Shifting Left: A Hands-on Introductory Guide to DevSecOps

MODUS CREATE

MODUS CREATE

 Before We Start - WiFi Instructions

BSides SF provides a wireless network named “BSidesSF-Static” which allows visitors to self-register for Internet access.

Follow these steps to complete the registration process.

1. Turn on your computer and enable your wireless network card
2. Select BSidesSF-Static from the list of available wireless networks
3. Password = bsidessf

MODUS CREATE

🏴‍☠️ Introduction to the workshop

- 💣 Team Intro:
  - 💣 **Andy Dennis** - VP Platform and Cloud
  - 💣 **Bill Reyor** - Director of Security 😷
  - 💣 **Dwayne McDaniel** - (GitGuardian) Stepping in to help today!
- 💣 Intro Modus Create:
  - 💣 Digital Transformation Consultancy based in D.C area
  - 💣 Platform and Cloud Practice has a focus on DevOps, DevSecOps, Cloud Infrastructure, IDPs,Security Assessments, Build Systems, SecOps and similar
- 💣 Intro course structure:
  - 💣 We'll be dividing the workshop today into two parts. A small 5 min break will be between part 1 and part 2.
  - 💣 The course will cover a variety of topics including security in your CI/CD environment.

MODUS CREATE

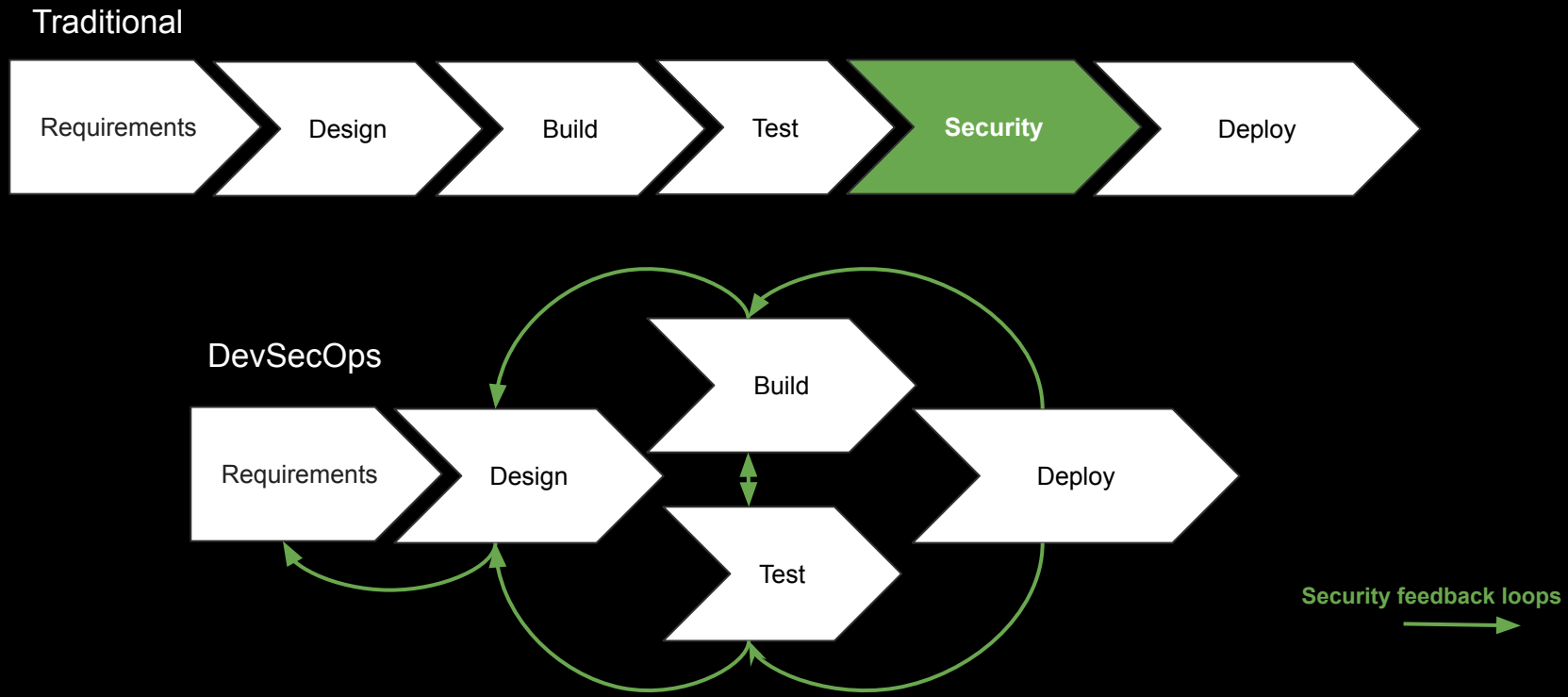
 Assumptions About You

To get the most out of this workshop, you should have

- **Technical Knowledge**
  - Familiarity with software development concepts, tools, and processes
  - An understanding of the importance of integrating security early
  - Comfort with hands-on learning and an interactive workshop format
- **Experience with GitHub**
  - A GitHub account and basic knowledge of forking repositories
  - Ability to navigate and use core GitHub features and workflows
- **Flexibility to explore new tools**
  - Willingness to setup and experiment with tools like GitHub Actions
  - Openness to following along and troubleshooting in your GitHub environment

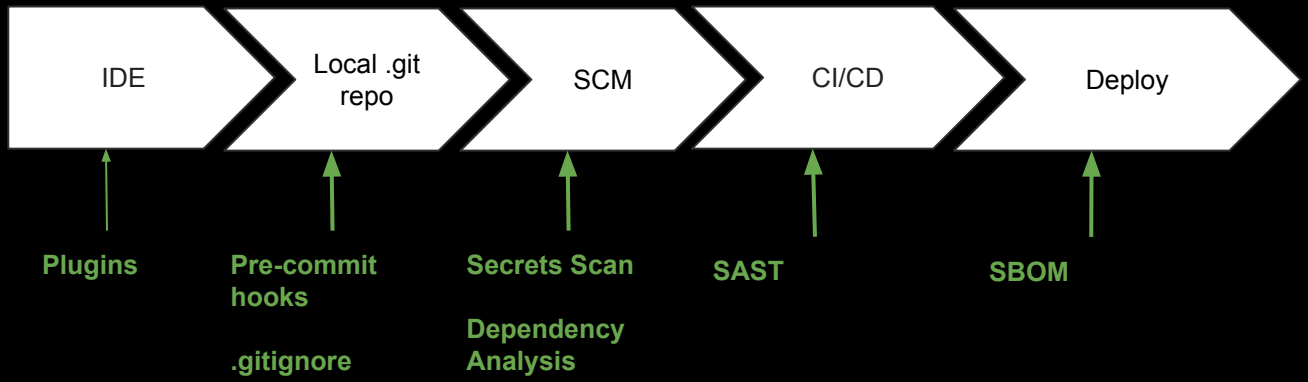
MODUS CREATE

What is DevSecOps?



MODUS CREATE

What is Shifting Left?



Security is constantly pushed further left in SDLC.

