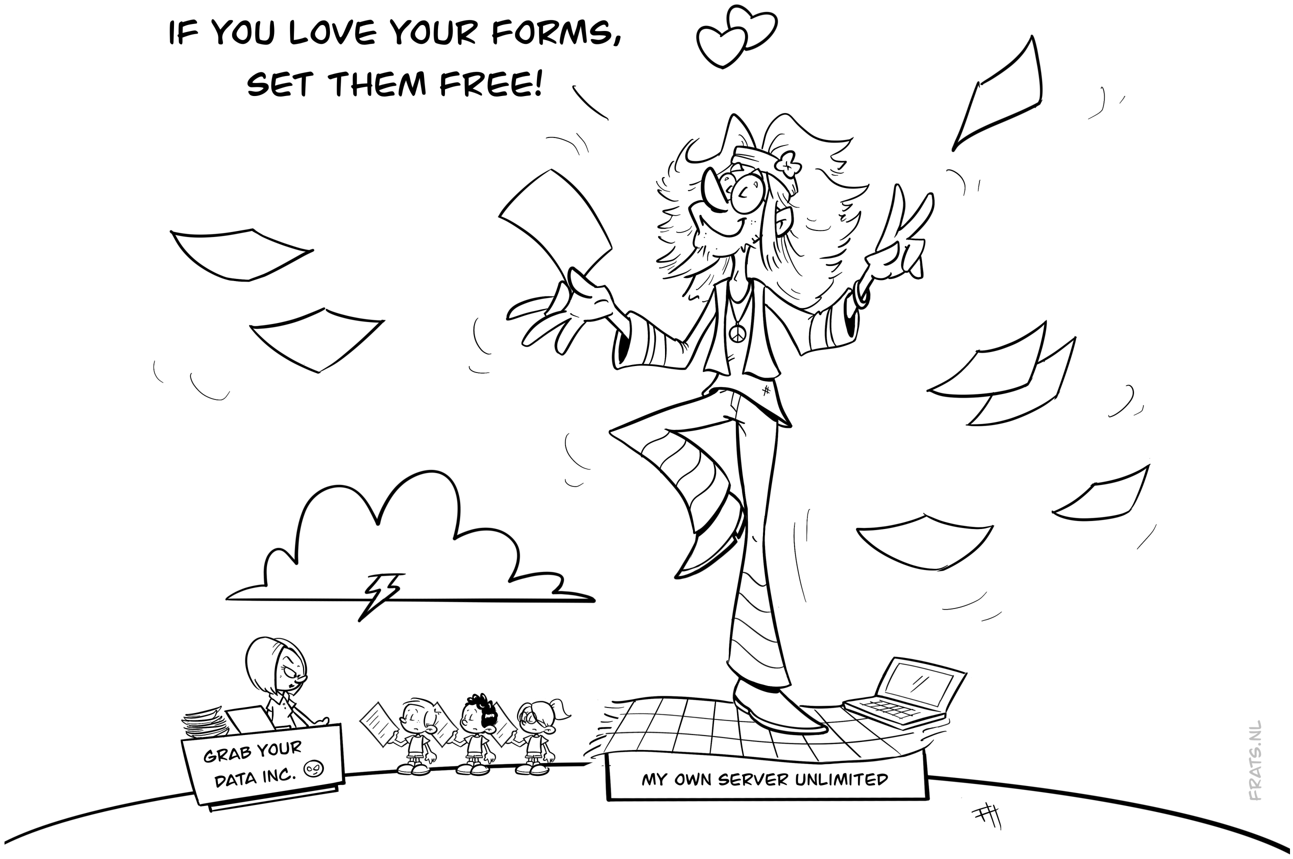
CRDT **Forms** **OpenPGP**

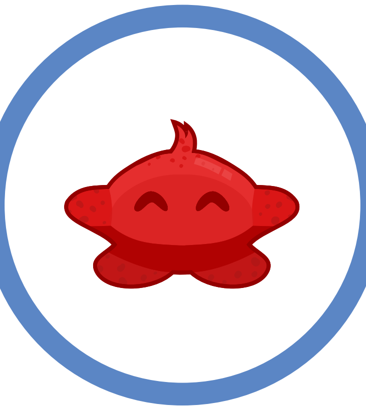
More info: https://nlnet.nl/project/LiberaForms-E2EE

Website: liberaforms.org
Repository: farga.exo.cat/LiberaForms

LiberaForms

NGI ASSURE


IF YOU LOVE YOUR FORMS, SET THEM FREE!
GRAB YOUR DATA INC.
MY OWN SERVER UNLIMITED
FRATS.NL



Audio/Video Calls in Libervia

Encrypted Audio/Video Calls in multi-frontend XMPP client

Description

Libervia is a multi-frontend, multi-purpose XMPP client. It doesn't just focus on instant messaging, and uses the open standard to provide features such as blogging/microblogging, calendar events, file sharing, end-to-end encryption, etc.

Some of the last major missing features include audio/video conferencing and desktop sharing. The goal of this project is to implement one2one calls first and then multi-user conferencing and desktop sharing, while using the e2e encryption mechanisms provided by the ecosystem where possible. These features will be available on the various front-ends, including web, desktop, and even command line.

Compatibility will be ensured with the wider XMPP ecosystem, to ensure that calls can be made without problems with other software such as Conversations or Movim.

AudioCall OMEMO Videocalling XMPP

More info: https://nlnet.nl/project/Libervia-AV

Website: libervia.org
Repository: repos.goffi.org/libervia-backend

Librecast

E2E encrypted multicast

Description

The Librecast project contributes to decentralising the Internet by enabling multicast. It builds transitional protocols and software to extend the reach of multicast and enable easy deployment by software developers. This can for instance help to synchronise large evolving datasets to many users at the same time (even hundreds of gigabytes of blockchain data) in an economic, reliable, transparent and fair way - unlike with unicast, everyone can get a copy of the same packets received by everyone else. Not depending on a centralised structure (anyone can be the upstream source), means it is very robust as well. LibreCast is energy efficient and as a next generation internet technology offers confidentiality and security - and is sustainable, has high scalability and throughput.

Librecast Live is a Multicast Live Streaming, Conferencing and Remote Collaborative Work Environment. It is a versatile multicast platform flexible and scalable enough to be used for live-streaming, classrooms and conferences - using an ad hoc or previously established web of trust. While using multicast helps solve the scalability inherent with this kind of setup, actually all messages are transmitted over encrypted channels - providing strong privacy and integrity assurances through E2E encryption.

Multicast **Videostreaming**

More info: https://nlnet.nl/project/LibreCastLiveStudio

Website: librecast.net
Repository: codeberg.org/librecast/librecast

libresilient

Create robust web presence with service workers and DHT

Description

A browser-based decentralized content delivery network, implemented as a JavaScript library to be deployed easily on any website. LibResilient uses ServiceWorkers and a suite of non-standard in-browser delivery mechanisms, with a strong focus on decentralized tools like IPFS. Ideally, users should not need to install any special software nor change any settings to continue being able to access an overloaded LibResilient-enabled site as soon as they are able to access it once.

CDN DHT Library P2P Robustness SelfHosting ServiceWorkers

More info: https://nlnet.nl/project/libresilient

Website: resilient.is
Repository: gitlab.com/rysiekpl/libresilient

The Libre-SOC Gigabit Router

Native Open Hardware chip implementation of crypto primitives

Description

The Libre-SOC Project is developing a Libre System-on-a-Chip in a transparent fashion to engender end-user trust. Based on the OpenPOWER ISA, the next logical step is to extend and modernise OpenPOWER into the cryptographic and blockchain realm, and to do so in a practical way: design a Router ASIC. Whilst many commercial ASICs would do this using hard-coded non-transparent blocks or instructions, true transparency really only exists if the ISA has general-purpose primitives that can be Formally (mathematically) validated. The Libre-SOC Crypto-router Project therefore goes back to mathematical " first principles" to provide general-purpose Galois-Field, Matrix abstraction and more, on top of Simple-V Vectorisation. This provides flexibility for future cryptographic and blockchain algorithms on a firm transparent foundation.

ASIC Cryptography FormalProof Libre-SOC OpenHardare OpenHardware OpenPower

More info: https://nlnet.nl/project/LibreSOC-GigabitRouter

Website: libre-soc.org/nlnet_2021_crypto_router
Repository: git.libre-soc.org

lpnTPM

TPM 2.0 compliant open hardware Trusted Platform Module

Description

lpnTPM is Open Source Software (OSS), and Open Source Hardware (OSHW) Trusted Platform Module (TPM, also known as ISO/IEC 11889) is an international standard for a secure cryptoprocessor, a dedicated microcontroller designed to secure hardware through integrated cryptographic keys. What makes lpnTPM different from generally available solutions is openness. Software and hardware of lpnTPM can, without limits, be audited, fixed, and customized by communities and businesses. Open design address the lack of trustworthiness of proprietary closed source TPM products, which currently dominate the whole market. lpnTPM in production mode protects software by secure boot technology, and only the lpnTPM owner will update it. TPM modules enable measured boot and support verified boot, Dynamic Root of Trust for Measurement, and other security features. Another benefit of lpnTPM would be physical design, which solves the lack of standardization around pinout and connector. The ultimate goal of lpnTPM is to provide a trustworthy platform for future open evolution of Trusted Platform Module software and its application to various computing devices, resulting in better adoption of platform security.

Firmware TPM

More info: https://nlnet.nl/project/lpnTPM

Website: lpnplant.io
Repository: github.com/lpn-plant

LumoSQL at-rest data security

Modern embedded database with encryption and signed data

Description

LumoSQL is an embedded database that combines various modern database technologies into a single powerful abstraction while remaining a drop-in replacement for the most-used database worldwide, SQLite. LumoSQL brings to embedded databases features including built-in encryption, per-row checksum verifiability of all data (without the overhead of e.g. a blockchain), and a choice of storage backends.

In this project the LumoSQL community works towards the 1.0 version which will add a slew of attractive features such as encrypted embedded data at-rest (which can be unlocked either through role based access control or even outside of unmodified apps with a hardware token like Nitrokey), signed data rows and data tables (so users can cryptographically verify the integrity of data), as well as improved documentation and cross-platform availability. In addition the project is producing valuable tools such as the not-forking project, which addresses the root cause of many real-world security issues as customisation without such a tool requires hard-to-maintain forking.

Database

More info: https://nlnet.nl/project/LumoSQL-at-rest

Website: lumosql.org
Repository: lumosql.org/src/lumosql/doc/trunk/README.md

Maemo Leste Telepathy

Modernise open source real-time communications stack

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.

In this effort project the Maemo Leste team will update the Telepathy real time communications framework (which should benefit all other users of that ramework) and add among others double ratched based OMEMO encryption to XMPP.

MobileOS SIP XMPP

More info: https://nlnet.nl/project/MaemoLeste-Telepathy

Website: maemo-leste.github.io
Repository: github.com/maemo-leste

Maemo Leste

Manyverse Private Groups

Implement SSB Private Groups in Manyverse

Description

Manyverse is a peer-to-peer social network built on the SSB protocol where users themselves are responsible for the network. It is used by thousands of people, on both mobile and desktop. Users can share public posts with each other, but there is currently no way to write private messages to closed communities of a dozen members or more. With this project, we want to implement and improve SSB Private Groups for adoption in Manyverse. This is a cryptographic mechanism to ensure that communities can talk in private. Additionally, we want to make sure that these communities have the tools they need to moderate and prune their social space for safety.

