Weekly progress reporting for OSTIF and Alpha-Omega collaborative work.

This is intended to be a continuous document that will be added to weekly. This creates a history of events and progress reports with full context and changes available.

Tuesday the 4th of November 2025

Statuses on Funding:

Proposal for io_uring is still pending AO approval.
https://docs.google.com/document/d/1Ggm4ZM2gehA97oJv1wyKX5nSSmQfWBrS_vhgJxbH0IU

This document is the final version and has been approved by the kernel maintainers.

Puercos First OSTIF Project:

Puerco (Adolfo Garcia-Veytia) has been boarded on to OSTIF and we are now working together to select his first big project, which is likely going to be Scala. This is because we are already in touch with the Scala maintainers to conduct an audit, and they'd tremendously benefit from his assistance.

1. It is a primer supply chain target as it is a very early component of other build processes
2. The repository uses modern pipelines techniques and buildchain which allows us to come in with the most up to date tools.
3. It has a fairly complex build process with various stages that can be protected.
4. It builds artifacts (ie it is not just a library or source release)
5. publishes to repositories (eg chocolatey/maven, etc)

Puerco is proposing to build out provenance metadata attestations, build automatic SBOM generation into the build system, and dramatically improve supply chain security at multiple levels by applying SLSA compliance steps and fixing vulnerabilities in the present build system.

This work is not directly funded by Alpha-Omega, however, the volume of work that Alpha-Omega collaborates with OSTIF on allows us to offer Puerco's services gratis to all eligible projects. This work greatly extends our reach and impact on our projects moving forward without increasing costs.

Full Audits:

Paramiko/Cryptography/Rust-OpenSSL bindings - Auditing has completed and the projects are in the remediation phase. The audit report handed to the project is over 100 pages. The vast majority of the issues are info/low and no critical issues were discovered. We are assessing timelines for fixes now.

LLVM BOLT based binary scanner - Research is ongoing. The first tests are coming together now. Alexandra from Google has created a test that seeks indirect branches located in the wrong portion of the cache line (Looking for CPU prefetch issues, which is what Google is working on). Quarkslab has started their work on memory tests.

PyTorch/Executorch - Trail of Bits will begin their audit in January. This is a changed timeline due to the availability of a key engineer on the ToB side. We have already had an intro call with the Pytorch and Executorch maintainers and they know what to expect and the related timelines. They also gave us some additional information about blocks of code that are new and need additional scrutiny, as well as one component that they are less confident is stable/secure.

Requests/CacheControl/urllib3 - 7A security audit is ongoing.

Mbedtls - Quarkslab final agreement needed minor revisions. Expected to be resolved this week and for the audit to be scheduled for roughly December 2025 going into 2026. (At the request of the mbedtls.) Mbedtls 4.0 launched last week with a lot of new code and breaking changes.

vLLM - Finalized the project plan with X41 this week. They will use AI to help build codeql rules and fuzzing harnesses for the project, as well as employing some code analysis with a senior security expert validating the potential results. We are instructing the audit team to upstream any improvements that they make to the open source tooling that they use, ensuring the largest community benefit. We have opted not to integrate the tool into the vLLM CI, as it still requires a lot of security expertise and current tooling produces a lot of false positives which maintainers will quickly choose to ignore. We plan to offer CI integration to a number of projects that are backed by companies and have dedicated security staff on hand to help deal with the flow of potential issues.

Langchain - Langchain project plan is final and we are going through the audit team selection process.

Rapid Security Assessments:

The rapid security assessments program has launched, with the 58 projects in the first tranche being evaluated now. This process will help us discover the actual time required for each project with the criteria that we’ve developed. We will adjust the program based on these initial results, and likely adjust it further as automation for these processes improve. We are looking into all options for making this process as efficient as possible.

The spreadsheet for data gathering has been finalized. The data processing backend and Looker Studio dashboard are being developed now with an ETA of 2-3 weeks.

Our Ongoing AI / LLM Research in Unrelated Projects:

We are applying some AI / LLM based tooling in our audits to evaluate their efficacy in the hands of an expert, and see if it improves the efficiency of their work, their findings, both, or neither. We will have some results on this in the coming months as these AI-assisted audit workflows are assessed.

Additionally, we have proposed a significant amount of research with DARPA and OpenSSF to evaluate the bug reports, fixes, and harnesses created by the AIxCC competition, and if approved, we will have significant data on the usability of Cyber Reasoning Systems to automatically create fuzzing infrastructure, find issues, and submit fixes to projects that will pass the CI/CD testing pipeline. This will give us extremely valuable data on how we can find and fix bugs at scale without much maintainer work required, similar to the original vision of Omega.