🏴‍☠️ Getting setup - repository

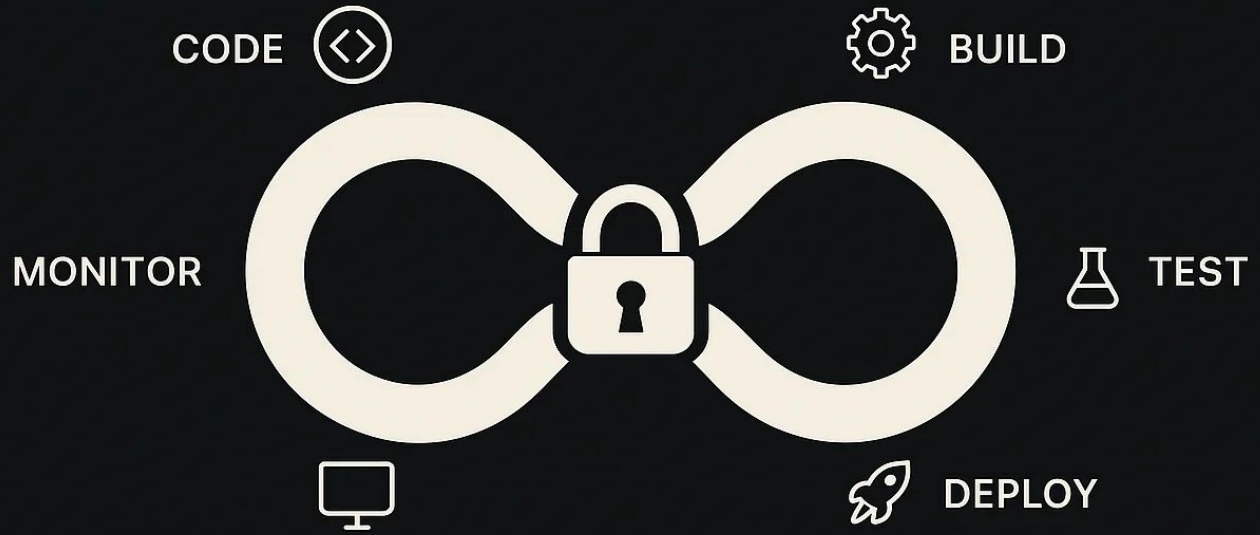
We have setup a repository for you to work with during this course.

- 💣 Located at Modus Create repo
  https://github.com/tweag/bsidessf-hands-on-devsecops-2025

- 💣 Fork it
  https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/working-with-forks/fork-a-repo

If anyone needs help, raise your hand and we will stop by.

