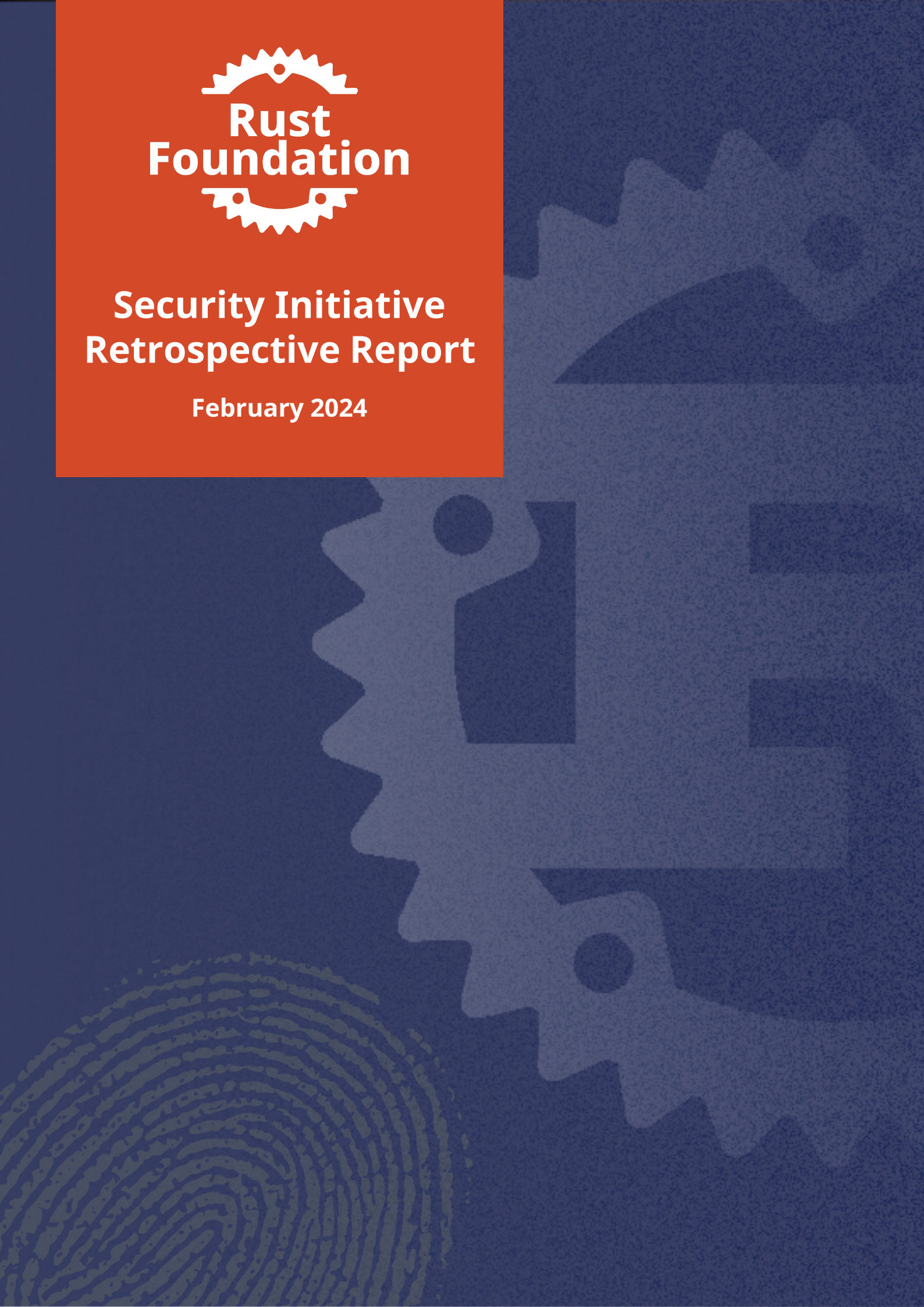


Security Initiative Retrospective Report

February 2024

CONTENTS

Foreword by Dr. Rebecca Rumbul, Executive Director & CEO of the Rust Foundation

On behalf of the Rust Foundation, I am pleased to present the second installment in an ongoing series of reports that detail the activities, progress, and accomplishments of our Security Initiative.