Some hard data on this topic will greatly influence our work going forward.

As always, we are open to suggestions on things that we should investigate in the space, as it is fast moving and there are new tools and techniques daily in this space.

Small note: Tuesday the 28th of October had no significant updates and didn't justify weekly reporting.

Tuesday the 21st of October 2025

Statuses on Funding:

Funding has been received or is confirmed in processing for all approved projects. All projects moving forward. Proposal has been created for io_uring at the request of the linux kernel team. It has the blessings of the kernel maintainers.
https://docs.google.com/document/d/1Ggm4ZM2gehA97oJv1wyKX5nSSmQfWBrS_vhgJxbH0IU

I’m also attaching the current draft to the email that accompanies this document. The hours and costs are likely final, the changes are minor edits to scoping focus and general accuracy.

Welcoming Puerco to OSTIF:

Puerco (Adolfo Garcia-Veytia) is being hired by OSTIF as a CI/CD, supply chain, and provenance expert. OSTIF will offer his services (gratis) to all projects receiving audits from us going forward. Additionally, Puerco will be reviewing the CI/CD of the current suite of audits with AO. This greatly expands the scope of our work and allows us to do a lot more to help projects with their security. Puerco will also be key in helping us develop frameworks and guidance for how to implement AI tools securely in a CI/CD pipeline. Finally, he is also reviewing the rapid assessment program draft to see if there are any CI/CD / supply chain elements that we should add to the scope.

Full Audits:

Paramiko/Cryptography/Rust-OpenSSL bindings - Auditing has completed and the projects are in the remediation phase. The audit report handed to the project is over 100 pages. The vast majority of the issues are info/low and no critical issues were discovered. We are assessing timelines for fixes now.

LLVM BOLT based binary scanner - Research is ongoing. The first tests are coming together now. Alexandra from Google has created a test that seeks indirect branches located in the wrong portion of the cache line (Looking for CPU prefetch issues, which is what Google is working on). Quarkslab has started their work on memory tests.

PyTorch/Executorch - Trail of Bits will begin their audit in January. This is a changed timeline due to the availability of a key engineer on the ToB side. We have already had an intro call with the Pytorch and Executorch maintainers and they know what to expect and the related timelines. They also gave us some additional information about blocks of code that are new and need additional scrutiny, as well as one component that they are less confident is stable/secure.

Requests/CacheControl/urllib3 - 7A security audit has started.

Mbedtls - Quarkslab final agreement needed minor revisions. Expected to be resolved this week and for the audit to be scheduled for roughly December 2025 going into 2026. (At the request of the mbedtls.) Mbedtls 4.0 launched this week with a lot of new code and breaking changes.

vLLM - Finalizing the agreement with X41 this week. They want to apply some AI tests to the project, and we are negotiating how maintenance of such a tool would work if it was integrated into their CI, as it would have ongoing maintenance challenges as well as ongoing compute costs.

Langchain - Finalizing proposal with langchain this week.

Rapid Assessments:

The proposed rapid security assessment program is going through its final review phase now with all of the elements in place. The data entry form is also near-final with processes in place with input validation and the dashboard is being developed in Google Looker Studio. Will link when it is ready.

The draft version of this program is here: (near final)
https://docs.google.com/document/d/1NoT2NrAwhy4pP4-FF-icZQyvNyjDjsiIJDKC55HGqoE

The metrics will be input into a spreadsheet here: (work in progress)
https://docs.google.com/spreadsheets/d/1mVDrwnAhKV9WIJpHt1Gu-QALI1670zdvA9JJlsNviqw

We have a full list of maintainer channels to contact with a form letter notifying them of our activity and what to expect for their project.

A team has been selected and has already begun the initial scouting for the assessments.

A Primer on AI / LLM Based Research for Security:

OSTIF is continually researching new AI-based tooling and looking for ways to improve the efficiency of our work. We have multiple ongoing projects related to this, and some preliminary findings that we can share.

General thoughts on usage for audits.

LLM Agents that run tooling and just give you the outputs are nice - Using LLMs to apply a battery of testing to projects in an automated fashion and generate a report is nice. It is one step above scripting in that it can adapt to languages and the specifics of what a project does if your prompts are tight enough, and then employ agents to run the scans and report. This saves a few hours off of every engagement.

LLM tools are bad at triage - They find and validate issues where there are none because they don’t understand context. “This function can overflow” “It has trusted input because it’s a wrapper for malloc.” “This function can overflow” “...”