DevSecOps



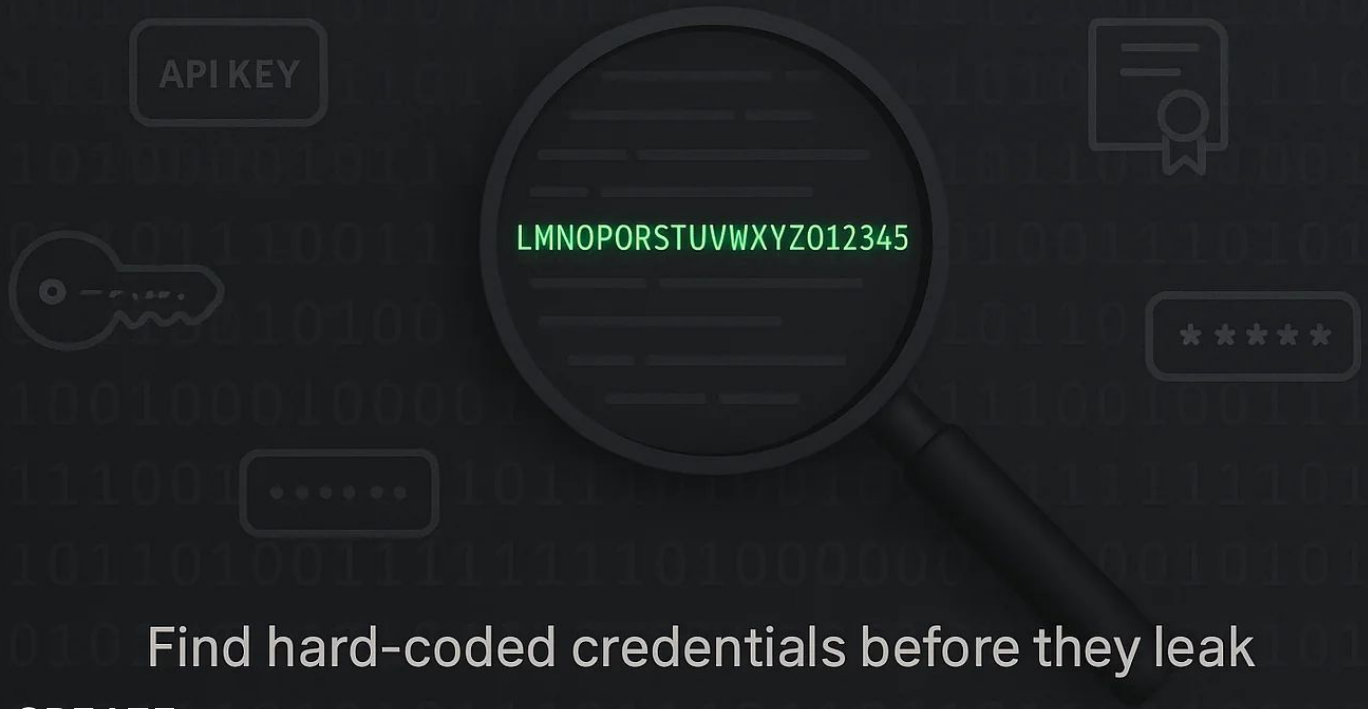
Security in the CI/CD pipeline

Welcome to Part 1

In this section of the workshop we will cover:

- Secrets scanning
- Handling secrets in GitHub
- Detecting security vulnerabilities in the repository
- Vulnerable dependencies
- Static Analysis (SAST)
- Rule sets

💀 SECRETS SCANNING 💀



Find hard-coded credentials before they leak

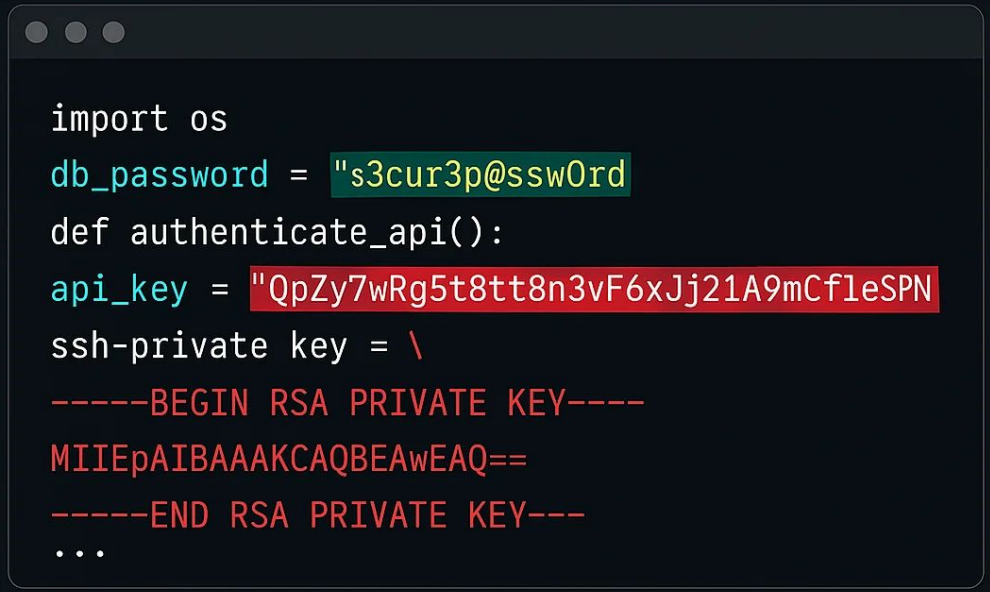
Secrets Scanning

We are now going to look how we can handle secrets scanning in GitHub, before looking more broadly at how to handle vulnerabilities.