2023 was a productive and exciting year for both the Rust Foundation and the Rust Project's hardworking contributors. As global interest in and adoption of Rust grows, the Rust Project, Rust community, and Rust Foundation will need to be in close collaboration to properly and safely scale Rust for all.

The Security Initiative is a fantastic example of what we can achieve with the Rust Foundation's expert stewardship.

As you'll find in this second Security Initiative Report, we have continued to make excellent contributions toward strengthening and expanding security in the Rust ecosystem since we issued our first report in this series. Thanks to the Rust Foundation's talented team of software engineers, we have been in close and active collaboration with key Rust Project teams and working groups and have made real improvements to the state of Rust security in under a year.

There is already so much to show for this program, from several new open source security projects to completed and publicly available security threat models. None of the milestones detailed in this report would have been possible without support from Rust Foundation Platinum Members AWS and Google or infrastructure support from Wiz. I would be remiss not to give a special thanks to OpenSSF's Alpha-Omega project for supporting the entire Security Initiative with two years of funding.

Additional thanks are owed to the leaders of the Rust Foundation's Technology Team for turning the intentions of our Security Initiative into reality. It has been exciting to watch our engineers take on new responsibilities within the Rust Project over the last six months, including Joel Marcey being named editor of the forthcoming Rust language specification, and Jan David Nose becoming co-lead of the Infrastructure team. At the Rust Foundation's Rust Global event in September 2023, a packed audience of Rust advocates and enterprise leaders had the opportunity to hear about our Security Initiative directly from our Technology Team – I anticipate more opportunities for the global Rust community to benefit from their expertise and security advocacy in 2024.

With guidance from the Rust Foundation's Technology Team, resources from our generous supporters, and collaboration with Rust Project and industry security leaders, we will continue making fantastic strides toward a more safe, stable, and secure Rust ecosystem.

Dr. Rebecca Rumbul

Executive Director & CEO
The Rust Foundation

Executive Summary

Since the first stable release of the Rust programming language in 2015, the Rust ecosystem has grown tremendously, thanks to the tireless work of its maintainers, contributors, and advocates. The many advantages of Rust are becoming increasingly evident to developers and organizations alike.

But the meteoric rise of Rust introduces a number of complexities and risks. When the user base of any programming language grows, it becomes more attractive to malicious actors. As a programming language ecosystem expands with more libraries, packages, and frameworks, the surface area for attacks increases with it. Rust is no different.

As the steward of the Rust programming language, the Rust Foundation has a responsibility to provide a range of resources to the growing Rust community. This responsibility means we must work with the Rust Project to help empower contributors to participate in a secure and scalable manner, eliminate security burdens for Rust maintainers, and educate the public about security within the Rust ecosystem.

In September 2022, the Rust Foundation announced its commitment to fulfilling these responsibilities through the Security Initiative. In Q3, we issued an initial report of our contributions between December 2022 and July 2023. Since then, we have made critical progress toward our security focus areas.

Recent Achievements of the Security Initiative Include:

- Completing and releasing Rust Infrastructure and Crates Ecosystem Threat Models.
- Further developing Rust Foundation open source security project Painter and releasing new security project, Typomania.
- Utilizing new tools and best practices to identify and address malicious crates.
- Helping reduce technical debt within the Rust Project, producing/contributing to security-focused documentation, and elevating security priorities for discussion within the Rust Project.

... and more!

Over the Coming Months, Security Initiative Engineers Will Primarily Focus On:

- Completing all four Rust Security Threat Models and taking action to address encompassed threats.
- Standing up additional infrastructure to support redundancy, backups, and mirroring of critical Rust assets.
- Collaborating with the Rust Project on the design and potential implementation of signing and PKI solutions for crates.io to achieve security parity with other popular ecosystems.
- Continuing to create and develop tools to support Rust ecosystem, including the crates.io admin functionality, Painter, Typomania, and Sandpit.

Program Mission & Personnel

The mission of the Rust Foundation Security Initiative is to support and advance the state of security within the Rust Programming language ecosystem and community.

