Security Training

Andrew Scoppa

Agenda

Setting the scene

Security features and how they fit into a secure development workflow.

Configuring access

Creating teams and applying appropriate permissions.

Reviewing and analyze alerts

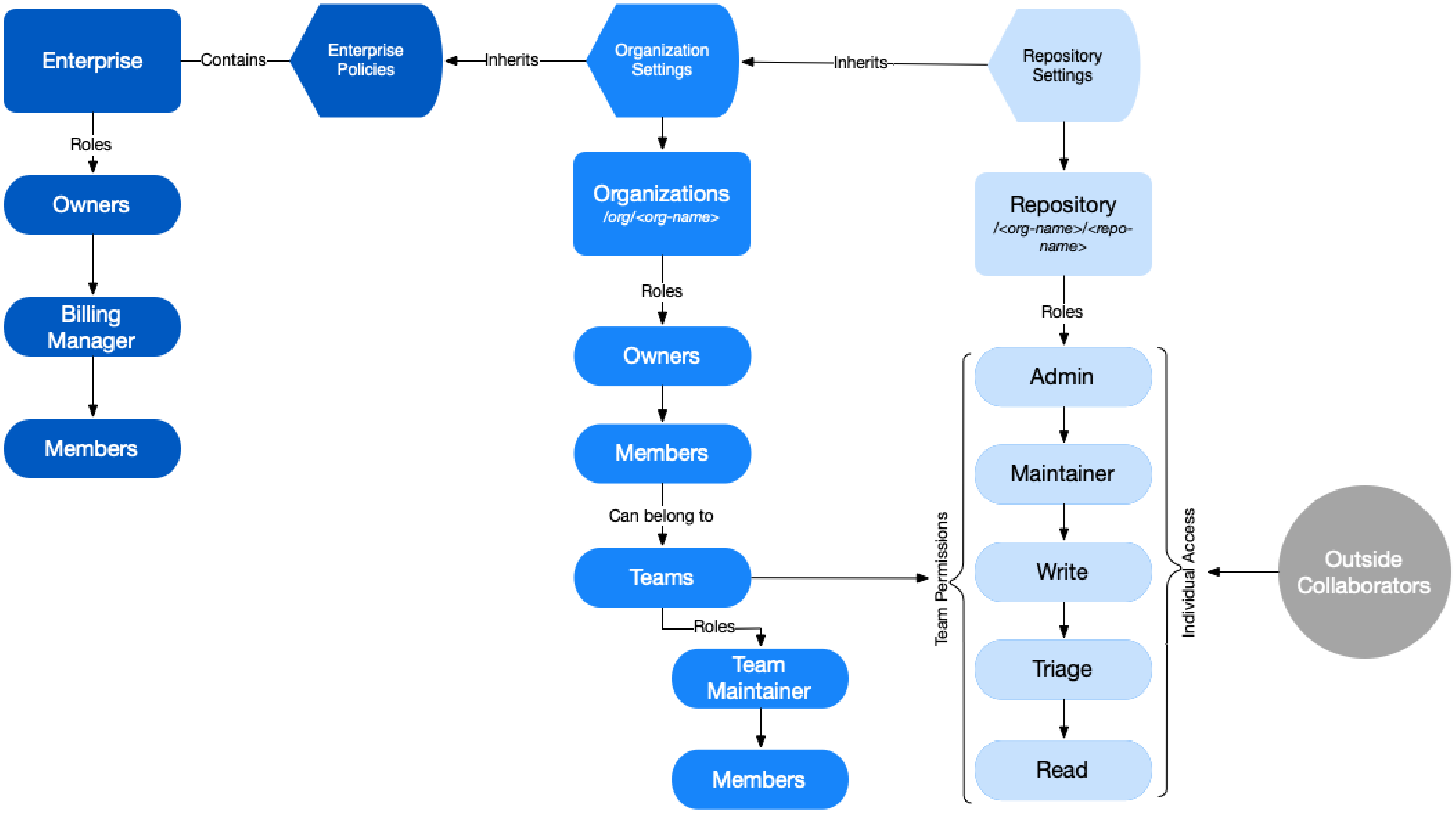
Use the integrated reporting facilities to identify common issues and understand risk factors.

Securing your supply chain

Understanding vulnerabilities in dependencies and patching them.