Using LLMs to find classes of bugs that are very difficult to find in complex scenarios works, but requires a lot of manual validation and triage by humans - They seem to be particularly well suited for finding classes of bugs that often aren't found by SAST/fuzzing like TOCTOU issues, race conditions, auth bypass. In this case it's not a time saver, but an extension in capability.

We are currently working with DARPA on the AIxCC Cyber Reasoning Systems - The AIxCC Challenge gave us a number of open source tools that leverage LLMs to create fuzzing harnesses, find issues with those harnesses, and then create fixes for the issues that were found. This is a very labor intensive process currently, and if this can be automated it would be a massive increase in efficiency for one of the more difficult to implement types of testing. We are currently researching the efficacy of the tooling, as the findings that happened during the competition didn't match the findings that OSTIF gets from our typical fuzzing engagements. This could be because the AI tooling created bad harnesses sometimes, was covering irrelevant code, or was fuzzing redundant areas of code that overlapped with other harnesses. This research will ultimately give us a greater understanding of whether these tools require a lot of human intervention presently, if there are significant improvements that can be made to the AI tooling, and ultimately, we can make a reasonable cost-benefit analysis on utilizing these tools over the long term.

Additionally, we are looking at creating frameworks and guidance on how open source projects can safely integrate AI tools into their CI/CD pipelines. This would allow for AI orchestrated workflows to happen without introducing new vulnerabilities.

And finally, we are looking to add an AI "assurance layer" to CI/CD implementations, where it can check for flaws in the CI/CD pipeline and warn maintainers if something is misconfigured or otherwise vulnerable. It would look like a "dependabot for CI/CD maintenance."

We are also open to new ideas on this frontier, as this is a very fast moving area right now and new capabilities and new problems are surfacing daily.

Tuesday the 14th was skipped due to schedule conflicts.

Tuesday the 7th of October 2025

Statuses on Funding:

Funding: Funding has been received for the primary tranche of projects. We have not received the funding for the Ruby projects yet, so they are on hold. All other projects are moving forward.

Full Audits:

Paramiko/Cryptography/Rust-OpenSSL bindings - Auditing has completed and the first draft of the report is expected by the end of the week. If remediation is required, we will have rough estimates on that next week.

LLVM BOLT based binary scanner - Research is ongoing. Will elaborate on major developments as they come together.

PyTorch/Executorch - Trail of Bits will begin their audit in approximately two weeks. We are setting up an intro call with the PyTorch maintainers, whom ToB already has a working relationship with.

Requests/CacheControl/urllib3 - 7A security has set up their initial meetings with CacheControl and urllib3 for their audit. We are instructing them to create one report for this to avoid duplicating work on documentation and to give them the maximum amount of time to find issues.

Mbedtls - Quarkslab final agreement needed minor revisions. Expected to be resolved this week and for the audit to be scheduled for roughly December 2025 going into 2026. (At the request of the mbedtls.)

vLLM - Finalizing the agreement with X41 this week. (Was intended to be done last week, but their proposal to us needed some edits and clarifications on scope.)

Langchain - Working with langchain to define the final audit scope. They have some new issues that they are triaging now. (This thorough audit can’t come soon enough for them.)

Rapid Assessments:

The proposed rapid security assessment program is going through its final review phase now with all of the elements in place. The data entry form is also near-final with processes in place with input validation and the dashboard is being developed in Google Looker Studio. Will link when it is ready.

The draft version of this program is here: (near final)
https://docs.google.com/document/d/1NoT2NrAwhy4pP4-FF-icZQyvNyjDjsiIJDKC55HGqoE

The metrics will be input into a spreadsheet here: (work in progress) https://docs.google.com/spreadsheets/d/1mVDrwnAhKV9WIJpHt1Gu-QALI1670zdvA9JJlsNviqw

Our OSTIF project manager has gathered all of the maintainer information and we are building a form letter on thursday to notify projects on our activities and ask them the very short questionnaire that we have for each evaluation.

A team has been selected and has already begun the initial scouting for the assessments.

Tuesday the 30th of September 2025

Statuses on Funding:

Funding: Funding has been received for the primary tranche of projects. We have not received the funding for the Ruby projects yet, so they are on hold. All other projects are moving forward.

Full Audits:

Paramiko/Cryptography/Rust-OpenSSL bindings - Final phase of auditing being done by the assurance team at Quarkslab.

LLVM BOLT based binary scanner - Research is ongoing, the team thinks that they have some easy targets to extend the tool and we're starting with those to get some easy wins under our belt. Then they're going to try to tackle something harder like stack clash checks that can have multiple anti-patterns that lead to vulnerabilities.

