Rust Foundation

Security Initiative Report

December 2022 - July 2023

Contents

Foreward by Rebecca Rumbul 3

Executive Summary 4

Program Mission & Leadership 5

Support & Sponsorship 6

Financial Report 7

Focus Areas: December 2022 - July 2023 8

Upcoming Areas of Focus 16

Recent News 17

Acknowledgements 18

Foreword by Rebecca Rumbul

Executive Director & CEO of the Rust Foundation

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

As the popularity and interest in the Rust language continue to climb, refrains of its benefits are becoming louder in all corners of the open source ecosystem. Just as developers are turning to Rust in increasing numbers to build performant systems, prominent government agencies are beginning to see Rust as a safer coding solution – particularly for security in the software supply chain. Indeed, the hardworking maintainers of Rust have always prioritized security and safety – many of its built-in features are evidence of this. Rust's stellar reputation as a safety and security tool in coding has grown more robust along with its visibility, popularity, and adoption.

But to properly support the future of Rust and its growing community, the Rust Foundation believes that proactive measures must be taken to strengthen and expand the project's security and safety. That's why we launched our Security Initiative in September of 2022 with generous founding support from OpenSSF's Alpha-Omega project, technical support from Platinum Member JFrog, and subsequent financial support from Platinum Member Amazon Web Services.

The Security Initiative has already enabled the Rust Foundation to hire dedicated software security experts, identify initial security priorities in collaboration with Rust Project leadership, begin conducting a thorough audit of the Project and community, and start engineering real solutions.

In this report, you'll find details on the leaders involved in this work, our activities to date, and our plans for the next phase.

I am thrilled to share this first Security Initiative Report, which covers the progress and successes of all parties involved in this ongoing effort. I hope you find this first chapter in the story of the Rust Foundation Security Initiative to be informative and consider it through the lens of the great things we can do with a safe, stable, and secure Rust language ecosystem.

Finally, I would like to thank the hardworking Technology Team at the Rust Foundation, the sponsors and donors who made financial and in-kind contributions to the Security Initiative, and the leaders from within the Rust Project who have partnered closely with us to make this critical work possible.

Rebecca Rumbul

Rust Foundation Executive Director & CEO

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. Every day, Rust's many advantages are becoming more evident to developers and organizations alike.

But like any programming language, 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 any programming language ecosystem expands with more libraries, packages, and frameworks, the surface area for attacks increases. 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 empower contributors to participate in the Rust Project 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. Less than one year after that announcement, we are proud of the progress made toward our initial focus areas.

Achievements of the Security Initiative between December 2022 and July 2023 include:

- Recruitment, onboarding, and productivity of full-time security engineering expertise within the Rust Foundation
- Completion of several threat modeling exercises with plans to publicize results in the near future
- Public availability of Painter: an open source tool that creates a complete call graph across the entire crates ecosystem
- A variety of crates.io security improvements, including the creation of a crates.io admin console, scoped API tokens, and technical debt reduction.

Over the coming months, Security Initiative engineers will be focused on:

- Collaborating with all Security Initiative stakeholders to build out the "Rust Security Toolkit"
- Completing our first Rust security audit
- Releasing the tools and threat models we have created to the public

... and much more!

Program Mission & Leadership

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

Walter Pearce
Security Engineer

Tobias Bieniek
Software Engineer

Jan David Nose
Infrastructure Engineer

Adam Harvey
Software Engineer

Rust Project Teams & Working Groups

Rust Project Infrastructure Team

Rust Project Security Response Working Group

crates.io Team

Rust Project Secure Code Working Group

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

Support & Sponsorship

The Security Initiative would not exist without financial and in-kind support from a number of 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 can be found on the Alpha-Omega GitHub repo.

aws

Financial Support

Special acknowledgment is 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.

JFrog

In-Kind Support

Rust Foundation Platinum Member JFrog generously donated the time of Security Engineer Shachar Menashe which we utilized 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