We are grateful for the widespread collaboration amongst a variety of stakeholders who share the security values driving this work forward. The Rust Foundation would like to thank the following individuals for their many contributions to the Security Initiative at the time of this report:

Rust Foundation Technology Team

Joel Marcey
Director of Technology
The Rust Foundation

Walter Pearce
Security Engineer
The Rust Foundation

Adam Harvey
Software Engineer
The Rust Foundation

Tobias Bieniek
Software Engineer
The Rust Foundation

Jan David Nose
Infrastructure Engineer
The Rust Foundation

Rust Project Teams & Working Groups

Rust Project Cargo Team

Rust Project crates.io Team

Rust Project Infrastructure Team

Rust Project Security Response Working Group

Rust Project Secure Code Working Group

Additionally, the Rust Foundation would like to acknowledge all other individuals within the Rust Project who have contributed to the Rust Foundation Security Initiative and provided valuable input thus far.

Security Initiative Support & Sponsorship

The Security Initiative would not exist without financial and in-kind support from these generous organizations...

Alpha-Omega

Founding Financial Support

The Security Initiative is underwritten by OpenSSF's Alpha Omega project, which partners with open source software projects and maintainers to improve the global software supply chain security. We are grateful for their ongoing support. Monthly updates on the Foundation's Security Initiative work can be found on the Alpha-Omega GitHub repo.

aws

Financial Support

Special thanks are owed to Rust Foundation Platinum Member AWS for their generous financial donation to support the Security Initiative – and for lending us the engineering support of Dan Gardner in 2023.

JFrog

In-Kind Support

JFrog generously donated the time of Security Engineer Shachar Menashe which we utilized in 2022 before welcoming our three full-time engineers to the Rust Foundation team.

WIZ

Infrastructure Donation

Wiz, a platform that helps locate and mitigate security issues in cloud infrastructure, generously donated access to their platform to the Security Initiative.

Google

Consultation

Rust Foundation Platinum Member Google has lent its time and expertise to the ongoing development of the Security Initiative. We look forward to further collaborating with them as this work continues.

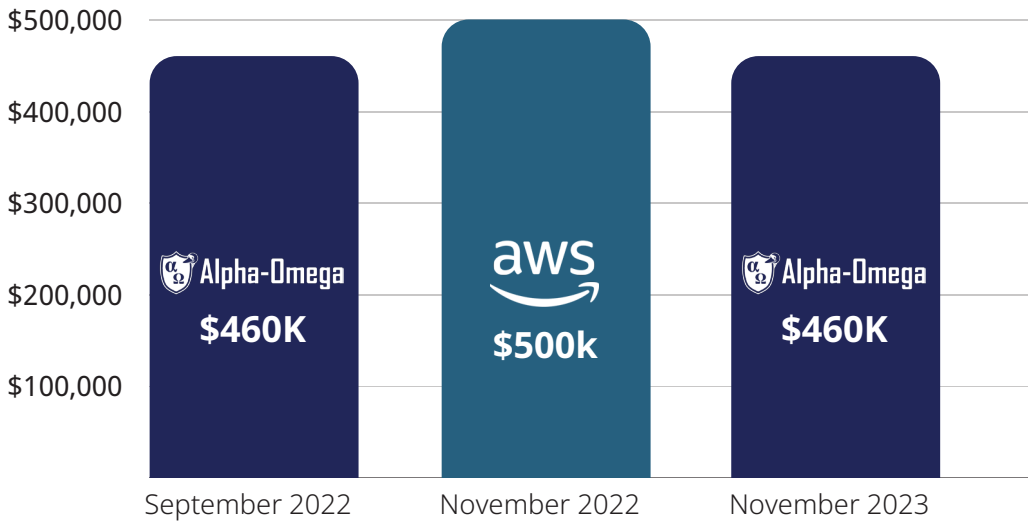
If your organization is interested in supporting the security of the Rust language ecosystem through our Security Initiative, please email us at contact@rustfoundation.org.

Financial Report

Below, you will find key data about the funding of the Rust Foundation's Security Initiative. We strive to run a lean operation and make good use of a relatively modest budget.