GroupMessaging SBB

More info: https://nlnet.nl/project/Manyverse-PrivateGroups

Website: manyver.se
Repository: gitlab.com/staltz/manyverse

Mellium

Add OMEMO support to XMPP library

Description

Mellium is an XMPP library that helps other projects safely interoperate using the most widely used, federated, real-time communication protocol in use today. Unfortunately, it does not currently provide a mechanism to enable projects using it to communicate in an end-to-end encrypted manner, meaning those projects must do the hard (and potentially dangerous) work of implementing encryption themselves. This project aims to create an easy to use implementation of the OMEMO encryption standard (XEP-0384: OMEMO Encryption) that is compatible with popular instant messaging clients. This will encourage projects depending on Mellium to implement strong privacy protections by lowering the barrier to entry for end-to-end encryption.

OMEMO XMPP

More info: https://nlnet.nl/project/Mellium

Website: mellium.im
Repository: codeberg.org/mellium

MirageVPN

Robust OpenVPN client and server, and QubesOS client

Description

OpenVPN is a virtual private network protocol which is still widely used. We will extend the existing MirageOS OpenVPN implementation in three aspects: develop a unikernel suitable for QubesOS, develop an OpenVPN server, and add recent features (e.g. tls-crypt v2) .

The project builds on top of MirageOS: a library operating system developed in OCaml — a memory-safe functional programming language. In MirageOS, each service is a separate unikernel with a minimal attack surface that only contains the code required to run it. These unikernels are normally executed as a virtualized machine such as KVM, VirtIO, Xen. MIrageOS also supports using a strict security feature of the Linux kernel called seccomp.

The elliptic curve primitives used in this project are correct by construction (and free of timing side channels), and have been developed in Coq as part of the Fiat-Crypto project.

Unikernel VPN

More info: https://nlnet.nl/project/MirageVPN

Website: robur.coop/Our%20Work/Projects#miragevpn
Repository: github.com/robur-coop/miragevpn

Securing Decentralised Live Information with m-ld

Collaborative editing of LInked Data based on CRDT

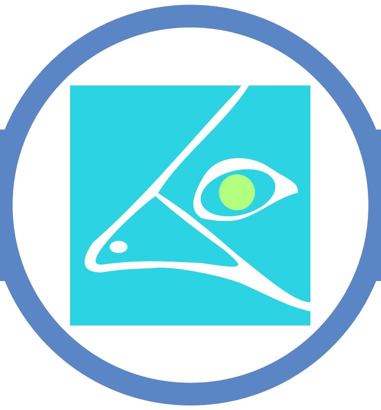
Description

m-ld is a software technology for live information sharing. It enables software engineers to reliably add real-time collaboration, support for offline working, and service resilience to both new and existing software architectures. It achieves this by operating at an " information" level, creating reusable patterns for maintaining the consistency and integrity of application content that is being edited from multiple locations at once. m-ld is built from the ground up on a W3C standard information representation, contributing ideas for its evolution, and is committed to open standards and open source. This project will research and prototype modifications to the primitives of the m-ld core protocol to natively support strong assurance of data integrity and traceability, with authority assignable to identified users or groups, so that they can be reliably assured of the integrity and controlled availability of their data.

CRDT LinkedData RealTimeCollaboration SPARQL W3C

More info: https://nlnet.nl/project/m-ld

Website: m-ld.org
Repository: github.com/m-ld/m-ld-spec



Monal IM

Free Jabber/XMPP client for iOS and macOS

Description

Monal is a open source XMPP instant messaging client for MacOS and iOS which strives to be the go-to client for these platforms just like the app Conversations is for Android. XMPP in general is an open and standardized protocol for real time communication. Anyone can host their own server and communicate freely with each other, just like with email and just like email the used addresses are of the form " user@ domain.tld" . In this project, Monal will among others add end-to-end encryption to its chat interface, in this case the OMEMO XEP which uses a so call double ratchet mechanism to provide strong protection of the confidentiality of messages.Within the project, the team will also implement various other XEPs such as audio and Video (A/V calls), adding modern functionality and improving interoperability with other clients.

Encryption **InstantMessaging** **RealtimeCommunication** **XMPP**

More info: https://nlnet.nl/project/Monal-IM

Website: monal-im.org
Repository: github.com/monal-im/Monal

secsync

SecSync

Efficiently combine end-to-end encryption with CRDTs

Description

While popular CRDT implementations like Yjs or Automerge offer several designs and even implementations on how to asynchronously exchange data using servers, there is no plug & play implementation serving end-to-end encrypted systems. Focus of the first version of SecSync is to provide a protocol to efficiently exchange and resolve e2e encrypted CRDTs. It comes with a plug and play reference implementation on top of Yjs and should be well documented. By leveraging snapshots as well as operations logs referencing snapshots the load times should reduced while still offering real-time collaboration.

Benchmarking **CRDT** **CollaborativeEditing**

More info: https://nlnet.nl/project/Naisho

Website: www.secsync.com
Repository: github.com/serenity-kit/secsync

Namecoin: Electrum-NMC

Security hardening and futureproofing Namecoin and Electrum-NMC

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 independently from the DNSSEC root trust chain, and can thus offer additional security under some circumstances. This project will focus on improving Namecoin's lightweight client (Electrum-NMC) in the areas of security (e.g. sandboxing and test coverage), scalability (e.g. more compact network protocol), UX (e.g. domain management GUI improvements), and packaging (e.g. for Debian and derived distros).

NamingSystem

More info: https://nlnet.nl/project/Namecoin-Electrum-NMC

Website: namecoin.org
Repository: github.com/namecoin/electrum-nmc

NeoChat

Native Matrix encrypted instant messaging client

Description

NeoChat is a client for Matrix, an open and decentralized chat protocol. NeoChat is using Qt and KDE technologies to run on many platforms: Linux, Windows, macOS, Plasma Mobile and Android. One of the biggest missing features for NeoChat is support for end-to-end encryption. Currently, all the messages are sent unencrypted and encrypted conversation can't be read in NeoChat. This is not a problem for public rooms since they are usually not encrypted, but it makes NeoChat unsuitable for usage in a private or professional context. The goal of this project is to enable support for encryption in NeoChat. Since NeoChat uses libQuotient, a client library for the matrix protocol, most of the work will take place in libQuotient. This means that the work done in the project will also help other Matrix clients and bots built with Quotient, in particular Spectral and Quaternion.

Cryptography **DoubleRatchet** **InstantMessaging**

More info: https://nlnet.nl/project/NeoChat

Website: apps.kde.org/nl/neochat
Repository: invent.kde.org/network/neochat

Packet classification extensions for Netfilter

High throughput packet classification of tunneled traffic

Description

With the advent of virtualization and containers, datacenter traffic is becoming prominently tunneled through layer 2 and layer 3 encapsulation techniques such as VLAN, GRE, VxLAN, GRETAP and Geneve among others. Extended packet classification through advanced string-matching also allows to proactively detect malicious traffic patterns and to improve overall datacenter network security. Performance is also a paramount aspect to improve resource utilization and to allow packet classification to scale up to the increasing demands in latency and bandwidth.

Nftables is the next generation packet classification software that replaces {ip,ip6,eb,arp}tables which reuses the existing main components of the Netfilter frameworks such as Connection tracking, NAT and logging. This project aims at three goals: 1) Enhancing Nftables packet classification by extending its tunneled packet classification capabilities to allow to match on inner header, 2) add string-matching infrastructure for Nftables and 3) evaluate performance to analyze bottlenecks and deliver upstream enhancements for the Netfilter packet classification datapath.

VPN

More info: https://nlnet.nl/project/Netfilter-PacketClassification

Website: www.netfilter.org
Repository: git.netfilter.org/nftables

neuropil

DHT based overlay network

Description

The neuropil protocol is a new integration protocol for the IoT, which can be embedded into applications and devices. It facilitates and recombines messaging paradigms with distributed hash tables, self-sovereign identities and named-data networks to establish a new kind of privacy- and security-by-design overlay network. The protocol itself embraces self-containment, reducing the need for external systems/dependencies. Our goal is a trustworthy, democratized access control mechanism for the internet of everybody. Within our project we would like to leave the beta-phase and realize the first full release of our protocol. To reach this goal we will add two remaining critical parts to our protocol: distributed time calculations and distributed linked time-stamping authorities. The first addition is not only crucial for systems without an RTC, but it also enables a de-centralized time service with a much lower attack surface. The second builds upon the first and is a key requirement to establish trust between entities using the protocol. It can also be used to ensure the integrity and to keep-track of (search-) contents of peers. Furthermore we will review our current reference implementation for efficiency and use less power-hungry algorithms whenever possible to support the green deal of the European Union.

DHT EmbeddedSystems NDN

More info: https://nlnet.nl/project/Neuropil-DHT

Website: www.neuropil.org
Repository: gitlab.com/pi-lar/neuropil

NextGraph

Interlinked data graphs, with privacy, security, data locality, and interoperability in mind

Description

NextGraph brings about the convergence between P2P and Semantic Web technologies, towards a decentralized, secure and privacy-preserving cloud, based on CRDTs. This open source ecosystem provides solutions for end-users and software developers alike, wishing to use or create decentralized apps featuring: live collaboration on rich-text documents, peer to peer communication with end-to-end encryption, offline-first, local-first, portable and interoperable data, total ownership of data and software, security and privacy. Centered on repositories containing semantic data (RDF), rich text, and structured data formats like JSON, synced between peers belonging to permissioned groups of users, it offers strong eventual consistency, thanks to the use of operation-based CRDTs. Documents can be linked together, signed, shared securely, queried using the SPARQL language and organized into sites and containers. Long-term goals include developing or integrating wikis, knowledge bases, search engines, groupware, productivity tools, supply chain solutions, marketplaces and e-commerce solutions, social networks, smart contracts and DAOs. With NextGraph, users can now create and access freely their own interlinked data graphs, while preserving privacy, security, data locality, and interoperability.

CRDT KnowledgeGraph RDF SPARQL

More info: https://nlnet.nl/project/NextGraph

Website: nextgraph.org
Repository: git.nextgraph.org/NextGraph

NixOS/Clevis

Unattented disk decryption with Clevis on NixOS

Description

Whether they should or not, organisations are moving their data to third party servers (aka the " cloud"). While full disk encryption of servers should be an everywhere standard in order to protect the sensitive data that they inevitably hold, its adoption is still lagging. This isn't just lack of awareness, but also part of the tooling is missing. With full disk encryption comes a big pain point: restarting the server needs for the root file system to be unlocked before booting the OS.

While it is possible to remotely log into a server to unlock it remotely, this does create a dependency on a human operation in order to boot a server without compromising security. This is sometimes a non-acceptable drawback : it rules out unattended reboots, recovery from power loss, and it doesn't scale well with the number of servers.

This project will make on disk encryption with remote unlocking part of NixOS - bringing together a number of innovative mechanisms such as system extensions images and stage1-networkd. While this does not make using the cloud safe and private in and by itself (this is impossible), it will contribute to make it somewhat more safe and more private.

Additionally the project will port the Proxmox Hypervisor on NixOS, in order to benefit from NixOS-style declarative host configuration and deployment (which is very valuable when managing a cluster of machines to avoid configuration rot). ProxMox is a hypervisor that can run little to middle sized VM clusters and is capable of handling multi-node clusters.

FullDiskEncryption **Nix** **RemoteDecryption**

More info: https://nlnet.nl/project/NixOS-Clevis

Website: nixos.org
Repository: github.com/NixOS/nixpkgs/pull/257525

Securing NixOS services with systemd

Securing NixOS services with systemd

Description

NixOS, with the nix package manager, provides different services that can be installed and configured in a reproducible, declarative way. But how does one know whether software sticks to what it is supposed to do, and prevent a malicious application to spy on others?

Systemd provides users with ways to specify fine-grained sandboxing options for their running service, taking advantage of the Linux kernel's security facilities. This project will improve the default configuration of the services that are available in NixOS using systemd, so that users may deploy services without granting them too much trust: the services would only have access to the parts of the system they require. From a security point of view, this limits the attack surface of the system and improves a lot of defense in depth. This also means that services wouldn't be able to snoop on all of the user's system.