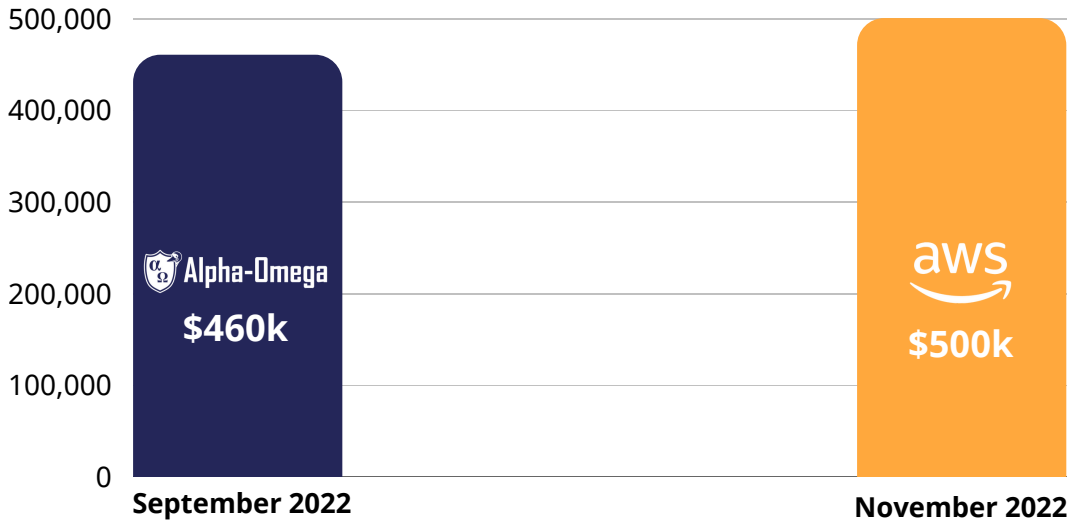
Rust Foundation Platinum Member Google has lent its time and expertise to the ongoing development of the Security Initiative. We look forward to further collaborating with them as this work continues.

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.

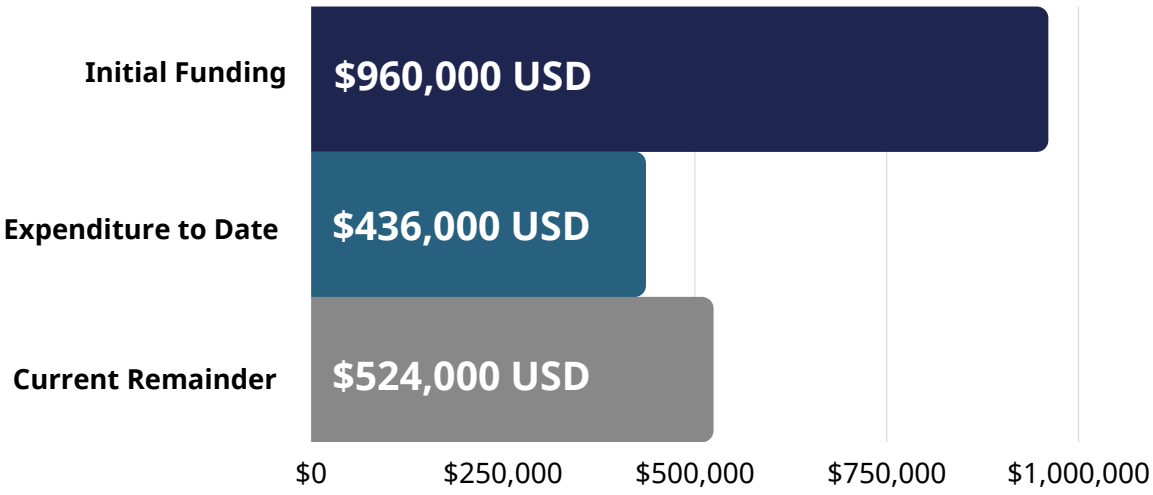
Financial Report

Below, you will find key data about the funding and budget of the Rust Foundation's Security Initiative. We are providing these details for general transparency and to demonstrate the work we are able to carry out through the generous funding we have received from donors.

Financial Contributions (as of July 2023)



Funding v.s. Expenditure (as of July 2023)



Rust Foundation

Focus Areas: December 2022 - July 2023

From December 2022 through July 2023, our work under the Security Initiative was guided by the following priorities...

1. **Hiring Rust Foundation Security Engineering Expertise**
2. **Conducting a Security Audit of the Rust Ecosystem**
3. **Conducting Threat Modeling for the Rust Ecosystem**
4. **Advocating for Security Practices Across the Rust Landscape**
5. **Developing Tools, Features & Recommendations Based on Research**
6. **Developing Documentation to Demystify Security in the Rust Ecosystem**
7. **Addressing Recommended Rust Security Issues Identified Through Research**

On pages 9 through 15, you'll find details on the progress we've made toward each of these priorities.

Rust Foundation

Focus 1:

Hiring Rust Foundation Security Engineering Expertise

> In order to make the Security Initiative possible, the Rust Foundation required the leadership and knowledge of a full-time engineer with a well-established background in security to help shape the work of this program. We also required the support of several full-time engineers to carry out the work being directed by the Security Engineer.

Progress:

On January 13, 2023, the Rust Foundation hired Walter Pearce as our first full-time Security Engineer. Walter has over 14 years of relevant security engineering experience and has already helped the Rust Foundation make tremendous progress toward the goals of this program.