Useful Documentation

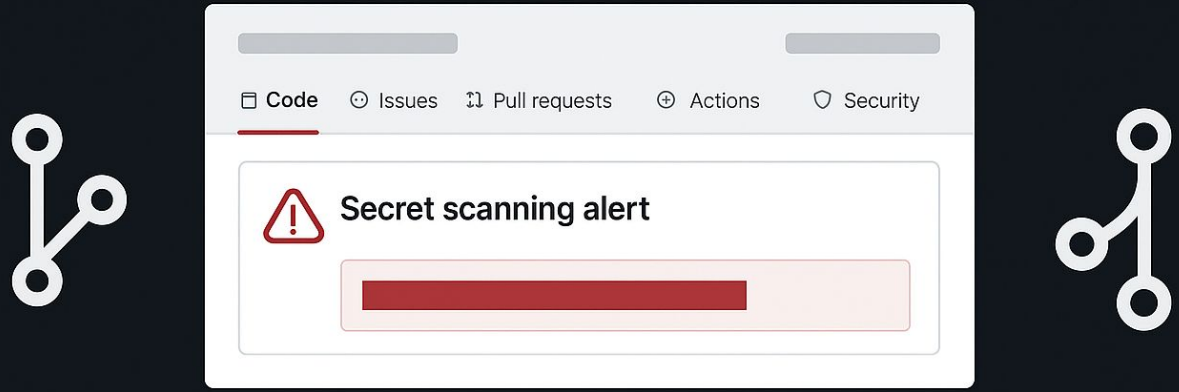
- TruffleHog: https://github.com/trufflesecurity/trufflehog
- GitHub Secrets Scanning: https://docs.github.com/en/code-security/secret-scanning/about-secret-scanning
- Horusec: https://horusec.io/site/

Demo — Secrets Scanning



Watching for hard-coded credentials in real time

💀 Handling Secrets in GitHub 💀



Handle secrets securely before merging into main

Secrets in GitHub

So in summary some best practices for handling secrets in GitHub include:

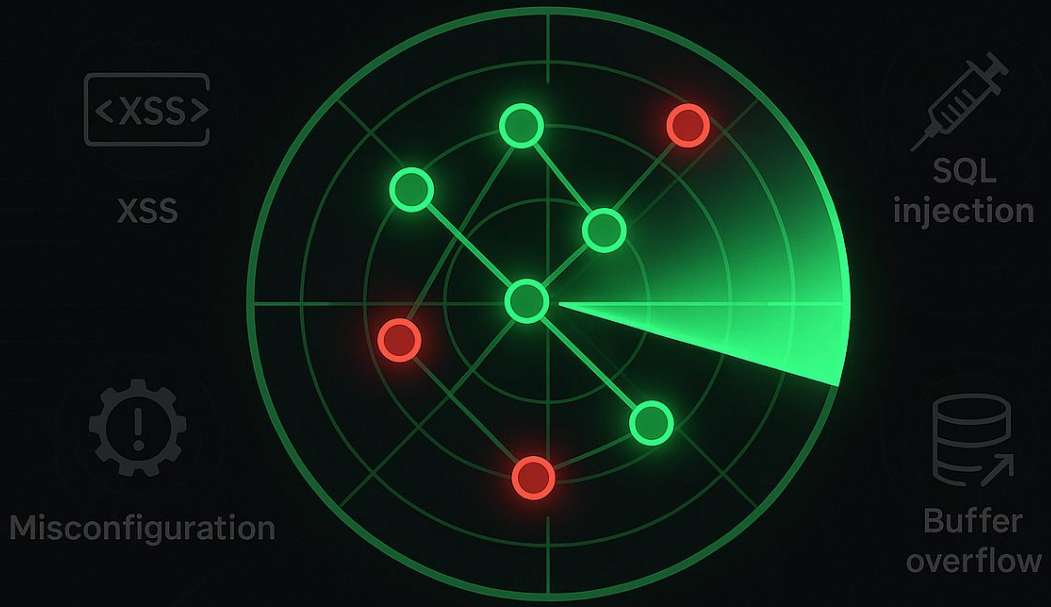
- **Prepare** - Use a secret manager (GitHub has one)
- **Prevent** - GitHub native paid tooling blocks commits
- **Detect** - pipeline checks, we can also use TruffleHog
- **Respond** - Rotate and remove (BFG / Git-filter) - not covered in this workshop

Let’s look at secrets storage in GitHub quickly

Demo - Handling Secrets

MODUS CREATE

Detecting Security Vulnerabilities



Identify weak spots before attackers find them.