To gain long-term benefits from this project, we will develop automated tools to help with finding the right configuration for a given service, and we will write documentation to help people who will want to secure other services with their task.

Configuration ConfigurationManagement DeclarativeConfiguration

More info: https://nlnet.nl/project/NixOS-Services

Website: github.com/thejohncrafter/nixos-harden-systemd

NixOS

UEFI Secure Boot support for NixOS

Add a self-sovereign root of trust as part of supply chain security

Description

This project combines the power of the reproducible package manager Nix with the cryptographic protections of UEFI Secure Boot to provide concrete assurances about the authenticity of the software being booted into. Supply chain security works upward from a root of trust, which has to be in place before the very first bytes of code are even executed by a host's CPU. UEFI Secure Boot helps provide this root of trust. Using UEFI Secure Boot, the host's firmware will only boot the operating system if it is signed by a key stored in the firmware. This key may be issued by Microsoft, or in this project's case, be generated by the user. This can help resist attacks from malware or other attacks against the system's integrity. Obviously, when people use a commodity operating system commercially available to everyone (like Microsoft Windows) the security protection is far less and the risks are far greater than when someone generates a custom operating system with a reproducible tool like Nix. The Host and signing service will use TPM-backed attestation keys to mutually attest the authenticity of the requests.

This tool will initially support systemd-boot and uboot, however the project will be specifically designed with the intention of supporting additional bootloaders.

Bootloader **DigitalSignature** **TPM**

More info: https://nlnet.nl/project/NixOS-UEFI

Website: github.com/nix-community/lanzaboote

Type Inference for Nix

Adding static typing and type inference to Nix

Description

Nix is a tool to configure systems and manage packages. It comes with a programming language, also called Nix, to describe packages and configurations. Typically, when a change is made to the configuration of a system, the new configuration is evaluated and then applied. However, configuration errors are only reported after the failure of the evaluation. So, users often have to edit the configuration, evaluate it, understand the evaluation errors, fix the errors and try again. This feedback loop is very inefficient and frustrating for users. Similarly, developing the abstractions to make the Nix package collection (nixpkgs) work can be challenging. Indeed, dynamically typed languages with reflection like Nix do not provide many safeguards.

This project aims to retrofit a static type system, with type inference, on the existing Nix language while being backwards compatible with existing code. Types provide timely feedback to developers to help them during development, thanks to localized error messages. Furthermore, a type system for Nix would supercharge language server protocols and provide immediate feedback to Nix programmers. In addition to acting as some form of documentation, static types enable new exiting possibilities like better optimizations for Tvix in order to get faster evaluation and more advanced type-based function search with Noogle.

Nix StaticTyping TypedLanguage

More info: https://nlnet.nl/project/Nix-TypeInference

Website: github.com/NixOS/nix/issues/14

Adopting the Noise Key Exchange in Tox

Improved security of Tox instant messaging with NoiseIK

Description

Tox is a P2P instant messaging protocol that aims to provide secure messaging. It's implemented in a FOSS library called " c-toxcore" (GPLv3). The project started in the wake of Edward Snowden's disclosure of global surveillance. It's intended as an end-to-end encrypted and distributed Skype replacement. The cryptographic primitives for the key exchange (X25519), authentication (Poly1305) and symmetric encryption (XSalsa20) are state of the art peer-reviewed algorithms. Tox' authenticated key exchange (AKE) during Tox' handshake works, but it is a self-made cryptographic protocol and is known to be vulnerable to key compromise impersonation (KCI) attacks. This vulnerability enables an attacker, who compromised the static long-term private X25519 key of a Tox party Alice, to impersonate any other Tox party (with certain limitations) to Alice (reverse impersonation) and to perform Man-in-the-Middle attacks. The objective of this project is to implement a new KCI-resistant handshake based on NoiseIK in c-toxcore, which is backwards compatible to the current KCI-vulnerable handshake to enable interoperability. Further Noise's rekey feature will be evaluated for adoption.

AKE InstantMessaging KeyExchange NoiseIK P2P

More info: https://nlnet.nl/project/Noise-Tox

Website: tox.chat
Repository: github.com/TokTok/c-toxcore

Oil Shell

A new dialect of shell that is less error-prone

Description

Oil is a new Unix shell. Shell languages provide an (IEEE standardised) interactive command language and interactive scripting environment used to control computer operating systems. Shell scripts are deployed and used visibly and invisibly to command or glue together different applications and control the execution of tasks. Oil is the upgrade path from traditional shells like bash to a better and more structured language and runtime. It already runs thousands of lines of unmodified POSIX compliant shell scripts (as well as bash scripts which aren't compliant), but in a safer and more reliable way.

OSH can be smoothly upgraded to YSH, a new shell language influenced by Python, Ruby, JavaScript, JSON, and YAML. YSH also offers a basic interactive shell UI, and a " headless" API for building GUIs on top of shell. Through its set of specification languages, scripts can be translated to fast C++.

Compiler Scripting Shell

More info: https://nlnet.nl/project/OilShell

Website: www.oilshell.org
Repository: github.com/oilshell/oil

Oil Shell

Modern shell language and runtime

Description

Oil is a new Unix shell. Shell languages provide an (IEEE standardised) interactive command language and interactive scripting environment used to control computer operating systems. Shell scripts are deployed and used visibly and invisbly to command or glue together different applications and control the execution of tasks. Oil is the upgrade path from traditional shells like bash to a better and more structured language and runtime. It already runs thousands of lines of unmodified POSIX compliant shell scripts (as well as bash scripts which aren't compliant), but in a safer and more reliable way.

OSH can be smoothly upgraded to Oil, a new shell language influenced by Python, Ruby, JavaScript, JSON, and YAML. Oil also offers a basic interactive shell UI, and a " headless" API for building GUIs on top of shell. This project will finish the translation from statically typed Python to C++. This will let it match the speed of bash and existing shells, while offering reliable error handling, safe processing of user-supplied data, the elimination of quoting issues and better error messages and tools.

Shell

More info: https://nlnet.nl/project/OilShell-OSH

Website: www.oilshell.org
Repository: github.com/oilshell/oil

Improve Okular digital signature support

Improve open source tooling for digital signatures

Description

Okular is a Free Software document viewer that supports multiple file formats such as PDF and OpenDocument Format, and besides viewing allows for annotation and digital signatures. It was initially created for desktop Linux and UNIX operating systems but meanwhile has grown into a universal, vendor-neutral document tool for all platforms - including an increasing amount of mobile operating systems such as Android, postmarketOS and pureOS. Digital signatures allow people to establish the source of documents, but can also be used to enter into legally binding agreements or contracts - so having a reliable and transparent solution is important. The aim of this project is to improve the support of PDF digital signatures in Okular both from the point of view of features and usability, making it easier for users to interact with this crucial privacy and security functionality.

DigitalSignature PDF

More info: https://nlnet.nl/project/Okular

Website: okular.kde.org
Repository: invent.kde.org/graphics/okular

OpenCryptoHW

CGRA- based reconfigurable open-source cryptographic IP cores

Description

OpenCryptoHW aims to develop reconfigurable open-source cryptographic hardware IP cores for Next Generation Internet. With the Internet of Things (IoT) upon us, security and privacy are more important than ever. On the one hand, if the security and privacy features are exclusively implemented in software, the risk of breaches is high. On the other hand, if implemented solely in hardware, it is impossible to fix bugs or deploy critical updates, which is also a threat to security and privacy. Hence, we propose to use reconfigurable hardware, providing the flexibility of software and the trustworthiness of hardware. Hacking into it requires first hacking the device's configuration infrastructure and then hacking the algorithm itself, which is way more complicated. There have been proposals to implement cryptographic IP cores using Field Programmable Gate Array (FPGAs). However, the FPGA configuration infrastructure is cumbersome and proprietary, increasing device cost and compromising safety. Therefore, we propose to use open-source Coarse-Grained Reconfigurable Arrays (CGRAs) instead of FPGAs. CGRAs have much lighter configuration circuits and are not controlled by any private entity. With OpenCryptoHW, hardware and system designers will be able to download CGRA-based cryptography IP cores for free and under a permissive license, ready to integrate into their silicon designs.

CGRA OpenHardware

More info: https://nlnet.nl/project/OpenCryptoHW

Website: www.iobundle.com
Repository: github.com/IObundle/iob-soc-opencryptohw

OpenCryptoLinux

Make Linux run on OpenCryptoHW

Description

OpenCryptoLinux aims to develop an open, secure, and user-friendly SoC template capable of running the Linux operating system, with cryptography functions running on a RISC-V processor. The processor will control a low-cost Coarse-Grained Reconfigurable Arrays (CGRAS) for enhanced security, performance, and energy efficiency. Running Linux on this SoC allows non-hardware experts to use this platform, democratizing it. This project will help build an Internet of Things (IoT) that does not compromise security and privacy. The project will be fully open-source, which guarantees public scrutiny and quality. It will use other open-source solutions funded by the NLnet Foundation, such as the RISC-V processors from SpinalHDL and the OpenCryptoHW project.

CGRA RISC-V

More info: https://nlnet.nl/project/OpenCryptoLinux

Website: github.com/IObundle/iob-soc-opencryptolinux

OpenCryptoTester

System-on-Chip for hardware/software testing

Description

This project aims to develop a System-on-Chip (SoC) used mainly to verify cryptographic systems that improve internet security but can also be used on any SoC. It is synergetic with several other NGI Assure-funded open-source projects – notably OpenCryptoHW (Coarse-Grained Reconfigurable Array cryptographic hardware) and OpenCryptoLinux. The proposed SoC will support test instruments as peripherals and use OpenCryptoHW as the System Under Test (SUT), hopefully opening the way for open-source test instrumentation operated under Linux.

OpenHardware Testing

More info: https://nlnet.nl/project/OpenCryptoTester

Website: github.com/IObundle/iob-soc-tester

Open MLS Infrastructure

End-to-end encrypted group messaging

Description

The Open MLS infrastructure project aims at designing and implementing infrastructure components for the MLS (Messaging Layer Security) protocol currently under development by the IETF (https://datatracker.ietf.org/doc/draft-ietf-mls-protocol/). While it is theoretically possible to run MLS peer-to-peer, most use-cases will require central components that take care of ordering and queueing messages, as well as managing group state. Our goal is to create components that are secure, metadata-minimizing, modular, and that allow for federation. This lays a foundation for improving existing and future messaging applications, and will allow to validate a potential future application-layer specification.

E2EE IETF MLS

More info: https://nlnet.nl/project/OpenMLS-infra

Website: openmls.tech
Repository: github.com/phnx-im/infra

OpenPGP CA

Hardening OpenPGP CA deployments

HSM support for OpenPGP key infrastructure

Description

OpenPGP CA is a tool for managing and certifying OpenPGP keys in organizations. Today, the private key material of OpenPGP CA instances is stored and used locally. This project will add support for two hardened modes of operation: 1) Using a hardware-token OpenPGP Card) based key for the CA, and 2) Split OpenPGP CA deployments, in which critical operations are performed on a highly protected machine (e.g. air-gapped), while regular operation can take place conveniently on an online CA instance.

In addition the project will build an OpenPGP CA based tool for version control signing workflows (e.g. git), with a focus on providing a smooth user experience for signing with OpenPGP card devices.