We are providing these details for general transparency and to demonstrate the work the Rust Foundation is able to carry out through the generous funding we have received from donors

Financial Contributions to Date (as of January 2024)



$1.42M in total financial contributions to date.

Funding vs. Expenditure Notes:

- In August 2023, the Security Initiative had $528,398 in available funding.
- Between August 2023 and January 2024, we utilized $436,113 of this funding.
- Thanks to Alpha-Omega's second donation of $460k in November 2023, we have $552,285 available as of January 31, 2024.

Recent Focus Areas:

From August 2023 through January 2024, our work under the Security Initiative was guided by the following priorities...

1. Rust Foundation Security Initiative Leadership
2. Conducting a Security Audit of the Rust Ecosystem
3. Threat Modeling
4. Continued Advocacy for Security Best Practices Across the Rust Ecosystem
5. Tools, Features, & Recommendations Based on Security Research
6. Rust Security Documentation
7. Prioritizing & Addressing Key Rust Security Issues

On pages 9 through 16, you'll find details on the progress we've made toward each of these priorities.

Focus #1:
Rust Foundation Security Initiative Leadership

As detailed in our July 2023 Security Initiative Report, the Rust Foundation's Technology Team was well-equipped with security engineering and general software engineering talent by the second half of 2023 after several important hires in Q1-Q2. This progress allowed us to support two of our Technology Team members in pursuing additional responsibilities within the Rust Project, through which they can contribute critical security expertise.

Progress:

In late July 2023, Pietro Albini (a longtime leader within the Rust Project,) shared his decision to step down from the Rust Project's Infrastructure Team after four years of excellent work in the role. Following this announcement, the Infrastructure Team collaboratively decided on a new system of rolling team co-leads. In September, the Infrastructure Team announced that the Rust Foundation's Infrastructure Engineer, Jan David (JD) Nose, was named co-lead of the Infrastructure Team, alongside Jake Goulding.

In December of 2022, the Rust Project published an RFC to gain consensus on developing a Rust language specification, which will serve as a definitive guide for how the Rust language should behave. During the process of this RFC being approved, it was decided that the Rust Foundation would coordinate the development of the specification and help facilitate the interviewing process for the editor role. The Rust Project's specification team and the Rust Foundation jointly decided that Joel Marcey (Director of Technology at the Rust Foundation and a key leader of the Security Initiative) had the proper experience to fill the editor role.

The Rust Foundation is pleased to support JD and Joel's new roles within the Rust Project and we see these as critical Security Initiative milestones. As co-lead of the Rust Infrastructure Team, a member of the Rust Foundation Technology Team, JD can help us work towards hardening Rust infrastructure assets with support from our Security Initiative audits and threat models. Because the infrastructure that hosts the Rust Project contains assets which, if compromised, can affect the entire project, we are carefully thinking about access roles for those who have credentials to impact infrastructure, crate signing, backups, mirroring, and other security areas. As editor of the forthcoming Rust language specification, Joel will help ensure that the language and toolchains are compatible with the Rust security best practices. We feel that JD and Joel are both extremely qualified for these positions. Given their leadership roles at the Rust Foundation and within the Security Initiative, we see great opportunities for security advocacy and closer collaboration between the Rust Project and the Foundation.

Focus #2:
Conducting a Security Audit of the Rust Ecosystem

Over the past several months, our team has continued to audit the state of security within the Rust ecosystem. This work will allow both the Rust Foundation and the Rust Project to better anticipate risks and define how security can be economically maintained over an ongoing basis. Given the size of our team and the Rust ecosystem, we have a unique opportunity to learn from the examples of other ecosystems and implement appropriate remediations for them at a smaller scale.

Progress

As the official package repository for the Rust programming language, the health of the packages on crates.io depends on deeper insight into crate security. One of our goals for Focus 2 is to enable further insight into crate security and more prominently feature information about crate security. Through its integration with the crates.io publishing pipeline, Typomania (introduced in Focus 5) has already identified several malicious crates, which the crates.io team was able to quickly act upon. Security Engineer Walter Pearce conducted crate checks via a tool called Sandpit, allowing him to identify additional malicious crates as well.

