Security
@
Open Source w/ SLU Roadmap

Plans for Spring 2026

Introduction

This is a general roadmap for what our expectations are and what goals we should complete by given time frames. This semester is broken up into 6 biweekly sprints that extend through the semester. Out of these 6 sprints, they are packaged into 2, 3-sprint chunks called Iterations, and these iterations will be what we plan the majority of our work around. We will also have ongoing projects that we will aim to contribute to throughout the semester as well as external resources that we may contribute to as time allows. While we may not have as clean units of work to submit at the end of each sprint, it's still expected that we are measurably closer to our goal at the end of the 2-week period.

This document will be broken up into sections with the title of the section being the period of work, followed by the task to be completed in a key-value pair. Subsections describing the work, what the need is (the "why"), and the expected date of completion will follow. This document is not binding and circumstances may change that may require certain deadlines to be altered. This document will not be edited to reflect such changes should they arise, but changes to deadlines will be discussed and addressed publicly as to avoid confusion.

Iteration 1: Github SAML

Description

On December 9th, Security @ Open Source w/ SLU was tasked to investigate what impact enabling SAML (Security Assertion Markup Language) Authentication on Github would have on Open Source w/ SLU. Mainly, the impact analysis will focus on balancing SLU's concern of security and access control and Open Source w/ SLU's concern of maintaining outside contributions and remaining an Open Source compliant organization. This will involve making sample accounts for testing both inside and outside contributors, and finding a series of settings, if there are any, that would enable Open Source w/ SLU to achieve this balance.

Need

Saint Louis University over the years has been on the receiving end of some very high-profile cybersecurity attacks and has ramped up efforts to ensure that student accounts and organizational assets are handled safely and securely. This includes enabling our secure single sign-on (SSO) Okta on platforms where organizational assets are regularly handled by

SLU students. Regardless of our actions, SLU Information Technology Services (ITS) will mandate SSO for Github organizations within the Saint Louis University Enterprise, which would include Open Source w/ SLU. This sandboxing and testing will give us vital insight into what the impact is and how we can continue to accept contributions from external users, otherwise we will no longer be Open Source compliant.

Estimated Time Frame

We are aiming to require SAML Authentication by March 11. Therefore, we need all relevant information and findings by March 6, to give time for configuration of Enterprise and Organizational security settings.

Iteration 2: Security Documentation

Description

Security Documentation is a loose term that describes a series of documents that provide insight to an organization's security posture. They are sometimes used for audits, but their real power comes in during the incident response lifecycle, especially in planning and remediating cybersecurity incidents. Last semester, each project in active development made a Threat Model for their product. It is our job to review these and advise our Tech Leads on where they may need additional support, in addition to ensuring that they are up to date. We are also looking to add documentation articulating a Cybersecurity Framework, which may be implemented this semester (outlined in Stretch: Implementing a Cybersecurity Framework).

Other documents include a Security Controls Inventory (SCI) and an Incident Response Playbook (IRP). A SCI is used to document what security controls are available to an enterprise and what can be deployed to protect, detect, contain, eradicate, and recover from a security incident. An IRP on the other hand is a set of steps to do in case of a specific alert or incident, taking guesswork out of incident response to allow for ease of remediation.

Need

Cybersecurity knowledge and protection at the enterprise level should not be gatekept between every employee. Knowledge needs to be passed down and preserved so that future security teams can build off of what others had created rather than rebuild. We cannot protect

Return to top

against the threats of tomorrow, and if we do not leave behind artifacts of what we have learned and why, future teams will be stuck trying to piece together our security systems, which may leave critical gaps undiscovered, and a team too focused on recreating what we've already made to deal with the security landscape. In addition, we need to prepare records and documents should an organization want to provide funding to Open Source w/ SLU and require proof of security, or wish to conduct an audit. These documents will be vital for auditing and record keeping of what we've done and where we should focus in the future.

Estimated Time Frame

Security Documentation work can start early, but Iteration 1: Github SAML must take priority. Expect to begin working on this fully on March 16, and this should carry mostly through the end of the semester. If this task is completed in full before the end of the semester, there will be other work we can do in the meantime.

Ongoing: Security Alert Clearance

Description

As one of our chief responsibilities we are tasked with triaging and managing security alerts and advisories. For alerts, we have 3 main types: Dependabot, Code Scanning, and Secret Scanning. Dependabot is the least problematic, and mostly requires an update to the dependencies. Code Scanning is more severe: the AI code scanner detects code smells that are typical of a specific vulnerability. While more problematic, these are far from gospel, and should be investigated before being patched. There are patches generated by Copilot that will automatically fix the issue, but talk with the tech lead before implementing. Finally, Secret Scanning alerts are created when something that looks like a secret (private key, token, password, etc.) is included in the repository. These are almost always serious and should be investigated.

Managing these alerts is how we keep on top of the security posture of our products. In addition, if an advisory is submitted via our repository (i.e. if a security researcher or a member of the security community discloses a vulnerability), we are the ones who must triage it and respond to it with the assistance of the development team for the specific project. Should this

Return to top

arise, it may be acceptable to pause work to deal with it. Although, this is highly dependent on the circumstances.