Certificates **HSM** **OpenPGP** **Rust**

More info: https://nlnet.nl/project/OpenPGPCA-HSM

Website: openpgp-ca.org
Repository: gitlab.com/openpgp-ca/openpgp-ca

Sequoia-PGP

Improving OpenSSH's Authentication and PKI

Improving SSH Authentication with OpenPGP transitive trust

Description

It would not be a stretch to say that ssh secures the Internet - it is the protocol most relied on to log into servers of any type. Yet, its authentication model is inflexible, rarely used properly, and inadequate. OpenPGP’s transitive trust (aka " web of trust") mechanisms and revocation certificates can help to provided additional automated assurances. By publishing and certifying OpenPGP keys for servers, an ssh client may be able to automatically check whether an encrypted connection is not only encrypted, but also authenticated. Similarly, server administrators can automatically find the right public key for users. And when a server key or user key is compromised, using OpenPGP, it is straightforward to ensure that it won’t be trusted: just publish a revocation certificate. This project will add OpenPGP support to OpenSSH to improve and simplify these workflows.

OpenPGP SSL WebOfTrust

More info: https://nlnet.nl/project/OpenPGP-OpenSSH

Website: codeberg.org/wiktor/ssh-openpgp-auth

OpenQRNG

Open source, certified Quantum Random Number Generator

Description

Cryptography is key to protecting our modern secrets, and random numbers form the basis of the technical assurances given by that approach. However, true randomness is hard to achieve. Quantum number generators lever unpredictable physical phenomena to deliver quality randomness, and as such can be of great utility. However, currently there are only proprietary QRNG sources with a significant price tag - which means that the technology is not widely in use and that those people that do have the means have to essentially trust the vendor in question. The project will develop an open hardware QRNG device, which can be inspected from top to bottom - and made available at low cost.

OpenHardware RNG

More info: https://nlnet.nl/project/OpenQRNG

Website: http://www.openqrng.io
Repository: github.com/openqrng/QT1

In open hardware quantum random number generators we may trust
OPENQRNG
NGI ZERO

oqsprovider

Post-quantum/quantum-safe cryptographic algorithms for OpenSSL ●

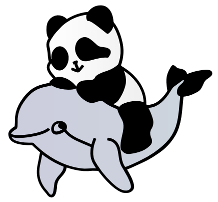
Description

Quantum computers will bring to an end integrity and confidentiality provided by " classic" public key cryptography such as RSA and implemented in security application frameworks such as OpenSSL. Therefore, a new class of " post-quantum" or quantum safe crypto algorithms (QSC) is being standardized by NIST. In order to bring QSC to easy deployment, these algorithms need to be added to existing security installations: oqs-provider is a standalone integration of QSC into the OpenSSL software framework. By simply inserting an oqs-provider binary, any OpenSSL installation as well as all applications built on top of OpenSSL permitting crypto-providers is (to be) automatically enabled to use any QSC algorithm supported by the liboqs open source framework. liboqs in turn provides the QSC algorithms that are either finalists or candidates of the NIST Post-Quantum Cryptography standardization competition. This way, users of oqs-provider-enabled OpenSSL installations can cease to be concerned about the risk that quantum computers create. The Open Source communities working on OpenSSL and OpenQuantumSafe can benefit in turn from mutual validation and re-use of their respective work efforts.

PostQuantumCrypto

More info: https://nlnet.nl/project/oqsprovider

Website: openquantumsafe.org
Repository: github.com/open-quantum-safe/liboqs



p2panda

p2p protocol and event-driven data store

Description

p2panda is a peer-to-peer protocol and framework for building local-first applications that store and exchange user data in a distributed database. p2panda's goal is to drastically extend the range of software projects that can be realized with a decentralised architecture by providing a wide range of features that alleviate common issues with this approach. A focus is set on data sovereignty, developer friendliness and supporting collaborative software.

This project will validate these claims by applying p2panda to a real-world use case and improve p2p networking by extending data replication capabilities.

CRDT **MLS**

More info: https://nlnet.nl/project/P2Panda

Website: p2panda.org
Repository: github.com/p2panda

NixOS

p4-nix

Combine Programming Protocol-independent Packet Processors language with declarative Nix packaging

Description

This project is aiming to democratize high capacity and high performance networking stacks by integrating the P4 DSL into Nix and making it easy to make an infrastructure relying on the technology by bringing up functional programming to the P4 world.

Bringing P4 to Nix gives us amazing flexibility for dealing with network devices, making it easy to deploy, make artifacts, and so on, all the while exposing it to end-users who wouldn't necessarily know or use P4 otherwise. This also gives us the opportunity to look into automated deployment of hardware based networking devices, such as FPGA targets, directly from within Nix.

P4 **PacketProcessing**

More info: https://nlnet.nl/project/p4-nix

Website: nixos.org

Peppol for the masses

Hybrid self-hosted e-invoicing with decentralized identities

Description

Peppol is an EU-backed e-Invoicing network which uses a top-down certification infrastructure to establish trust between the sender and the receiver of an invoice. In the " Peppol for the Masses!" project, we will implement Peppol in PHP (so far only Java and C# implementations are available), and package its core components (the AS4 sender and the AS4 receiver) as a Nextcloud app, so that users of the popular Nextcloud personal cloud server can send and receive invoices over AS4 directly into their self-hosted server.

Due to the top-down nature of Peppol's trust infrastructure, it's not possible to self-host a node in the Peppol network unless you go through a reasonably heavy certification process. Therefore, we will extend our implementation with support for self-hosted identities, using the " WebID" identity pattern which was popularized by the Solid project. We will also develop a re-signing gateway which replaces the signature on an AS4-Direct invoice with a Peppol-certified signature. In a follow-up project, we will also host an instance of this re-signing gateway and make it available free of charge, similar to how the LetsEncrypt project has made TLS certificates available free of charge.

This project will lower the (cost) barrier for machine-readable cryptographically-signed e-Invoicing messages, and at the same time increase the sovereignty of end-users, towards a human-centric internet of business documents.

Ledger

More info: https://nlnet.nl/project/Peppol-Decentralised

Website: github.com/pondersource/peppol-php

Adding Web-of-Trust Support to PGPainless

Web-of-Trust specification support for Java

Description

Reliable authentication of public key certificates is a hard requirement for strong and effective end-to-end encryption. The " Web-of-Trust" (WoT) serves as an example of a decentralized authentication mechanism for OpenPGP. While there are some existing implementations of the WoT in applications such as GnuPG, their algorithms are often poorly documented. As a result, WoT support in client applications is often missing or inadequate.

PGPainless is an easy-to-use, secure-by-default OpenPGP library for Java and Android. This project will extend PGPainless with an implementation of a recently published, new Web of Trust specification. The goal is to make the Web of Trust more interoperable and accessible to client applications, overall increasing the usability and ergonomics of OpenPGP for the end-user.

Library OpenPGP Web-Of-Trust

More info: https://nlnet.nl/project/PGPainless

Website: pgpainless.org
Repository: github.com/pgpainless/pgpainless

Post-Quantum Crypto in DNSSEC

Experimental platform for DNSSEC with post-quantum cryptography ●

Description

PQ-DNSSEC is an open-source tool set for exploring DNSSEC based on post-quantum cryptography. It includes implementations of authoritative DNS servers and DNS resolvers that support various post-quantum signature schemes as well as tools to evaluate performance and the compatibility of these implementations with the existing DNS infrastructure in the global Internet. PQ-DNSSEC also provides a collection of example zones to the general public. This way, the project will help the DNS community to prepare for transitioning to post-quantum secure DNSSEC.

DNSSEC **PostQuantumCrypto**

More info: https://nlnet.nl/project/PQ-DNSSEC-Testbench

Website: pq-dnssec.dedyn.io
Repository: github.com/desec-io

Probabilistic NAT Traversal

Last resort ad hoc connections for GNUnet

Description

With the Probabilistic NAT Traversal project, we want to significantly improve the ability of users to directly connect with each other. For establishing a peer to peer (p2p) network among regular internet users, unhindered connectivity is anything but self-evident. Today consumer devices are often not directly reachable via the internet but quite often are behind a so called NAT delivering only indirect internet connectivity. There are several methods to reach peers who are behind a NAT, but there are as many reasons those existing methods might fail. Manual configuration for example, as it is possible for example with home routers, often does not work for mobile devices like mobile phones. We will implement a new way of NAT traversal that we think of being independent from the existing network configuration, and does not require a third party with a direct internet connection helping two peers to connect to each other. Existing NAT traversal methods using third parties which are permanently required for communication. Our Probabilistic NAT traversal method does require some third party only at the beginning of the communication. The selection of third parties to start the connection establishment is based on previous work from the Layer-2-Overlay project. Probabilistic NAT Traversal will greatly improve the connectivity of GNUnet and other P2P networks that adopt it.

CCBV_Testing CCBV_testing HolePunching NAT

More info: https://nlnet.nl/project/ProbabilisticNAT

Website: www.gnunet.org/en/probnat

Prosody IM

Implement SASL authentication mechanism for XMPP ●

Description

XMPP is the most widely deployed standard protocol for real-time messaging today, and is a very popular choice among individuals and organizations who wish to manage their own internet communications, instead of submitting to other (e.g. commercial/data-driven) communication platforms. For an XMPP user to log in to their account today, two things are required: a username and a password. This has remained unchanged for many years, while other technologies have been steadily advancing to support security-enhancing features such as multi-factor authentication or even self-sovereign identities.

XMPP uses an authentication umbrella standard known as SASL to authenticate all connections.The way XMPP integrates SASL is defined in RFC 6120 and assumes a very simple challenge-response flow, which has worked well in allowing us to upgrade the network from older SASL mechanisms such as DIGEST-MD5 and onto more modern mechanisms such as SCRAM-SHA-1 and SCRAM-SHA-256.

To gain new authentication features beyond simple password authentication, we need to evolve XMPP's relationship with SASL. This project will deliver just that, and will be the first complete implementation of a proposed standard (XEP-0388: Extensible SASL Profile) into the popular Prosody XMPP server. It will also implement support for per-session access control throughout Prosody, and support for XEP-0386 (Bind 2.0).

Authentication **SASL** **XMPP**

More info: **https://nlnet.nl/project/Prosody-SASL**

Website: prosody.im
Repository: hg.prosody.im

ProveThis

Prove statements about authenticated API resources

Description

ProveThis allows users to prove statements from websites and APIs using TLS without revealing private information. Although efforts like TLSNotary can currently be used to prove the authenticity and origin of a full HTML page, we extend the capabilities of TLSNotary and allow users to make zk-SNARK based zero knowledge proofs about statements in complexity class NP. More concretely, this can allow users to prove statements about e.g. their banking data (how many transactions did you send in a certain period), social media data (how many friends are you away from knowing Barack Obama) or other data sources. Such proofs can generally be used to reduce fraud without compromising privacy and confidentiality.

Notarisation ZeroKnowledgeProof

More info: https://nlnet.nl/project/ProveThis

Website: github.com/summitto/ProveThis

Statime

Memory-safe high-precision clock synchronization

Description

Of all severe software security bugs, a big chunk (50-70%) has one single source: memory corruption. The underlying cause is that, traditionally, systems software is implemented in languages that are not memory-safe. The way forward is to replace these pieces of software with memory-safe alternatives, one by one. Doing so will not just mitigate, but eliminate this category of bugs entirely. This project picks out one piece: the Precision Time Protocol (PTP). High-precision clock synchronization plays a crucial role in networking, with application areas such as high precision localization, finance, broadcasting, security protocols, smart grids, and cellular base station transmissions. Our proof-of-concept implementation will conform to the IEEE standard for PTP and will focus on the software implementation of a slave-only PTP ordinary clock. In the future, our work is expected to become part of a wider open-source roadmap for reliable and memory-safe keeping of network time, that will seek to expand the feature set of our implementation and work towards growing its adoption.

Statime is part of Project Pendulum.