Focus #3:
Threat Modeling

Threat modeling enables the Rust Foundation and Rust Project to better understand the risks identified by the Security Audit. While developing all four threat models described below, we have consulted with the Rust Project's crates.io Team, Infrastructure Team, Security Response Working Group, Secure Code Working Group, and specific external stakeholders.

Progress:

In 2023, the Rust Foundation completed - or made significant progress toward - a total of four threat models as of January 2024.

Completed

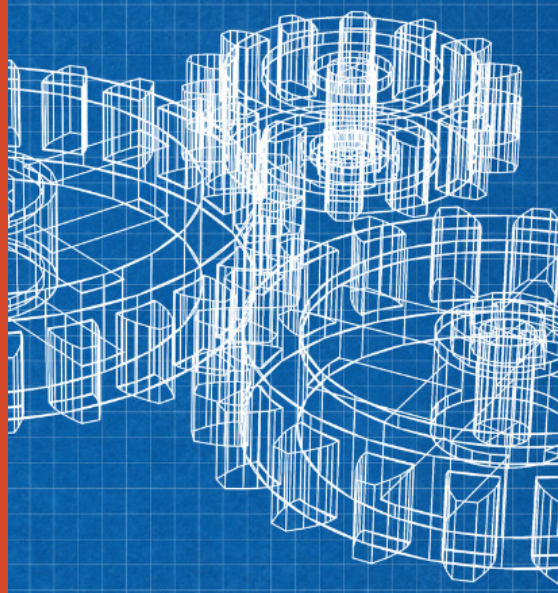
The Crate Ecosystem Threat Model covers the crate ecosystem of the Rust Programming Language. This model was crafted in collaboration with various teams across the Rust Project, corporate stakeholders, and end users of the language. The Crates Ecosystem Threat Model is available to the public here.

The Rust Infrastructure Threat Model covers the software infrastructure supporting the Rust Project, including crates.io, docs.rs, Bors, Playground, Crater, Rustwide, Docker, rust-lang repositories, cargo, and Rustup. The Rust Infrastructure Threat Model is available to the public here.

In Progress

The Rust Project Threat Model will cover the key processes of the Rust Project and the people involved in maintaining, contributing to, and leading it. It will illustrate the risks associated with bad actors damaging the Rust Project and its reputation.

The crates.io Threat Model will cover the crates.io infrastructure and code, illuminating areas for improvement within management, administration, deployment, and code quality.



Focus 4:
Continued Advocacy for Security Best Practices Across the Rust Ecosystem

Advancing the security of the Rust language hinges on the safety of its associated tools. By promoting security awareness and education within the official package manager and build system for Rust (Cargo), the official package repository for Rust (crates.io), and the broader Rust ecosystem, the Security Initiative will help developers use Rust responsibly and empower them to make informed decisions about its use. Between August 2023 and January 2024, we made meaningful progress toward these important goals.

Progress

Public Key Infrastructure (PKI) for Crates

Since starting the Security Initiative, the Rust Foundation Technology Team has gained a better understanding of how critical crate signing and verification are to a holistic security strategy for the Rust ecosystem. The Rust Project does not yet have official Public Key Infrastructure (PKI) required to perform package,mirror, code, or release signing; There is a high demand for capabilities that require such PKI infrastructure, such as mirroring and binary signing.

To remedy gaps that exist in this domain, Security Engineer Walter Pearce initiated a discussion with the Rust Project between August and December about developing a PKI. We anticipate that this draft will soon be published as an official Rust Project RFC for discussion with a wider audience of Rust Project and community members. Our expectation is that this upcoming RFC will be an important step towards using signing to help mitigate some of the threats surfaced by the Crates Ecosystem threat model.

Security Best Practices Advocacy

We spent time over the past several months assessing various industry security frameworks that could be utilized in helping secure the Rust ecosystem, including OpenSSF's scorecard and SLSA compliance. In November, the Rust Foundation accepted an invitation from OpenSSF to serve as a signatory of their Secure Software Guiding Principles. We intend to continue assessing similar frameworks to guide our security initiative efforts over the coming months.

Additionally, the Rust Foundation and crates.io teams worked with our legal counsel to add an amendment to our privacy policy, which outlines how we would operate if a legally binding request for data is issued. Our goal will be to identify opportunities to minimize the level of data we retain without compromising the management of the Rust infrastructure.