Later in Q1 of 2023, the Rust Foundation hired two additional full-time Software Engineers: Adam Harvey, who is partially focused on the priorities identified by the Security Initiative, and Tobias Bieniek, who is focused on supporting the crates.io ecosystem. Walter, Adam, and Tobias have been essential contributors to and leaders of the Security Initiative activities and we are thrilled to have them as part of the Rust Foundation team.

Focus 2:

Conducting a Security Audit of the Rust Ecosystem

> An audit of the state of security within the Rust ecosystem will allow both the Rust Foundation and Project to anticipate risks better and define how security can be economically maintained on an ongoing basis. Given the size of our team, the community, and the ecosystem at large, we have a unique opportunity to learn hard lessons from 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 was to enable further insight into crate security and more prominently feature information about crate security.

Work is currently focused on software supply chain security. The Rust Foundation and crates.io teams are currently collaborating to surface individual crate security information. Assessment efforts include leaked secrets, malicious crate detection, and security best practices scoring models.

The team has not identified any actively malicious crates thus far. Multiple cases of leaked credentials were discovered, and the team has actively contacted the affected owners.

Additionally, the Rust Foundation and crates.io teams collaborated in June to produce a statement on our shared, general approach if either party ever receives a legally-binding request for data. While no such request has been received at the time of this report, we plan to work with our legal counsel to add an amendment to our privacy policy that 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 3:

Conducting Threat Modeling Exercises for the Rust Ecosystem

> Threat modeling exercises enable the Rust Foundation and Rust Project to better understand the risks identified by the Security Audit. In developing all four threat models described below, we have consulted with the Rust Project's crates.io Team, Infrastructure Team, Security Response Working Group, and Secure Code Working Group, in addition to specific external stakeholders. We look forward to sharing the details for all threat models soon.

Rust Foundation

Focus 3:
Conducting Threat Modeling Exercises for the Rust Ecosystem

Progress:

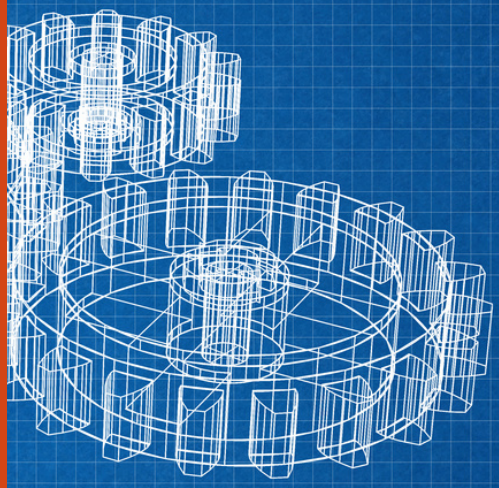
The Foundation is currently focused on the following four threat models as of this report:

The Crates Ecosystem – Threat model has been completed and reviewed by the Infrastructure Team, Security Response Working Group, and Secure Code Working Groups.

The Rust Project – Threat model outline has been developed and is currently under review by the Rust Project teams listed above.

Rust Infrastructure – An initial draft of the threat model and an initial assessment is complete and under review by the Rust Project teams listed above.

crates.io – A threat model outline is complete. The Rust Foundation and crates.io teams are collaborating on gathering information and conducting a security assessment based on the findings.



Focus 4:

Advocating for Security Best Practices Across the Rust Landscape

> Advancing the security of the Rust language hinges on the safety and security of its associated tools. By promoting security awareness and education within the official package manager and build system for Rust (Cargo), the central 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.

Progress:

crates.io Technical Debt Reduction

Since the arrival of the Rust Foundation's two full-time software engineers in Q1, we have been helping the Rust Project reduce the amount of technical debt accrued over time.

Rust Foundation Software Engineer Tobias Bieniek has been particularly focused on reducing technical debt within crates.io. Achievements include:

- Fixing the handling of releases with build metadata in their version strings
- Migrating to the "secrecy" crate to avoid accidental leakages of credentials and other secrets
- Introducing cargo-deny to signal possible vulnerabilities
- Updating the crates.io operations guide
- Simplifying the test suite in several ways
- Adding tracing information to many code paths
- Cleaning up broken crate files stored on S3

... and more. To learn more about Tobias' activities and accomplishments during his first three months at the Rust Foundation, please visit our blog.

We believe that these fixes and the ones to come will help contribute to a more secure and efficient crates.io.

API Token Improvements

Roughly three years ago, crates.io team member Pietro Albini created a proposal that would implement scopes for crates.io tokens, allowing users to choose which endpoints the token is allowed to call and which crates the token is allowed to affect. Due to a heavy load of work on the shoulders of the crates.io team, this valuable work was on pause for several years.

We are pleased to report that Tobias Bieniek was able to prioritize this issue and implement it. API token scopes and expiry dates on crates.io are now in production.