IEEE PTP TimeSynchonisation

More info: https://nlnet.nl/project/PTP-Rust

Website: tweedegolf.nl/en/pendulum
Repository: github.com/pendulum-project/statime

PyCM

Evaluate the performance of ML algorithms

Description

The outputs and results of machine learning algorithms are usually in the form of confusion matrices. PyCM is an open source python library for evaluating, quantifying, and reporting the results of machine learning algorithms systematically. PyCM provides a wide range of confusion matrix evaluation metrics to process and evaluate the performance of machine learning algorithms comprehensively. This open source library allows users to compare different algorithms in order to determine the optimal one based on their preferences and priorities. In addition, the evaluation can be reported in different formats. PyCM has been widely used as a standard and reliable post-processing tool in the most reputed open-source AI projects like TensorFlow similary, Google's scaaml, torchbearer, and CLaF.

AI

More info: https://nlnet.nl/project/PyCM

Website: www.pycm.io
Repository: github.com/sepandhaghighi/pycm

R5N-DHT

Formalisation within IETF of R5N Distributed Hash Table design

Description

Decentralization and digital sovereignty are fundamental building blocks to strengthening European values of freedom of information and informational self-determination against particular interests of foreign state and commercial actors. Decentralization is often based on Distributed Hash Tables; DHTs are already an important component for many NGI components such as decentralized web applications (IPFS, Web3) or components in the blockchain ecosystem. The GNUnet/R5N-DHT - a Free Software distributed hash table and P2P protocol - provides additional and relevant properties like Byzantine fault tolerance and censorship resistance. The project will improve, implement and specify the R5N protocol as an IETF RFC (Informational). This supports other efforts such as the GNU Name System protocol (GNS).

ByzantineFaultTolerance CensorshipResistance DHT IETF Interoperability Library StandardSetting

More info: https://nlnet.nl/project/R5N-DHT

Website: gnunet.org
Repository: git.gnunet.org

rasn

Safe ASN.1 codec framework for Rust

Description

ASN.1 is a suite of protocols and data formats first introduced nearly 40 years ago, and is used extensively throughout the industry, from SIM cards to satellites, from web certificates to 5G radios, all of these are using ASN.1 in their communication stack. However parsing ASN.1 remains a large source of security vulnerabilities due its complexity and needing to be written in traditionally memory unsafe languages for speed and portability.

Rasn is a codec framework for writing safe ASN.1 code in Rust, that encodes ASN.1's data model into Rust’s type system, empowering developers to write Rust code that is as safe, portable, and as easy to write as the original ASN.1 module. Rasn supports BER, CER, and DER encoding rules, and can be extended to support custom data formats. Rasn also provides a number standards out of the box including LDAP, PKIX, and SNMP.

ASN.1 Parser Rust

More info: https://nlnet.nl/project/RASN

Website: github.com/librasn/rasn

Rosenpass

Post Quantum Security Add-On for WireGuard

Description

Rosenpass is a formally verified, post-quantum secure VPN that uses WireGuard to transport the actual data. The implementation does not create a VPN connection itself, instead it performs a key exchange and hands this key to WireGuard; i.e. it **enhances** WireGuard's security without replacing it. This reduces the complexity of implementing the protocol and ensures that all the performance-advantages of WireGuard are available with Rosenpass. There is some extra latency to make a connection, but after that, WireGuard and Rosenpass are as fast.

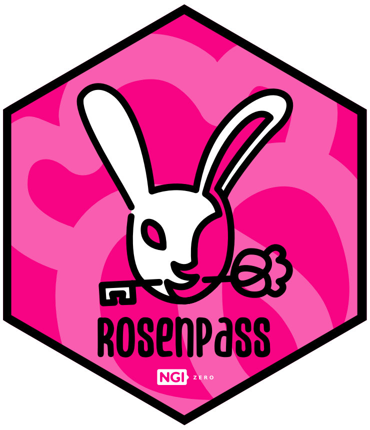
The protocol used by Rosenpass is based on the handshake designed by Hülsing, Ning, Schwabe, Weber and Zimmermann and improves upon the protocol by using cookies to provide resistance against state-disruption attacks. State-disruption attacks exist against the first version of the post-quantum WireGuard protocol and against classic WireGuard when NTP is used to synchronize the system-clock.

Internally, the protocol uses two post-quantum KEMs (key exchange methods) and no post-quantum signature schemes to provide ephemeral secrecy and deniability.

KeyExchange **McEliece** **PostQuantumCrypto** **VPN**

More info: https://nlnet.nl/project/Rosenpass

Website: rosenpass.eu
Repository: github.com/rosenpass/rosenpass



Rosenpass

What does this project mean to users?

Today, Virtual Private Networks (VPNs) are a cornerstone of the modern Internet. When you go online outside of your house or office, for instance on a public wifi spot in your favourite restaurant - your connection can be vulnerable to so called man-in-the-middle attacks. Instead of the hotspot connecting you to the internet as you would assume to be the case, someone operating the network you use to access the internet can tamper with your traffic.

Obviously, that can have disastrous results in terms of security. In professional context, VPNs are therefore meanwhile everywhere. Whenever you connect to your workplace from your #sym.quote.double home office#sym.quote.double , you most probably already, consciously or not, use VPN software to ensure that all data flowing from your computer at home to your employer's office are safe from being tampered with. There are many more common, daily use cases – from online research to increasing privacy to bypassing network misconfigurations and other disturbances.

In recent years, we have seen the rise of "Quantum Computers"– a new class of specialised computers that operate in a fundamentally different way from how traditional computers operate. While practical utility of such Quantum Computers for most day to day usage would for now still be limited (this will change in the future, no doubt), and the earliest computers of this type are prohibitively expensive (so not many organisations can afford one), they are known to do at least one thing particularly well: solving the kind of mathematical challenges on which many cryptographic standards are based.

The increasing availability and equally growing capabilities of Quantum Computers means traditional cryptography has to be phased out. While some of the most widely used cryptography is still considered safe on conventional computers for now, it is no longer something you can trust on for protecting confidentiality - and this will get worse and worse once Quantum Computers become more powerful and more widely available. Of course this not only impacts the safety of our online banking or the authenticity of a website pretending - it also impacts VPNs which are essentially encrypted tunnels across the internet.

Rosenpass is an important practical countermeasure. You use Rosenpass in conjunction with the widely used WireGuard technology (which is a.o. part of the Linux kernel). Because it uses specific Post Quantum Secure (PQS) cryptography, based on the McEliece cryptosystem, it is expected to withstand Quantum Computer attacks. Rosenpass doesn't change the way WireGuard works (in fact, WireGuard encryption continue to work as it used to without Rosenguard). It does provide a post-quantum-secure key exchange in the spirit of the Noise protocol used by many of todays instant messaging solutions like Matrix, Signal, WhatsApp and XMPP.

Rosenpass API

Improved API's and platform coverage for Rosenpass

Description

Rospenpass deals with post-quantum security for the open-source, linux-kernel VPN WireGuard. It is a production-ready VPN solution, with security proofs and backed up by scientific papers. This solves the problem that classic WG alone will stop being secure once quantum computers are viable.

In this phase of the work, we focus on enhancements to support Rosenpass on additional platforms by providing initial support for Windows. Improvements to the Rosenpass protocol protect our key exchange against denial-of-service attacks by integrating WireGuard's cookie-based mechanism. To introduce more granularity with regard to system permissions required by the Rosenpass client, a broker-based architecture is being introduced. Achieving this goal entails creating a Unix sockets API infrastructure, API endpoints, and a special broker process to handle communication with WireGuard. Finally, the work also aims to promote scientific communication and research on post-quantum cryptography by creating scientific illustrations, and by authoring a user tutorial on using Rosenpass to secure TLS connections.

CrossPlatform PostQuantumCrypto SymbolicProof VPN

More info: https://nlnet.nl/project/Rosenpass-API

Website: rosenpass.eu
Repository: github.com/rosenpass/rosenpass

Subliminal Messaging

Embedded secure channels within traditional and internet telephony

Description

Most of todays telephony consists of digital transmissions, so given a codec without mangling or added noise, it becomes possible to treat (part of) that as a data channel, and pass meaningful data through it while maintaining an acceptable noise floor to the sound being transmitted. That data channel can give rise to information exchange, including key material and alternative contact options.

The project will work on various improvements that connect telephony and digital communication: (1) VPN setup with telephony protocols, (2) data communication over the PSTN backbone and its extensions into VoIP, (3) digital security for PSTN and VoIP calls.

Encryption **VoiP**

More info: https://nlnet.nl/project/SecureDataChannel

Website: gitlab.com/0cpm/subliminal

Sequoia-PGP

Sequoia PGP

Improve interface of Sequoia PGP commandline

Description

Sequoia PGP is a new OpenPGP implementation, which is written in Rust and focuses on ease of use. To date, the main product is a library. This project will focus on sq, Sequoia's command line tool. The project consists of three parts. First, useful functionality will be added to sq making sq comparable to gpg. Second, the human-readable interface will be augmented with a JSON interface. This will make it easier and robuster to use sq from scripts. Finally, this project will add an acceptance test suite to sq thereby strengthen the foundation for future changes.

OpenPGP WebOfTrust

More info: https://nlnet.nl/project/Sequoia-commandline

Website: sequoia-pgp.org
Repository: gitlab.com/sequoia-pgp/sequoia

A Secret Key Store for Sequoia PGP

Standards-compliant private key store for OpenPGP

Description

This project implements a private key store for Sequoia, a new OpenPGP implementation. Currently, Sequoia-using programs use private keys directly. A private key store mediates applications' access to private keys, and offers three major advantages relative to the status quo. First, a private key store is in a separate address space. This means that private keys that are in memory are in a different address space from the application. This was underlying cause of the Heartbleed vulnerability. Second, a private key store can provide a uniform interface for accessing keys stored on different backends, e.g., an in-memory key, a key on a smart card, or a key on a remote computer, which is accessed via ssh. This simplifies applications. Third, this architecture simplifies sharing private key material among multiple applications. Only the private key store needs to worry about managing the private key material, which improves security. And, when a user unlocks a key in one application, it is potentially unlocked in all applications, which improves usability.

KeyStorage OpenPGP

More info: https://nlnet.nl/project/Sequoia-Keystore

Website: gitlab.com/sequoia-pgp/sequoia-keystore

Sequoia GPG Chameleon

Implement well-known API's for using OpenPGP

Description

Sequoia's GnuPG Chameleon is a drop-in replacement for the widely-used encryption software GnuPG. It offers the same interface, while at the same time replacing the underlying OpenPGP implementation. This approach brings security benefits to everyone directly or indirectly using GnuPG before, while providing a smooth migration path that does not require changes to existing software.

OpenPGP WebOfTrust

More info: https://nlnet.nl/project/SequoiaChameleon

Website: sequoia-pgp.org
Repository: gitlab.com/sequoia-pgp/sequoia-chameleon-gnupg

Sequoia-PGP

Adding TPM Support to Sequoia PGP

Implement use of TPM 2.0 crypto hardware for OpenPGP

Description

Protecting cryptographic keys is hard. If they are stored in a file, an attacker can exfiltrate them - even if the harddrive is encrypted at rest. A good practical solution is a hardware token like a Nitrokey, which stores keys and exposes a limited API to the host. For most end users, a token is a hassle: one needs to carry it around, it needs to be inserted, and it is not possible to work if it is left at home. And, it needs to be purchased. There is a better solution, which doesn't cost anything. A trusted computing module (TPM) is like an always-connected hardware token only more powerful (the keys can be bound to a particular OS installation, it can store nearly an unlimited number of keys, not just three) and TPMs are already present in most computers. This project will add support for TPMs to Sequoia PGP including comprehensive test suites and in-depth documentation for both software engineers: as an API and end-users as a way to use TPM bound keys through Sequoia's command-line interface (sq) for decryption and signing.