Detecting security vulnerabilities

Multiple tools for detecting vulnerabilities.

Secrets scanning - pattern/entropy detection

SAST - Static Analysis Security Testing

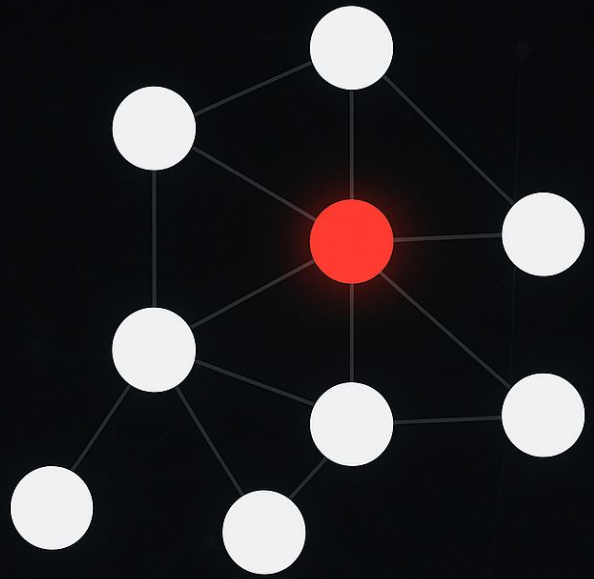
SCA - Software Composition Analysis

SBOMs - Software Bills of Materials

GitHub has the following features:

- SECURITY.md (Provide instructions for how to report a security vulnerability)
- Secrets scanning (which we just saw)
- Dependency analysis (Dependabot)
- CodeQL(SAST)via Actions
- GitHub SBOM

Dependency Vulnerabilities



Detect and fix outdated and insecure libraries before they enter your build.

 Dependency vulnerabilities

GitHub supports detection of vulnerable dependencies through **Dependabot**.

This tool allows us to check files such as package.json and pom.xml to understand if our application is including vulnerable components

Features include:

- 💣 Detect vulnerable and out of date packages
- 💣 Create automatic pull requests to remediate issues
- 💣 List vulnerabilities associated with the packages

Let’s take a look at an example of it in action.

Demo – Dependencies

```
Starting dependency scan...

Scheduled modules:
  express  4.17.1
  mongoose  5.9.10          VULNERABLE
  lodash  4.17.19           VULNERABLE
  debug  2.6.8              VULNERABLE
  async  3.2.0
  chalk 2.4.2
Dependency scan completed
```

- Start
- Review
- Remediation

Watch how dependencies are identified and addressed in real time.

☠ Static Analysis ☠

```
def run_query(query):
  sql = "SELECT * FROM table " +
  query
  result = connection.execute(sql)
  if (enception occr
     print("Error:." e)
  return result
```

Parsing

Control flow

Identify code flaws before they reach production

MODUS CREATE

🏴‍☠️ Static Analysis (SAST)

GitHub comes integrated with CodeQL

This supports static analysis security testing of a number of programming languages via Actions.

CodeQL is free to use for public repositories.

You can learn more here:

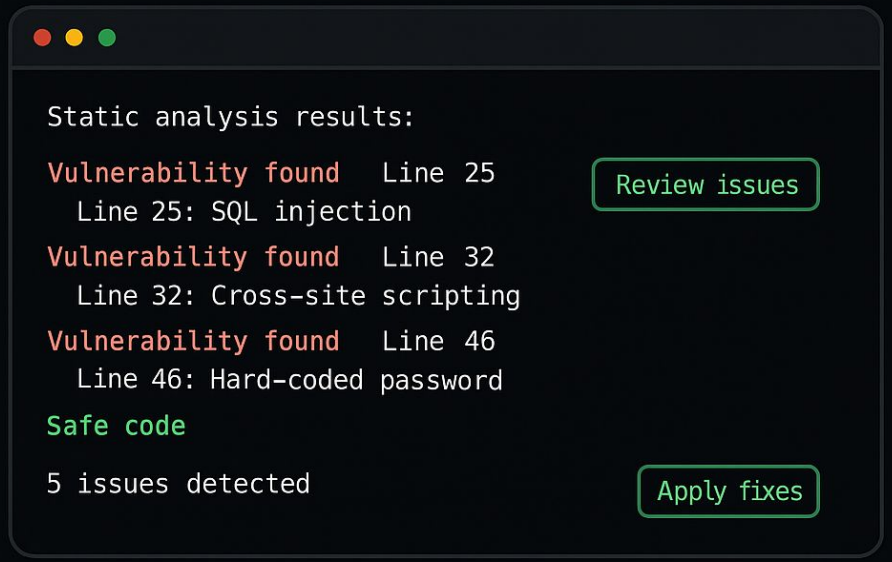
https://docs.github.com/en/code-security/code-scanning/introduction-to-code-scanning/about-code-scanning