Focus 5:
Developing Research-Based Tools, Features, & Recommendations

Though the Security Initiative is still young, it has already helped our team understand aspects of the Rust ecosystem that would be better served by new tools, features, and resources. Since our last report in July 2023, we have expanded the capabilities of our existing tools and developed new resources to help bolster maintainer security workflows and enable deeper insight into Rust threats.

Progress

Improvements to the Rust Foundation's First Open Source Project, "Painter"

In July, the Rust Foundation announced the public availability of Painter: an open source project created by Security Engineer Walter Pearce that creates a complete call graph across the entire crates ecosystem to reveal how crates relate to each other. When a vulnerability exists in one crate, Painter allows users to more easily assess potential or active risks to other crates.

Since Painter was released in July, we have added support for binary dependencies, crate and function-level data annotations (unsafe code coverage included), syscrates, and external library static analysis. Painter is also now able to extract unsafe code statistics, and parse third-party C-library dependencies. At the time of this report, Painter covers over 85% of the crate ecosystem.

We have been pleased by the public reception of Painter; several groups have already requested access to its real-time data, which we are currently evaluating.

Public Availability of "Typomania" Tool

In October, Software Engineer Adam Harvey released The Rust Foundation's second open source project, Typomania. This tool is capable of finding crates that may be trying to pretend to be another. While the Rust language is quite secure, crate package typo-squatting is one of the main threat vectors that exist. With Typomania, we hope to minimize this threat.

Adam has integrated Typomania into the crates.io publishing pipeline, ensuring that the Security Initiative's team members and other interested parties from the Rust Project's security response working group are notified when a potential typosquatting incident occurs.

crates.io Admin Consol

The crates.io admin console described in our last report is still under development. This console will allow for the ability to easily do security-related administrative tasks for crates.io, and quickly and nimbly react to potential incidents (e.g., yanking/deleting crates and blocking users).

Infrastructure Improvements

The Rust Foundation was pleased to facilitate a generous in-kind infrastructure donation from Wiz to support the Rust Project. Wiz has now been deployed into production and proactively alerts us to potential security vulnerabilities within **Rust's** infrastructure.

Ecosystem Scanning, codename "Sandpit"

Since our last report in July 2023, our team has continued to work on creating automated tooling and techniques for identifying possibly malicious activity on crates.io. "Sandpit" continues to be used internally as a tool to help identify malicious crates and has already found several malicious crates in its use.

Walter Pearce is currently working on operationalizing Sandpit to ensure it runs in production on Google Cloud Platform (GCP). Over the coming months, the Rust Foundation team will continue developing detections to monitor the ecosystem such as malware scanning, malicious activity detection, vulnerable dependency scanning, source code provenance, and more.

Crate Quarantining

The Rust Foundation team has written a Request For Comment for the ability to quarantine problematic crates after a set of security thresholds have been met. If approved, this feature will make it possible to keep a crate in a holding pattern from public use while security checks are made within the crates.io infrastructure to ensure its safety. It will also be added as an operation within the crates.io admin console (described above).

Continued Reduction of crates.io Technical Debt

Below is a recap of our team's recent crates.io technical debt reduction efforts:

- In early September our engineers started to migrate the crates.io logs away from the previous provider and into DataDog. This allows the crates.io team to query their logs in entirely new ways and makes their analysis significantly easier. One outcome of this work is a public dashboard that shows the distribution of Cargo versions across all of the requests that arrive at the crates.io servers.
- Maintainable and secure codebases rely on comprehensive test suites. Tobias spent considerable time over the past several months making it easier to write tests for the crates.io project, and then developing tests for the areas of the Project that were insufficiently protected. He has also improved the manner in which test coverage is tracked, making it easier to identify parts of the Project where more tests would be beneficial.
- Typically, the crates.io servers are updated multiple times per week. Previously, such updates would take up to a minute, resulting in many dropped requests, slowness, and CI issues for all users of crates.io. After an analysis showed the root cause to be the combination of nginx and Heroku, Tobias addressed the issue, resulting in significantly faster server updates and consistent performance.
- Additional work has been done over the past several months to give more individuals admin permissions on crates.io without granting them full access to the server and database. The admin permissions are now synchronized with the "team" repository – the central place that documents and defines all access control concerns of the Rust Project.