Need

Managing alerts is always a need. We have eclipsed 1000 alerts and the number keeps on rising. However, the vast majority of these alerts are Dependabot alerts that can be fixed with a simple patch. Also, projects not currently in development are not a priority to fix, with the exception of the oss_slu website as that is front-facing (although how vulnerable a static website is questionable). Regardless, the constant number rising still should be a fire that we must act on or else it may get to be too much. We especially cannot let alerts that are easy enough to patch and deal with to fester.

Estimated Time Frame

This is an ongoing task with no beginning or end date. Progress will be measured by the number of alerts. While it's unfeasible to be at 0, being under 5 alerts/active repository should be an achievable standard if time allows.

As Needed: Security Consultation for GradEval360

Description

GradEval360 is a new project that Open Source w/ SLU was tasked to create to help facilitate graduate students having regular meetings with their advisors. It's designed to fill a niche that SLU identified and it could provide a future source or revenue for the program and the University at large. Part of the requirements will be integrating SAML Authentication through Okta (similar to Iteration 1: Github SAML), and role-based access control (RBAC). While we have not been assigned to assist yet, the project could be one that the security team could be asked to assist with, especially as it goes through approval from SLU ITS. Contributions to this project can also be made as a stretch goal (see Stretch: Contributing to Open Source Projects).

While tasks have yet to be officially identified, a Threat Model will likely be required at some point which can be completed in conjunction with Iteration 2: Security Documentation. In addition, we will be able to provide some level of auditing to ensure that it meets muster the first time SLU ITS investigates it. The project's goal by the end of the semester is a minimum viable product, so we may need to work hard and fast if called upon.

Return to top

Need

The importance of this project cannot be stressed enough. This singular project was identified as important enough strategically for Open Source w/ SLU, that work on a project was cancelled for the semester because the Tech Lead for this project dropped from the program. It's so important that there was (joking) consideration to drop cybersecurity to move development on this project. So while our work may be beneficial to the future of the organization, if we're requested to work on this project, chances are this may affect the bottom line of the University, so we may need to drop everything and help.

Estimated Time Frame

The goal for this project is a minimum viable product by the end of the semester that can get tied into Okta and begin being utilized as soon as possible. For estimating when RBAC and Okta integration comes in, it's hard to judge, but best estimates would place that after Spring Break. More info on this project will come as it is available.

Stretch: Implementing a Cybersecurity Framework

Description

The NIST CSF is a voluntary set of standards many organizations implement in order "to understand, assess, prioritize, and communicate its cybersecurity efforts" (NIST Cybersecurity Framework). More or less it's an enterprise security technique to standardize various security practices into one cohesive unit. The NIST framework consists of various sections, including Govern, Identify, Protect, Detect, Respond, and Recover. While we meet some of these pieces, organizing it all and packaging it under the framework could be vital to not impeding cybersecurity management. There are additional resources and other frameworks that could be implemented, but the NIST CSF is the baseline standard.

Need

As of now, it's difficult to justify the need for the framework as we have other major projects to work on. But as we begin documenting our security posture, taking the time to do it under the framework could help us better understand coverage gaps and where we can better focus our time on solving. The need in this case is not necessarily that inaction will lead to

Return to top

vulnerability, but better ease of use should always be strived for. That, and the fact that future security teams won't curse our names for failing to articulate anything in a clearly defined pattern. Whichever means of motivation we prefer, a little today can go a long way tomorrow.

Estimated Time Frame

There's not necessarily a "best time", but we need some documentation before we begin implementing a security framework, so during or after Iteration 2: Security Documentation would likely be best.

Stretch: Contributing to Open Source Projects

Description

One of the major responsibilities of Security @ Open Source w/ SLU is the contribution to the security of the wider Open Source community. An easy way, and the main way we've done this in the past, is through contributing to the advancement of widely used, open source security tools. We've worked with OpenSSF in the past on their project Gittuf, and if possible it'd be nice to continue to work with them. But that doesn't necessarily need to stop with just security tools. As security conscious individuals, we have the ability and the capability of looking into open source projects and making security contributions, both within our organization (see As Needed: Security Consultation for GradEval360), and beyond.

Need

We one day may benefit from external security contributors, so building a healthy relationship with the wider Open Source community is not only a good thing, it makes all of software development safer. Open Source projects are widely used across enterprise code bases, and being able to be a part of the community can increase the reputation of our organization, not just for other open Source developers, but maybe even some large enterprises. That being said, it will not directly affect our security posture, nor advance needed organizational goals, so should you have free time, and a desire to overperform in the class, consider helping out.

Return to top

Estimated Time Frame

Even though it’s a part of our mission, there is no expectation surrounding when this gets done nor if it gets done (outside of any class requirements). That being said, if you do wish to contribute, it should likely happen after Iteration 1: Github SAML is complete.

Return to top