Focus 5:
Developing Tools, Features, & Recommendations Based on Security Research

> Since the inception of the Security Initiative, we have expected the results of our Rust security auditing and research to reveal the need for new open source tools and features to enhance maintainers' security workflows and unlock greater insight into vulnerabilities. While the first Rust security audit is not yet complete, we have already identified several urgent tooling needs and made progress toward their general availability.

Progress:

Public Availability of "Painter" Tool

We are proud to announce the first publicly-available tool in our Security Toolkit: Painter!

Painter is an open source project 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.

The tool is aimed at addressing issues and determining risks when using other tools (such as Cargo Audit). This allows users to not only determine if a vulnerable dependency exists but if the attack path is realized. Painter was created by Rust Foundation Security Engineer Walter Pearce and released for public usage in July 2023.

Crates.io Admin Console
A crates.io admin console has been developed. Once fully implemented, the crates.io Admin Console establishes more security guard rails for common operations and will advance the state of automation within the administration of crates.io.

Ecosystem Scanning, codename "Sandpit"
The team has been hard at work creating automated tooling and techniques for identifying possibly malicious crate activity on crates.io. Initial community discussion around the public surfacing of these results has commenced. The Rust ecosystem has not yet been the victim of a concerted effort to distribute malware among packages, but recent events in other open source package repositories have highlighted the need for such monitoring.

As the community aligns on how we want to surface this information, 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.

Focus 6:
Developing Documentation to Demystify Security in the Rust Ecosystem

> By outlining security risks, vulnerabilities, and considerations in 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, prevent common security pitfalls, and better educate the community about Rust's existing security benefits.

Progress:

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).

We have also initiated a community discussion around what detections to implement and how various levels of security guarantee should be surfaced to the user, including the aforementioned quarantine, with automated tooling.

Finally, the Rust Foundation team has begun interfacing with the Infrastructure team to begin documenting access control provisioning and de-provisioning within the Rust Project. We hope to aid them in documenting the process, as well as identifying gaps and areas for automation or improvement.

Focus 7:
Addressing Recommended Rust Security Issues Identified Through Research

> While the initial Rust security audit is not yet complete, our engineers tasked with researching the state of Rust security have already identified and prioritized the resolution of several security issues.

Progress:

With help from Rust Foundation Platinum Member JFrog, we identified a security vulnerability in the popular crate, Hyper. As we identify more security issues throughout the Rust ecosystem, we will report them through the proper channels, including the Rust Project Security Response Working Group as necessary.

Upcoming Areas of Focus

The initial focus of the Security Initiative has been on hiring, planning, and exploration, with some fantastic early-stage engineering achievements to build on. Throughout the remainder of 2023 and into 2024, our work will be focused on implementing solutions to serve the findings described in the threat models and other analyses.

The goal over the next year is to ensure that proactive measures have been taken to prevent potential threats and bad actors within the Rust ecosystem, while also supporting the technical and people capacity to quickly mitigate any active vulnerabilities that may arise. This endeavor will require continued coordination and collaboration with the Rust Project, appropriate Project teams, and a growing group of diverse stakeholders.

Over the coming months, the Rust Foundation team looks forward to further progressing toward these priorities and sharing details through reports like this one.

Recent Security Initiative press coverage highlights include:

DevOps.com — Rust Foundation Allies With OpenSSF and JFrog to Secure Code

SecurityWeekly — Solving Open-Source Security — from Alpha to Omega

SECURITY WEEK — INTERNET AND ENTERPRISE SECURITY NEWS, INSIGHTS & ANALYSIS — Rust Foundation to Identify and Address Security Defects in Rust Programming Language

THENEWSTACK — Rust Foundation CEO told State of Open Con Everyone Should be Paying to Secure Open Source

The Rust Foundation was also pleased to see the U.S. Department of Commerce's National Institute of Standards and Technology (NIST) cite Rust as an example of a "safer coding" tool as part of its Software Assurance Metrics and Tool Evaluation (SAMATE).

We are encouraged to see a large government body with global influence such as the U.S. Department of Commerce taking note of Rust's cybersecurity merits. This has been further buoyed by the recent White House National Cyber Security Strategy Implementation Plan, which stresses the benefits of memory-safe languages and the importance of adoption across critical infrastructure.

These developments are evidence that Rust is in a strong position to become even more key to performant and safe computer systems globally and that the work being carried out by the Security Initiative is both vital and timely.

Thank you for reading the first Rust Foundation Security Initiative Report for December 2022 - July 2023.

We look forward to issuing future reports detailing our work on these and other Security Initiative priorities.

Acknowledgments

In closing, we would like to thank the Rust Foundation's growing team of full-time software engineers, the leadership and participation of many key individuals and teams within the Rust Project, the generous investment from Alpha-Omega and AWS, and the support from Google, JFrog, & Wiz.

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

Rust Foundation

rustfoundation.org