PyTorch/Executorch - Agreement was signed today with Trail of Bits and scheduling is underway.

Requests/CacheControl/urllib3 - Agreement was signed today with 7A Security to move forward with the review. We've been working closely with the Python Software Foundation to ensure that we are taking a closer look at how these projects are used by pypi (the pip client and the warehouse server). We have configuration information for how cachecontrol is specifically configured in the warehouse for evaluation. Scheduling is under way.

Mbedtls - Finalizing the agreement with Quarkslab this week. Mbedtls has requested some additional time to complete their migration of the project into two distinct projects. They are working to separate the functionality into two separate libraries to reduce the attack surface of each, as applications rarely need ALL of the features of mbedtls. Estimates are to begin the audit in December.

vLLM - Finalizing the agreement with X41 this week.

Langchain - Working with langchain to define the final audit scope. Agreement pending soon.

Rapid Assessments:

The rapid assessment methodology is being finalized now. (We are running this by our security advisory council and a few open source maintainers who are contributing.)

We are also going to make a proactive effort to reach out to all of the projects to let them know about the rapid assessment program and that they will receive information about how their project fared with some gentle recommendations to improve their assessments. We will also ask them five simple questions for metrics that we cannot see by just viewing the git repo and CI/CD pipelines of the project.

This both allows us to community build and to check if maintainers respond for projects that we suspect may not have dedicated maintainers.

The draft version of this program is here:
https://docs.google.com/document/d/1NoT2NrAwhy4pP4-FF-icZQyvNyjDjsiIJDKC55HGqoE

This focuses on multiple goals - namely identifying risk factors for projects to determine if proper safety controls are in place, and applies some cursory review of the code with tooling of the auditors choice (SAST that isn't apparently run in the CI/CD of the project currently, and is unlikely to be used on the project regularly). It then takes a large number of metrics about each project in the list, and draws a number of interesting statistics about the perceived "riskiness" of the project based on those metrics.

The metrics will be input into a spreadsheet here: (work in progress)
https://docs.google.com/spreadsheets/d/1mVDrwnAhKV9WIJpHt1Gu-QALI1670zdvA9JJlsNviqw

Which will feed into a live Looker Studio dashboard that will track trends and statistics for the entire tranche of projects.
(Coming Soon)

From here, we will develop a “risk score” system based on the results of these engagements, and going forward will have a reasonable evaluation criteria, a stronger understanding of where typical projects lie, and can work together with security experts to develop a good methodology for scoring. This scoring system can be used in future engagements.

We do not intend to publish this work, counter to our normal work at OSTIF, for multiple reasons.

1. These assessments will make specific open source projects look bad, with no immediate action being taken to assist them further.
2. Projects may have vulnerabilities discovered that are not yet resolved when the assessments conclude.
3. Publishing these results COULD help the overall community, however, these assessments are opinionated and could raise debate/controversy about OSTIF.

The best course of action after this assessment takes place is to act to help the projects that need the most assistance, either through grants directly to the open source project to make security uplifts, grants to the open source project to fix issues, grants to OSTIF or a similar entity for full audits of their source code, or seeking other ways to uplift projects with support. AFTER that assistance takes place, the work should be published. This puts the actions in a much better light, as it includes directly assisting the project with improvements, rather than “doing a drive-by and throwing security issues over the fence” as open source maintainers so eloquently put it.

Potential changes to the project:
The SAST metrics will be fleshed out.
A metric that measures whether the community is growing or shrinking in the past year will be added.
There will be a standard bug tracking system for things discovered during a rapid assessment.
The Looker Studio tracking will be built.

Ruzzy and the Ruby Project Series:

Ruzzy has not been funded by Alpha Omega, but the four ruby projects rely on the integration of Ruzzy into ossfuzz in order to complete the four follow-up assessments.

The Ruzzy integration is already under way, but two issues have been raised which are now being worked on.

1. ASAN has some issues with how Ruby manages memory in a unique way, leading to false positives. We may be able to modify ASAN to disable some tests and reduce this false positive rate.
2. Ruzzy does not have any current method of coverage reporting. The team is working to build out coverage reporting tools (ideally getting introspector working, alternatively, creating a code coverage tool that gives users understanding of the coverage they are getting from their harnesses.)

Once Ruzzy is in a good state, the four Ruby projects (Ruby core, Ruby FFI, carrierwave, and psych) will be boarded onto ossfuzz by the same team, who will watch for false positives from ASAN and any other challenges that may arise with the new ossfuzz integration of ruzzy and adapt the tooling to ensure the best balance between usability and meaningful security findings.