HardwareIsolation OpenPGP TPM

More info: https://nlnet.nl/project/Sequoia-TPM

Website: wiktor.gitlab.io/tpm-openpgp

Servo Developer Experience Improvements

Improve productivity for Servo developers

Description

Servo is a cross-platform, open-source browser engine that next-generation browsers can be built on, including the Verso browser project. However, the current developer experience is lacking in some ways, including CI/CD, benchmarks, and documentation for integration in downstream projects. While the Servo project these things currently, ongoing maintenance to keep them up to date, as well as creation of new documentation and tutorials to aid newcomers to the project, is a task that always needs work. In order to make integration with Servo easier for both the Verso project, as well as new projects that want to use it, this project aims to bring modern enhancements and new content to these areas.

DeveloperTool

More info: https://nlnet.nl/project/Servo-DX

Website: versotile.org/verso
Repository: github.com/versotile-org/verso

Multi browsing context support in Servo

Allow Servo browser engine to render beyond atomic pages

Description

Verso is a browser application based on the Servo web engine. We want to build a new web browser using a different set of technical stacks than existing browsers. Hope it can improve the codebase of browser programming and grow the ecosystem along with it. In order to build an application around Servo, we need to implement several key features with it since Servo is merely a web engine and it doesn't control anything else outside of its own context. One of the challenges is supporting multiple browsing contexts all at the same time. So we can composite all web views into one single window to make it present as an ordinary application. We will need to improve the compositor of Servo to make it support multiview, and also implement the ergonomic interface in Verso for different purposes. It will be able to render not only web pages, but also UI panels, context menus, prompts, and more.

Browser Headless Rendering Webview

More info: https://nlnet.nl/project/Servo-Multibrowsing

Website: github.com/versotile-org/verso

SES - SimplyEdit Spaces

SimplyEdit Spaces - collaborative presentations

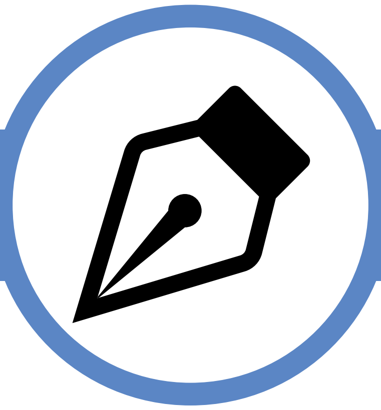
Description

SimplyPresent allows users to collaboratively create and deliver good looking presentation using CRDT's through Hyper Hyper Space - another project supported by NGI Assure. SimplyPresent is itself based on top of the open source SimplyEdit tool, adding advanced user-friendly presentation features. SimplyPresent allows team members to live edit a presentation and the presenter notes while the presentation is being given, control the presentation from any phone without complicated setup: all that is needed on the presenting system or with remote viewers is a URL which will sync through Hyper Hyper Space.

CRDT RealTimeCollaboration

More info: https://nlnet.nl/project/SES

Website: simplyedit.io
Repository: github.com/SimplyEdit/SimplyPresent



Signature PDF

Self-hosted tool to add signature to PDFs

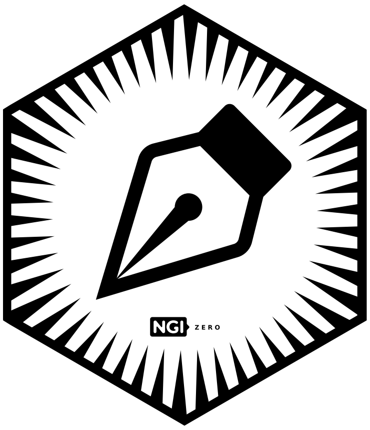
Description

PDF Signature is a free software (FLOSS) for online signing of PDF. Users can add signature, stamp, text or check marks individually, or collectively with the shared mode. The tool aims to be a free alternative to existing proprietary web services, in order to offer users more control and guarantee of what happens to the PDF processed by the software. It is easily deployable on a server, a personal machine, a nano-computer , a container image or a Yunohost instance. The future developments of this project will improve the confidentiality by encrypting the pdf stored on the server, study and improve the compatibility with the electronic signature standards (XAdEs, PAdES), internationalize the interface and add integration with Nextcloud.

PDF **Signature**

More info: https://nlnet.nl/project/SignaturePDF

Website: pdf.24eme.fr
Repository: github.com/24eme/signaturepdf



SignRoom

Zenroom based signature and credential platform

Description

Leveraging the quantum-proof cryptographic implementation done in Zenroom (along with Zenroom's other cryptographic flows) we are developing a simple to use web-based platform, allowing users to sign and verify messages and documents (PDF, Office files, pictures etc) using quantum proof signature, ecdsa signature and schnorr signature and multi-signatures. Document signatures are stored inside the document using the PADES and XADES protocols. The tool will also produce and verify zero-knowledge proof credentials, W3C-VC credentials for signature and verification. The platform is built as a PWA, is mobile friendly, has APIs for third party integration a library to integrate into mobile applications along with bindings for multiple programming languages.

DigitalSignature PostQuantumCrypto

More info: https://nlnet.nl/project/Signroom

Website: http://dyne.org

smoltcp RPL

Implement Routing Protocol for Low-Power and Lossy networks

Description

Smoltcp is a TCP/IP library written in the Rust programming language. The Rust language offers many advantages, such as memory safety. The smoltcp library recently gained support for the 6LoWPAN protocol, enabling IPv6 for IEEE802.15.4 devices. However, a routing protocol tailored for low power devices is still missing in the library (or even one written in the Rust programming language). In this project, an implementation of the Routing Protocol for Low-Power and Lossy Networks (RPL) will be added to the smoltcp library. This protocol is designed for Low-Power wireless networks that are generally susceptible to packet loss. By adding this protocol to smoltcp, we get closer to a network stack that is safer to use for the Internet of Things (IoT).

EmbeddedSystems Energy

More info: https://nlnet.nl/project/Smoltcp

Website: thvdveld.be/smoltcp-rpl-docs/introduction.html
Repository: github.com/smoltcp-rs/smoltcp

Peer-to-Peer Access to Our Software Heritage

Access Software Heritage data via IPFS DHT

Description

Peer-to-Peer Access to Our Software Heritage (SWH × IPFS) is a project aimed at supporting Software Heritage's mission to build a universal source code archive and preserve it for future generations by leveraging IPFS's capabilities to share and replicate the archive inadecentralized, peer-to-peer manner. The project will build a bridge between the existing Software Heritage (SWH) API and the IPFS network to transparently serve native IPFS requests for SWH data. In the short term, this allows users using IPFS to form their own Content Distribution Network for SWH data. Longer term, we hope this will serve as a foundation fora decentralized network of copies that, together, ensure that the loss of no one repository, however large, results in the permanent destruction of any part of our heritage. The end product would be a perfect application of IPFS's tools and a step in the direction of a decentralized internet services infrastructure.

Archiving DHT IPFS P2P Resilience

More info: https://nlnet.nl/project/SoftwareHeritage-P2P

Website: github.com/obsidiansystems/go-ipfs-swh-plugin

Solid Wallet

Authorization reasoning, rule-based controls and fluid integration for Solid

Description

Solid Apps display information collected by following linked data across the World Wide Web, writing changes to Solid Personal Online Data Stores (PODs). Following links can land an App on a protected resource somewhere on the Web, accessible only to a select group of actors specified in an associated Web Access Control Resource. Solid Wallet aims to build core libraries to reason over Solid Access Control Rules, limit access to what clients can request, publish keys and sign transactions. The same libraries will also be useable by servers to verify such claims. Finally, we will use these libraries to build a flexible prototype Wallet for Solid apps that run in the browser or server.

Cryptowallet **Solid** **Wallet**

More info: https://nlnet.nl/project/SolidWallet

Website: github.com/co-operating-systems/solid-control

Spritely (and OCapN)

Enable secure P2P applications with Object Capabilities

Description

OCapN (the Object Capability Network, and featuring CapTP, the Capability Transport Protocol) simplifies building otherwise complicated security-oriented peer to peer systems as a natural extension of ordinary programming patterns. OCapN/CapTP features intentional collaboration amongst networked objects, distributed garbage collection, networked promise pipelining for efficient distributed communication, a peer introduction and consensual resource sharing system, and an abstract networking layer compatible with Tor Onion Services, I2P, libp2p, and even more traditional DNS + TLS.

While multiple implementations exist within Spritely and elsewhere, these are all incompatible. The project will produce specifications, documentation, and test suites to encourage consistency, interoperability, and smooth adoption of the technology.

ObjectCapability

More info: https://nlnet.nl/project/SpritelyOCapN

Website: spritelyproject.org
Repository: github.com/ocapn/ocapn

Statime PTP Master

Statime - Zero-allocation cross-platform Precision Time Protocol

Description

High-precision clock synchronization is becoming increasingly important in application areas such as high precision localization, finance, broadcasting, security protocols, smart grids, and cellular base station transmissions. The Precision Time Protocol (PTP) is widely used for these critical applications and it is therefore important for it to be as secure and reliable as possible.

We have previously developed the first iteration of Statime, an implementation of a PTP slave in the Rust programming language. The outcome of that project is a secure-by-design implementation, leveraging the Rust borrow checker to guarantee memory-safety. With this project, we will expand our implementation in two ways. Firstly, we will expand the feature set to include a PTP master, conforming to the IEEE standard for PTP (the 2019 version, IEEE1588-2019), so we can run a full PTP instance with the memory-safety guarantees that our implementation provides.

Secondly, our implementation will be able to run without an operating system or system allocator. Those properties make the implementation inherently portable and more reliable. Our concrete goal for this second phase is that it runs on the stm32f7 microcontroller, a device with built-in PTP Ethernet support, but otherwise limited capabilities.

IEEE PTP TimeSynchonisation Timecoding

More info: https://nlnet.nl/project/Statime-PTP-Master

Website: github.com/pendulum-project/statime

Sustainable web apps with m-ld

Empower users and developers with distributed interlinked data using local-first principles

Description

Our hypothesis in this project is that web app data securely stored in reactive, replicated Linked Data sets can make it possible for app developers to meet today's and tomorrow's feature expectations without the high costs and limitations of today's distributed data architectures. This foundational design principle combines ideas from the semantic web (machine-readable publishable interlinked data), personal data stores (user control of user data) and local-first software (collaboration without obligatory third parties).

We believe the high costs of web app development have gone hand-in-hand with unwanted side-effects like user lock-in, attention theft, and abdication of control over personal data. Our core principle, like the ideas behind them, is designed to expedite the development of more sustainable apps: those without dependencies on specific service providers, with user empowerment in terms of service and data portability, and with linking of data between apps – including apps developed against similar technologies having these principles, such as those of the Solid ecosystem.

We will produce a set of concrete software components which demonstrate that such an approach is practical, and indeed offers a great experience for app developers, making it simple to create collaborative applications over Linked Data resources with compelling, responsive user interfaces.

CRDT CollaborativeEditing RealTimeCollaboration

More info: https://nlnet.nl/project/SustainableWebApps

Website: m-ld.org

Tahoe-LAFS

Great Black Swamp

Decentralized cloud storage with provider-independent security

Description

Tahoe-LAFS is a well-known open source distributed storage solution based on DHT, suited for sharing critical data in production. Currently, Tahoe-LAFS uses the Foolscap protocol for communication between client nodes and storage nodes. Foolscap has a small developer community, is only implemented in Python, and Tahoe-LAFS only uses a small subset of its features. This project will implement an HTTP-based storage node protocol for Tahoe-LAFS (Great Black Swamp, or GBS in short) which will help to eliminate unnecessary complexity, increase the pool of potential contributors, open the door to new implementations and improve runtime performance.