Features:

- 💣 Multiple language support (Java, JS, Ruby etc.)
- 💣 Highlights security vulnerabilities
- 💣 Allows for custom queries/query packs
- 💣 Execute as a gating mechanism in your CI/CD pipeline

Demo time!

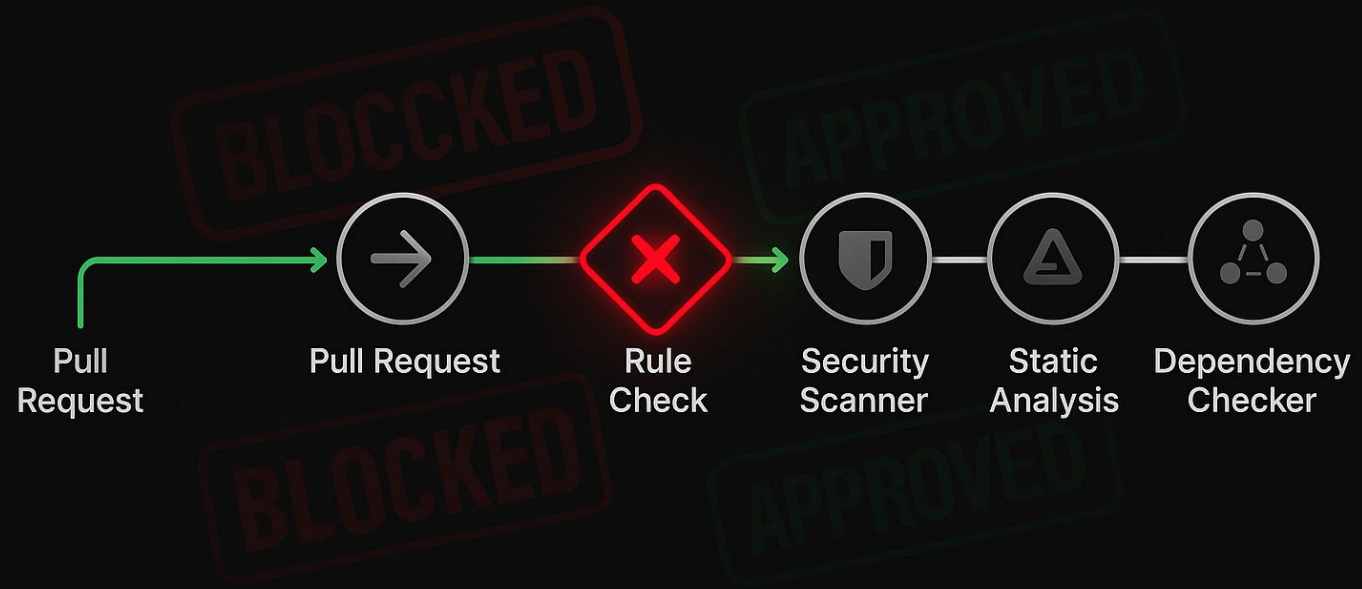
Demo – SAST



- Start scan
- Review issues
- Apply fixes

Watch how static code analysis finds and guides you through fixing security flaws in real time.

