GitHub for Security Testing

Steve Swiss
Solutions Architect

Agenda

- Problem Statement
- Approach
- Demo - Webhook for Repository Created events
- Alternative Approaches
- Future Opportunities
- References

Problem Statement

- Must Haves
  - Need to test candidate code updates for security issues
  - Testing should be automatic (not depend on developer or tester manual process)
  - Testing should be easy to scale for many repositories
  - Must capture new repositories as they are created, and enable security testing
- Other Opportunities
  - Dynamic security testing for major releases (pen testing, etc.)

Approach

- Use GitHub advanced security testing features (vulnerability check, secrets, etc.) on COMMITs to protected branches
- When new repositories are created
  - Use GitHub Webhooks and simple listener to update repo to enable advanced security using repository API
  - Check for README (indicates that repo has at least one branch)
    - If no README, UPDATE repo by pushing default README)
  - Update ACTIONS yaml file to reflect code languages

Demo - Webhooks Listener

- Scenario: A New repository is created in an organization. A web hook POST is sent to a listener, which then sends a PATCH request to the new repo to add security testing.

- Steps:

  - Show how a GitHub Organization is configured to generate web hooks when a new repository is created

  - Show how a web hook POST is sent to a web hook listener URL, which parses it to confirm it is for a new repo

  - Examine the code to use the Github Repository API to update the security code checking policy for the new repo.

- Code and README: https://github.com/SMSwissGitHubDemo/DemoSolution/blob/main/README.md

Alternates to GitHub Advanced Security

- If you have existing security testing resources you prefer:
  - Use existing testing resources when COMMITs to protected branches occur
  - Can use GitHub Actions in place of custom web hook listener
    - Easy to configure via yaml file to define testing
    - Runs on separate platform - does not burden dev platforms

Future Opportunities

- Adding dynamic security testing for major releases
  - Start with GitHub Actions alternative to static testing, but trigger on RELEASE event
  - Target code can create test environment (Docker, e.g.) and initiate third party DAST

References

- GitHub Repository for this demo: https://github.com/orgs/SMSwissGitHubDemo/repositories
- GitHub Webhooks, and setting up local listener server: https://docs.github.com/en/developers/webhooks-and-events/webhooks/about-webhooks and subsequent pages. The Ruby local server using ngrok is easy to set up and run.
- GitHub Actions: start with https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions#overview and subsequent pages for configuring a repository to send events, and how to set up workflows when events occur
- GitHub Advanced Security: https://docs.github.com/en/get-started/learning-about-github/about-github-advanced-security
- Dynamic security testing (DAST): https://en.wikipedia.org/wiki/Dynamic_application_security_testing for a Wikipedia intro to the topic. We can provide recommendations for third party DASTs if needed.
- CVE website: https://cve.mitre.org for known cybersecurity vulnerabilities, community sourced and maintained
- NVD website: https://nvd.nist.gov maintained by the US government, and is synced with CVE