DHT DataStorage DistributedFileSystem Encryption

More info: https://nlnet.nl/project/TahoeLAFS-GBS

Website: tahoe-lafs.org
Repository: tahoe-lafs.org/trac/tahoe-lafs/browser

Tauri Apps

A safer run-time for web technology based apps

Description

Tauri is a toolkit that helps developers make more trustworthy applications for the major desktop platforms - using virtually any frontend framework in existence. A popular use case is to create a desktop or mobile version of a web app, rather than wasting effort on creating native clients for each platform. Unlike other solutions (e.g. Microsoft's Electron), it is built in the type-safe language Rust - and the team has a focus on strong isolation, shielding the user from malicious or untrusted code downloaded " live" from the internet. After all, once breached, such an app can for instance siphon off cryptocurrencies or bootstrap other more persistent malware.

In this project, the team works among others on a particularly innovative feature, to prevent JS injection for all application types. In this approach Rust Code Injection is used alongside dependency-free EcmaScript, Object.freeze(), and a filtering iFrame that is the only subsystem permitted to communicate with the API. This will help to create more secure applications,

Isolation TypedLanguage Webview

More info: https://nlnet.nl/project/Tauri

Website: tauri.app
Repository: github.com/tauri-apps/tauri

Servo Webview for Tauri

Integrated portable webview based on Servo engine into Tauri

Description

The web ecosystem lacks a cross-platform, non-corporate controlled system for running web content. Tauri is a system for distributing cross-platform applications that relies on engines present on a system - effectively those owned by Apple, Google, and Microsoft. These permit varying levels of user control. The Servo project is a cross-platform, open source web engine.

While Servo's support for web features such as CSS and JS is still incomplete (making it difficult to rely on it for running arbitrary web content) it is actually a great match for Tauri already. This project would incorporate Servo into the Tauri project, enabling it to run applications in a consistent, open source web runtime on major desktop and mobile platforms. In doing so, the project would also identify and address the highest priority web compatibility issues in Servo, while preparing a roadmap for significant compatibility issues that remain unaddressed. Additionally, the project would identify any opportunities for reducing the binary size, supporting broad distribution of Tauri apps to as many users as possible.

Browser **Webview**

More info: https://nlnet.nl/project/Tauri-Servo

Website: github.com/versotile-org/verso

TerosHDL

Assisting hardware developers to deliver safer designs

Description

TerosHDL is an open source IDE for FPGA/ASIC development. It includes a backend, a front-end built on VSCodium/VSCode and a command line interface. The goal of TerosHDL is make the ASIC/FPGA development easier and reliable: to reduce the adaptation time for new users of HW languages and help professionals.

TerosHDL is multi-platform (Linux, Windows, MacOS), multi language (VHDL, Verilog, SystemVerilog) and it takes advantages of a lot of open hardware projects (such as Edalize, WaveDrom, VUnit…), integrating them in a common graphical user interface. The IDE tries to be as much self-contained as possible and simplify the installation process.

Some of the features are: linter, go to definition, syntax highlighting, code formatting, snippets, automatic documentation, dependencies viewer, simulators support…

IDE **QualityAssurance**

More info: https://nlnet.nl/project/TerosHDL

Website: terostechnology.github.io

FIDO 2.2

Open hardware implementation of FIDO CTAP 2.2

Description

WebAuthn in conjunction with FIDO2 is the latest standard for secure and convenient authentication in the Web. The Trussed framework's fido-authenticator is the main open source implementation of a FIDO2 security key and used by Solokeys and Nitrokey. It currently supports FIDO 2.0 and partially 2.1. This project will bring the fido-authenticator to its next stage by fully implementing the upcoming 2.2 standard among appropriate software tests, a hardware-in-loop test suite. The implementation will be confirmed by an official FIDO L1 certification.

CTAP Fido WebAuthn

More info: https://nlnet.nl/project/Trussed-FIDO2.2

Website: github.com/Nitrokey/fido-authenticator

TrustING

Ultrafast AS-level Public-Key Infrastructure

Description

TrustING is a human-transparent and agile Trust Infrastructure for a Next-Generation Internet. This infrastructure enables any two entities to establish secret keys that can be used to encrypt and authenticate data. The foundation of TrustING is the AS-level Public-Key Infrastructure (PKI) of the SCION Internet Architecture that provides sovereignty (ensuring absence of global kill switches), trust transparency, and algorithm agility, among others.

The TrustING service establishes symmetric keys with other domains in advance, and then relies on those keys to derive keys for local hosts. The core novelty of this approach is the ability to derive keys purely locally on both sides of the communication, without even requiring key transport. By making TrustING a control-plane mechanism offered by the network infrastructure, higher-level applications can make use of it without having to worry about complexities such as exchanging key material or establishing trust.

To show the viability of TrustING, we will implement TLS trust bootstrapping using TrustING and additionally demonstrate the efficiency of TrustING by using it to authenticate SCMP (SCION's equivalent of ICMP) messages.

PKI SCION TrustFlexibility

More info: https://nlnet.nl/project/TrustING

Website: scion-architecture.net
Repository: github.com/scionproto/scion

Trustix

Make build logs available as publicly verifiable, tamper-proof Merkle trees

Description

Software build infrastructure is vastly underestimated in terms of its potential security impact. When we install a computer program, we usually trust downloaded software binaries. But even in the case of open source software: how do we know that we aren't installing something malicious which is different from the source code we are looking at - for instance to put us in a botnet or siphon away cryptocurrencies? Typically, we have confidence in the binaries we install because we get them from a trusted provider. But once the provider itself is compromised, the binaries can be anything. This makes depending on 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 decentralizes 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.

In this project the team will work on further enabling trust delegation, by offloading log verification to trusted third parties - heavily inspired by the Delegated Proof of Stake consensus algorithm. It will bring Trustix into the Nix and the Guix ecosystems that are most amenable to Trustix' approach. The ultimate goal is for Trustix to integrate seamlessly into the entirely decentralized software supply chain so we can securely distribute software without any central corruptible entity.

Blockchain Reproducibility ReproducibleBuilds SupplyChain Transparency

More info: https://nlnet.nl/project/Trustix-Nix

Website: build-transparency.org
Repository: github.com/nix-community/trustix

Trust semantic learning and monitoring

Measure on-going trust between interacting agents

Description

Trust semantic learning and monitoring is part of a wide ranging effort to understand trust in network socio-technical systems. The expected outcome of this part is a methodology and proof of concept code library for qualifying and quantifying trust between agents in a network. In IT, trust is often treated as a binary " crypto token" , based on some validation test, and developers naively speak of zero trust systems without understanding the depth of what trust really is. But, trust is a deeply social phenomenon, which changes in real time based on social and technical interactions. By applying learning algorithms and data analytics to streamed interactions, this project attempts to qualify and quantify a measure of trust as a way of making realtime risk estimates.

Library Trust

More info: https://nlnet.nl/project/TrustSemanticLearning

Website: http://markburgess.org/trustproject.html
Repository: github.com/markburgess/TnT

Tvix

Alternative Rust-based software build transparency

Description

Tvix is a modern design and implementation of the Nix package manager (GPLv3). It brings a modular architecture in which components such as the build environment or package store are replaceable, which enables new use-cases and platforms. A graph-reduction evaluation model will make it possible to use Nix for package definitions and entire system configurations, its proven and tested use case, as well as for granular build definitions for individual components of software. Tvix will be fully compatible with nixpkgs, the existing package definition set for Nix, letting its users leverage more than a decade of community contributions and making it useful right out-of-the-box.

Reproducibility Rust

More info: https://nlnet.nl/project/Tvix

Website: tvix.dev
Repository: code.tvl.fyi/tree/tvix

TwPM

Open hardware implementation of Trusted Platform Module

Description

The Trusted Platform Module or TPM is a dedicated hardware component designed for providing additional security features for computing platforms. Currently, the market is dominated by the TPMs based on chips from large silicon vendors. The common characteristic of these modules is the proprietary firmware implementation.

TwPM project aims to increase the trustworthiness of the TPM module (hence the TwPM), by providing the open-source firmware implementation for the TPM device, compliant to the TCG PC Client Specification.

The main goal of the project is an attempt to create open-source firmware stack, implementing the TCG PC Client Platform TPM Profile specification. Project aims to use already available open-source software components whenever possible (such as TPM simulators for TPM commands handling), while developing new code when necessary (such as LPC FPGA module, or low-level TPM FIFO interface handling). Another challenge is to overcome hardware restrictions and allow users to use the open-source TPM implementation on generally-accessible development boards.

TPM

More info: https://nlnet.nl/project/TwPM

Website: twpm.dasharo.com
Repository: github.com/dasharo

TypeCell

CRDT-based collaborative block-based editor

Description

TypeCell aims to make software development more open, simple and accessible. TypeCell integrates a live-programming environment as a first-class citizen in an end-user block-based document editor, forming an open source application platform where users can instantly inspect, edit and collaborate on the software they're using. TypeCell spans a number of different projects improving and building on top of Matrix, Yjs and Prosemirror to advance local-first, distributed and collaborative software for the web.

CRDT CollaborativeEditing Matrix RealTimeCollaboration

More info: https://nlnet.nl/project/TypeCell

Website: www.typecell.org
Repository: github.com/TypeCellOS/TypeCell

UEFI isolation in VM from non UEFI firmware

Safer booting into UEFI-compliant operating system

Description

UEFI is the successor to BIOS, which initialises the bare hardware of a computer before handing over to a bootloader. The UEFI specification defines the architecture of platform firmware used for booting and its interface for run-time interaction with operating systems. As such, UEFI is responsible for bootstrapping pretty much every modern computer. In the majority of cases this is done with very little transparency for users - essentially relegating this enormously responsible position to a " black box" that just blips on the screen. Unfortunately trust in vendors to live up to their huge responsibility to make this safe and robust is not always justified: quite a few issues and security vulnerabilities in the (mostly proprietary) UEFI implementations have come to the surface via real-world exploits. The key open source booting mechanisms (like coreboot and Linuxboot/ u-root) are not UEFI compliant.

This project aims to close the gap in a pragmatic way: through virtualization - booting into a stripped down Linux and using the Kernel Virtual Machine (which is generally considered mature) to run the reference open source reference implentation of UEFI until it can hand over to a UEFI compliant boot loader. This is of course a security tradeoff (the early stage Linux used for virtualisation would not be able to use UEFI just yet itself in bootstrapping) , but it allows a single intervention to bridge to all different boot loaders and wholly avoid opaque proprietary ones by switching to open source ones. This also helsp to debug and assist in finding new solutions to cope with the shortcomings of native UEFI implementations.

More info: https://nlnet.nl/project/UEFI-isolation

Website: github.com/9elements/VMBoot

Servo improvements for Tauri

Verso offscreen + multiview

Description

Verso is a new browser initiative that is based on the Servo browser engine - a cross-platform, open source web engine written in Rust managed by Linux Foundation Europe. The project originates from an earlier effort to integrate Servo in Tauri, a widely used open source system for distributing cross-platform applications capable of running content and applications using web technology outside of the browser. The web ecosystem currently lacks a cross-platform, non-corporate controlled system for doing so, meaning that solutions like Tauri need to rely on the platform engines controlled by Apple, Google, and Microsoft. Obviously, this add complexity, has security and stability implications, lacks consistency, and involves limited levels of user agency. Integrating a portable browser engine would be a major step towards being able to run applications in a consistent, open source web runtime on major desktop and mobile platforms.

As part of that work, it became clear that several improvements to Servo are urgently needed. In order to speed up the development of those improvement, it turned out to be more efficient to transpose these requirements to a new standalone browser: Verso. The key tasks beyond improving developer efficiency and workflow (also for Mozjs and Spidermonkey) tackled in this project are offscreen rendering and multiwebview support.

Browser

More info: https://nlnet.nl/project/Verso

Website: versotile.org/verso
Repository: github.com/versotile-org/verso

Next Generation Browser Profile Workflow

A profile system for the Verso browser

Description

Users currently do not have much ownership over their browser data, including bookmarks, history, which extensions are activated, etc… Current web browsers do not really facilitate user agency, let alone in a standardised way. And we are not even mentioning the fact that synchronisation between devices is only possible through third parties, because there is no real transit between browsers (just imports). Even worse: despite this data being rather private, data is not really encrypted.