💀 Rule Sets 💀



Prevent merges until all security checks pass.

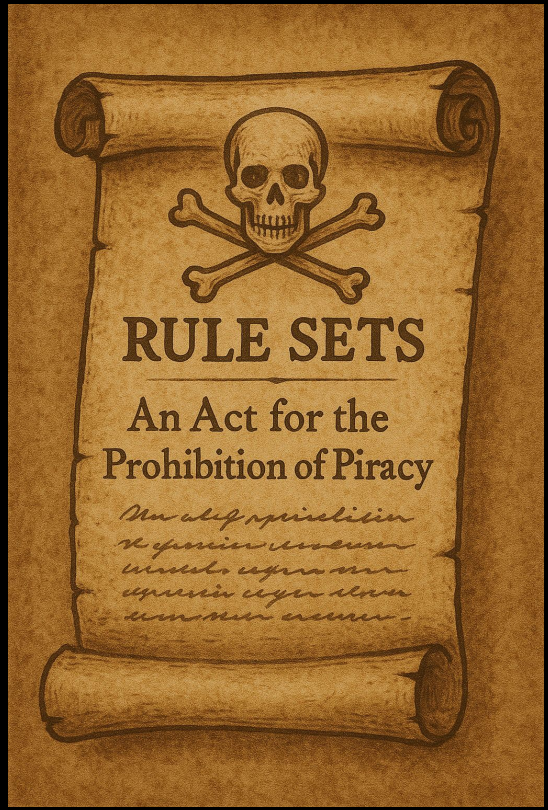
Rule Sets

Rule sets can be used to prevent code being merged until it passess all the relevant tests.

It also allows us to lock down who can commit to the branch.

Features include:

- Block merges until security checks are passed
- Lockdown branch ownership via **CODEOWNERS** files
- Require multiple individuals to approved a pull request before it can be merged



Demo - Rule Sets

MODUS CREATE

MODUS CREATE

 SBOMs

SBOM stands for Software Bill of Materials.

It’s a great way of getting a snapshot of all the components in your application.

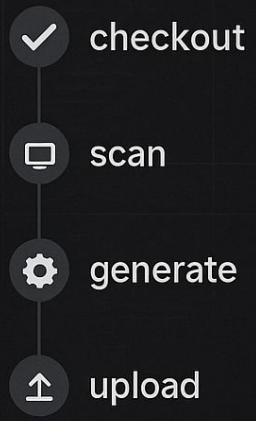
GitHub supports SBOMs out of the box for repositories stored there.

Features of SBOM include:

- CycloneDX format
  https://cyclonedx.org/
- SPDX format
  https://spdx.dev/
- VEX
  https://www.cisa.gov/sites/default/files/2023-01/VEX_Use_Cases_Aprill2022.pdf

Demo — SBOMs

```yaml
jobs:
   sboms:
    - uses: actions/checkout@v4
    - uses: aquasecurity/trivy-action@.0.16.0
      id: trivy
      with: scan-sbom-command sbom
            scan-vex-command vex
    - uses: actions/upload-artifact@v4
      with: name sboms
            path: sboms/**/*
```



Watch your pipeline generate SBOM and VEX files in real time.

Conclusion

MODUS CREATE

MODUS CREATE

Review of the workshop

In this workshop we covered a number of areas related to Shifting Left.

- Secrets scanning
- SAST
- Dependency scanning
- Rule sets
- SBOMS

Next Steps

After this workshop you can take the repository you forked and use it as a starting to point for your own projects.

Some ideas for other areas to explore are listed on the right.

- Try integrating with Infrastructure-as-Code. Check out Checkov: https://www.checkov.io/
- Experiment with other third party tooling for SAST. For example, the open source Horusec: https://github.com/ZupIT/horusec
- Look into container scanning: https://github.com/quay/clair

MODUS CREATE

Thanks for attending!

Thank you for attending the workshop today. A big thank you to the BSidesSF crew as well.

Remember to check out:

https://moduscreate.com/careers/

For our latest job posts!

Questions?