Focus #6:
Rust Security Documentation

Through our work to outline security risks, vulnerabilities, and key considerations within technical Rust documentation, developers will be in a better position to abide by security best practices and make informed choices when deploying Rust code. The expansion of Rust's high-quality security documentation will enable developers to identify and address potential vulnerabilities early in the Rust development process, preventing common security pitfalls, and further educating the community about the existing security benefits of Rust.

Progress

We have initiated a community discussion about proper detections and how security guarantees should be surfaced to users with automated tooling.

The Rust Foundation team has also begun interfacing with the Infrastructure Team to document access control provisioning and de-provisioning within the Rust Project. We are committed to helping them identify gaps and areas for automation or improvement.

Focus 7:
Prioritizing & Addressing Key Rust Security Issues

While the Rust security audit we initiated during the previous reporting period is still underway, our engineering team was heavily involved in addressing the resolution of a key security issue in August and prioritizing key changes to further safeguard the Rust ecosystem.

Progress

Malicious Crate Detection

In August, the Rust Foundation was notified by Louis Lang at Phylum of a new user who had uploaded nine crates that typosquatted popular crates with malicious intent. After the Rust Foundation surfaced the information to the Rust Project's crates.io Team, the crates were immediately yanked, the user account locked, and the crates fully-removed from the crates.io file store on August 18. Phylum and the crates.io team published post-mortems of the incident which include descriptions of how Walter Pearce was involved in the analysis process. The Rust Foundation is developing tools like Typomania, Painter, and Sandpit to help with malware detection and resolution.

Package Manifest Changes

Tobias and Adam worked towards treating "Cargo.toml" manifest files as the source of truth for any package metadata when uploaded to crates.io. Following a metadata confusion-related attack on the npm ecosystem in June 2023, our team worked with the crates.io and Cargo teams to determine that this particular attack vector was not viable in the Rust ecosystem. It was, however, determined that relying on uploaded metadata instead of extracting the metadata on the server itself is risky other reasons. This work concluded in early October 2023.

Upcoming Areas of Focus

During the Security Initiative's first full year of operation, we hired a team of fantastic engineers, began laying the foundation of our work, and met our initial goals within high-priority Rust security focus areas.

In the months ahead in 2024, the Rust Foundation plans to...

- Implement security solutions and best practices we developed in 2023 across the entire ecosystem.
- Make all of our threat models publicly available and address the risks they identify.
- Incorporate additional infrastructure to support redundancy, backups, and mirroring of critical Rust assets.
- Implement signing and PKI solutions to achieve security parity with other popular ecosystems.
- Help ensure that gaps in the Rust Project's infrastructure are filled and proactively signal any infrastructure security issues that may arise.
- Further support administrative functions in crates.io to advance the state of crate security.
- Utilize our ecosystem scanning services to detect malicious crates so appropriate action can be taken.
- Provide data about crate security to the public through mechanisms like badging, signing status, etc.
- Expand Painter's capabilities to provide even more scanning power.
- Help advance the off-by-default crate quarantine system proposal and recommendation RFC.

... and more!

Over the coming months, the Rust Foundation looks forward to building upon the initial progress we made in 2023 and delivering real-world solutions to all the known threats in the Rust ecosystem.

Thank you for reading the second retrospective Rust Foundation Security Initiative Report. We look forward to issuing future reports detailing our work on these and other Security Initiative priorities.

Please join us in congratulating our Technology Team, collaborators in the Rust Project, and our supporters for an impressive first year of the Rust Foundation Security Initiative!

Acknowledgments

We would like to thank the Rust Foundation's growing Technology Team for continuing to oversee the important work outlined in this report. We would also like to thank the many key individuals and teams within the Rust Project for their leadership and participation. Finally, thank you again to our generous supporters whose donations make the Security Initiative possible.

Supporting the Security Initiative

If your organization is interested in supporting the security of the Rust programming language through this program or if you have any questions about the Rust Foundation, please email us at contact@rustfoundation.org.