The solution is complex, and it starts with the rework of browser profiles and browser workflows conceptually. This project aims to define the standards of encapsulation of these profiles separately from the browser while keeping privacy and security in focus. The prototype would be integrated in the Verso browser, but along the way the underlying Servo engine also gets some improvements for accommodating these endeavours properly.

Browser **DID** **Privacy**

More info: https://nlnet.nl/project/Verso-Profile

Website: versotile.org/verso
Repository: github.com/versotile-org/verso

LIP6 VLSI Tools

Logical validation of ASIC layouts

Description

The software we run critically depends on the trustworthiness of the chips we use. LIP6's VLSI tools are one of the few user-operated toolchains for creating ASIC layouts where the full source code is available for inspection by anyone. This provides a significant contrast to commodity chips from vendors like Intel and AMD, where anything beyond coarse technical detail is shielded away by NDA's. This project will improve Coriolis2, HITAS/YAGLE and extend the whole toolchain so that it can perform Logical Validation. It will also upgrade the code to make it faster, able to handle larger ASIC designs, and add support for lower geometries (starting with 130nm) which are more energy-friendly.

ASIC VLSI

More info: https://nlnet.nl/project/VLSI-tools

Website: coriolis.lip6.fr
Repository: github.com/lip6

Vula

Encrypted ad hoc local-area networking

Description

With zero configuration, Vula automatically encrypts IP (v4) communication between hosts on a local area network (LAN) in a forward-secret and transitionally post-quantum manner to protect against passive eavesdropping. When the local gateway to the internet is a Vula peer, internet-destined traffic will also be encrypted on the LAN. With simple verification using QR codes, Vula is also able to disrupt active surveillance adversaries. Vula combines WireGuard for forward-secret point-to-point tunnels with cryptographically enhanced mDNS and DNS-SD for local peer discovery. Vula enhances the confidentiality of WireGuard tunnels by using CSIDH, a post-quantum non-interactive key exchange primitive, to generate a peer-wise pre-shared key for each tunnel configuration. Vula avoids the need for any Single Point of Failure (SPOF) such as a trusted third party. Vula is equally functional on otherwise air-gapped networks.

PostQuantumCrypto VPN

More info: https://nlnet.nl/project/Vula

Website: vula.link
Repository: codeberg.org/vula/vula

WikiRate: More Sites, More Cites

Persistent citation for Dekko-based open source data collections

Description

WikiRate.org is the largest open source registry of ESG data in the world with more than 3.5 million data points for over 100,000 companies. 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 needed to help and encourage companies to respond to the world's social and environmental challenges. To achieve this systemic change we need corporate accountability at scale. Focusing on the top 10, 100, or even 1000 companies, is not sufficient. Rather we need to monitor and understand impacts at industry and value chain levels, whilst leveraging individual corporate accountability to transform companies into positive agents of change. This follow-up project is focused on adding functionality to the underlying tool (Decko) which will allow in a finegrained way to point at specific data slices, as well as a history of any updates and corrections to such data.

API Accountability DataQuality GraphQL

More info: https://nlnet.nl/project/Wikirate-Cite

Website: wikirate.org
Repository: github.com/wikirate/wikirate

Winden/Magic Wormhole dilation

Improving Magic-Wormhole by implementing dilation and multiple file support for the web

Description

Winden is an open-source web app built on the Magic-Wormhole protocol, which allows two devices to connect and exchange data without requiring identity information. We are building Winden to make file-transfers for the web secure and private. With Winden, we are giving users control over their data without them needing to trust us. This project adds support for reconnection (referred to as the 'Dilation' protocol) and multiple file-transfers into both Winden and wormhole-william, the Go implementation of Magic-Wormhole used by Winden and other projects. Magic-Wormhole file-transfers require both parties to be online at the same time. Dilation allows for reconnection and changing networks during a transfer. This reduces the risks of connection interruptions during these synchronous transfers. Multiple file support is a much sought after need for transferring data, which requires Dilation (and Dilation's sub-channels).

FileSharing **WebApp**

More info: https://nlnet.nl/project/Winden-MWH-Dilation

Website: winden.app
Repository: github.com/LeastAuthority/winden

Wispwot

Implement generalized scalable protection against disruptive behavior in content discovery ●

Description

Spam and intentional disruption are a major problem in the clearnet. They make it infeasible to have comments on websites without moderation teams, privacy invading humanity checking, and access-restrictions, and they force social networks to decide between invasive censorship and exposing their community to abuse, propaganda and targeted harassment. The core of the problem is that spam scales better than spam-blocking.

This project brings the spam-defense from the Hyphanet Project to the fediverse. It replaces instant global visibility with incremental local visibility, fueled by positive social interaction and transitive blocking, so spammers quickly become invisible to most. To scale for groups of arbitrary size, it extends the system from Hyphanet by adding pruning of inactive accounts and efficient rediscovery. With this project, spam-protection scales better than spamming, reducing the work needed to cope with hostile communication, so group-communication won't require the outsourced, underpaid moderation teams that are prevalent in most centralized social networks.

ActivityPub Reputation

More info: https://nlnet.nl/project/Wispwot

Website: hg.sr.ht/~arnebab/wispwot

Yrs

Collaborative editing with CRDT written in Rust

Description

Yrs " wires" will be a native port (in the Rust programming language) of the Yjs shared editing framework. Abstractly speaking, Yjs allows many users to concurrently manipulate state that eventually converges. It is a popular solution for enabling collaborative editing (Google Docs style) on the web because it is indefinitely scalable, works peer-to-peer, and has a rich ecosystem of plugins. There are plugins that allow you to connect with other peers over different network providers (WebRTC, Websocket, Dat/Hyper, IPFS, XMPP, ..) and there are many editor plugins that allow you to make existing (rich-)text editors collaborative.

The Yjs project is about connecting projects with each other and providing a network-agnostic solution for syncing state. A native port will allow native applications (e.g. XI, Vi, Emacs, Android, iPhone, ..) to sync state with web-based applications. We chose Rust because it's well suited to be embedded in other languages like C/C++, PHP, Python, Swift, and Java. With Yrs, we want to connect even more projects with each other and provide a modern collaboration engine for native applications.

The Rust implementation will implement the full feature set of the shared types, including the event system. This will enable users to parse existing Yjs documents, manipulate them, and implement collaborative applications. The port will make it easy to " bind" to another language so that the shared state is available in other languages as well. There will likely be a WASM binding, a C++ binding, and a Python binding (provided by Quantstack). Other existing features like awareness, selective Undo/Redo manager, relative positions, and differential updates will be added after the initial release.

CRDT E2EE Framework Interoperability P2P RealTimeCollaboration

More info: https://nlnet.nl/project/Yrs

Website: docs.rs/yrs
Repository: github.com/yjs/y-crdt

Yrs Undo

Rust-based CRDT framework for real-time multi-user applications

Description

Yrs " wires" is a native port (in the Rust programming language) of the Yjs shared editing framework. Abstractly speaking, Yjs allows many users to concurrently manipulate state that eventually converges. It is a popular solution for enabling collaborative editing (Google Docs style) on the web because it is indefinitely scalable, works peer-to-peer, and has a rich ecosystem of plugins. There are plugins that allow you to connect with other peers over different network providers (WebRTC, Websocket, Dat/Hyper, IPFS, XMPP, ..) and there are many editor plugins that allow you to make existing (rich-)text editors collaborative. This project will add a selective Undo/Redo manager, include support for other native clients and to interop with languages like Java, PHP and Swift. The goal is to reach full feature compatibility with Yjs and improve its performance even more - bringing a collaborative, decentralized experience where users' data lies in their own hands.

CRDT CollaborativeEditing E2EE Framework Interoperability P2P RealTimeCollaboration TypedLanguage

More info: https://nlnet.nl/project/Yrs-Undo

Website: docs.yjs.dev
Repository: github.com/y-crdt/y-crdt

Quantum-Proof Zenroom

Implementation of Quantum-Proof Cryptography in Zenroom

Description

Zenroom is a tiny secure execution environment that integrates in any platform and application, even on a chip or a web page. It executes human-readable smart contracts for all kinds of use cases, such as databases, blockchains and much more. Zenroom is scriptable in an English-like language called Zencode.

During this project quantum-proof cryptography will be implemented in Zenroom by strictly adhering to ECDH specifications for common session exchanges, signature and verification, applying liboqs transparently as a back-end to existing Zencode scenarios. This makes it seamless to substitute existing EC implementations with the same Zencode. The result will be a fully portable software (plain C, no hardware acceleration) of the NIST quantum-proof competition winner algorithm and full alignment with its final test vectors.

PostQuantumCrypto SmartContracts TaskIsolation Virtualisation

More info: https://nlnet.nl/project/Zenroom-oqs

Website: http://zenroom.org
Repository: github.com/dyne/Zenroom

Group Theoretic Zero-knowledge Proofs (0KNOW)

Description

Zero-knowledge proof (ZKP) systems help principals verify the veracity of a piece of information without sharing the data. The overall goal of 0KNOW is to develop a lightweight group-theoretic zero-knowledge proof (GT-ZKP) system that can be employed as a cryptographic primitive in many security protocols such as identification, authentication, or credential ownership. They are widely used to preserve confidentiality and ownership of data. GT-ZKP can be seen as a reusable building block for making the future internet trustworthy and secure. In 0KNOW, we will focus on NP group-theoretic problems and design GT-ZKP by finding an appropriate platform group based on the selected difficult problem considering its applicability in the post-quantum era and we will develop an open-source implementation of GT-ZKP.

ZeroKnowledgeProof

More info: https://nlnet.nl/project/0know

Website: arxiv.org/abs/2206.13350
Repository: github.com/cansubetin/sdzkp

Colofon

NGI Assure is made possible with financial support from the European Commission's Next Generation Internet programme, under the aegis of DG Communications Networks, Content and Technology.

NGI Assure is a collaboration between the following partners:

— Innovation Engineering SRL (IT)
— Fundingbox Accelerator SP Zoo (PL)
— Fundingbox Communities SL (ES)
— NLnet Foundation (NL)

Designed and typeset in the Netherlands in Source Sans Pro with free & open source software only: Inkscape, Typst and Vim.

Hex logo's can be downloaded via: https://nlnet.nl/hex

Photo cover courtesy of **Manifestations festival 2023**.
Jules Sinsel - Entanglement in Machine Learning
Photographer: Edwin Smits

Project officers:

— Jean-Luc Dorel
— Stefano Foglietta
— Jorge Gasos
— Stergios Tsiafoulis

Financial officer:

— Stéphane Andries

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 957073.

European Commission

Between September 2020 and August 2024, over 150 project teams from over 40 countries worked on free and open source R&D , as part of a research programme called *NGI Assure*.

In this unique programme, a great diversity of technologies was tackled by many talented researchers from Europe and beyond: from post-quantum cryptography to productivity tools, from securing the software supply chain to digital signatures, from EDA design tools to end-to-end encrypted instant messaging, and from modern email standards to standardising object capabilities. The unifying factor: all of these projects worked on *digital commons* that improve our digital sovereignty, empower end users, increase systemic resilience and provide transparency, choice and self-determination.

NGI Assure was set up by Innovation Engineering, stichting NLnet and FundingBox, as part of the European Commission's *Next Generation Internet* initiative. NGI is an ambitous collaborative effort to re-imagine and re-engineer the internet for the third millenium and beyond. In NGI Assure, we see the heights of what human talent can accomplish levering the power of free and open source collaboration.

INNOVATION ENGINEERING

FundingBox

nlnet FOUNDATION

NGI ASSURE