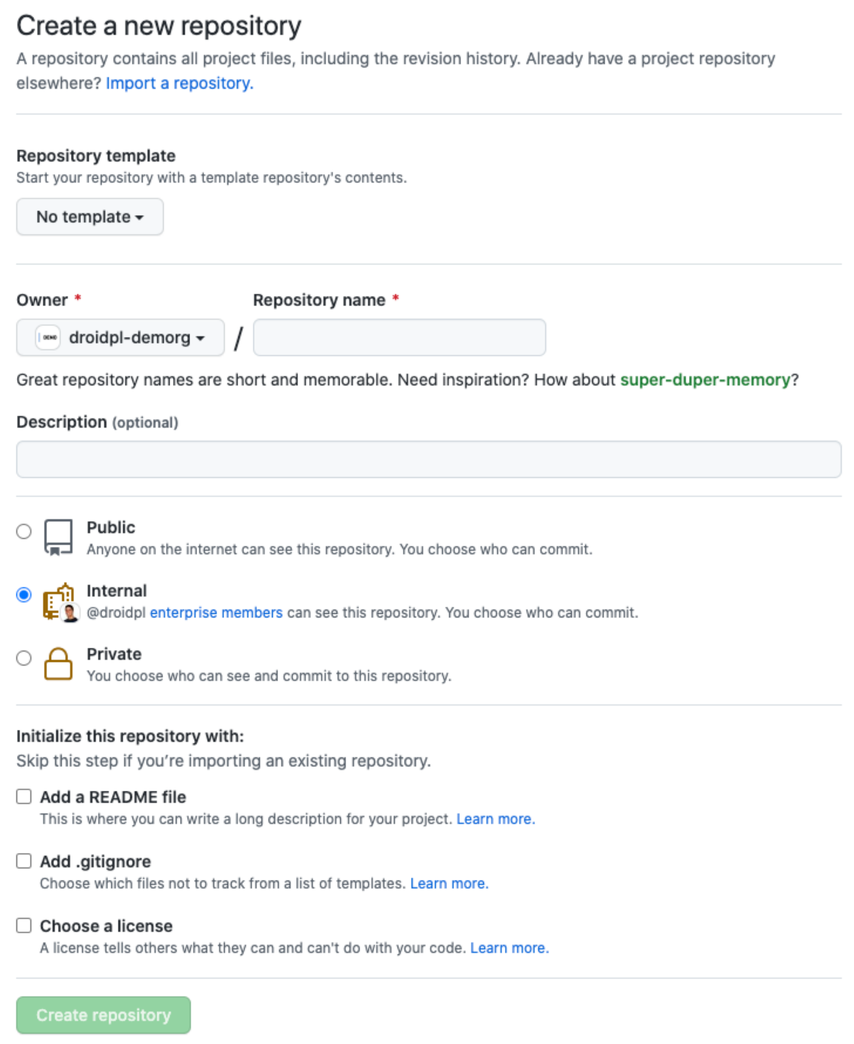
Configuring access

Flow of permissions



Repository visibility

- **Public** - Anyone on the internet can access (GHEC only)
- **Internal** - Organization members in the enterprise can access
- **Private** - Only people with explicit access



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? Import a repository.

Repository template
Start your repository with a template repository's contents.

No template ▾

Owner * **Repository name** *

droidpl-demorg ▾ /

Great repository names are short and memorable. Need inspiration? How about super-duper-memory?

Description (optional)

○ **Public**
Anyone on the internet can see this repository. You choose who can commit.

● **Internal**
@droidpl enterprise members can see this repository. You choose who can commit.

○ **Private**
You choose who can see and commit to this repository.

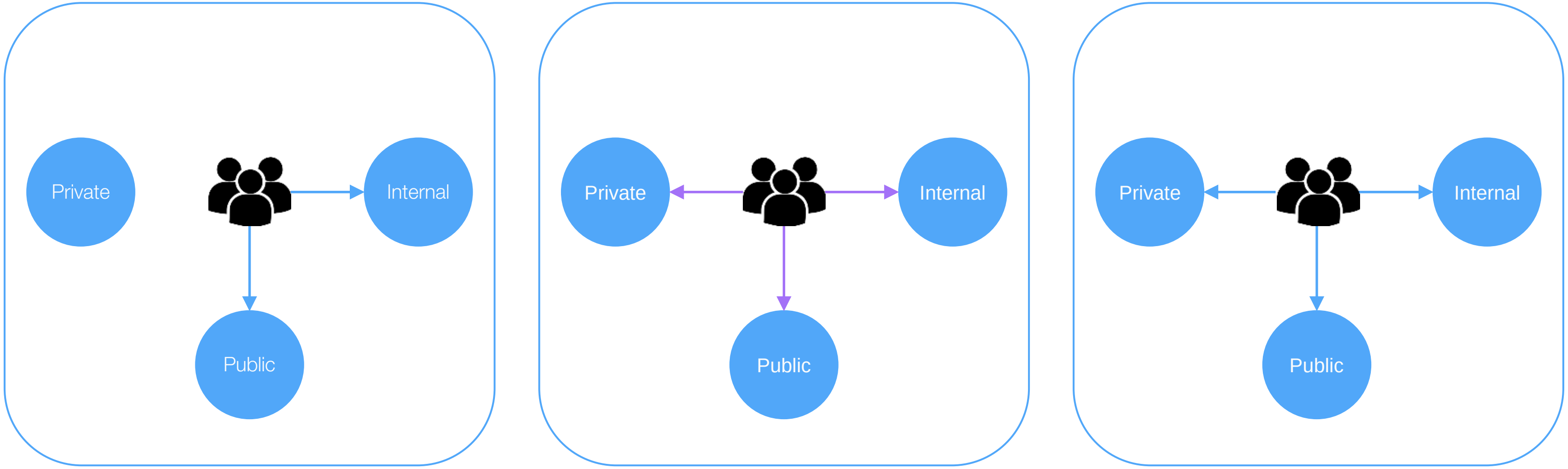
Initialize this repository with:
Skip this step if you're importing an existing repository.

□ **Add a README file**
This is where you can write a long description for your project. Learn more.

□ **Add .gitignore**
Choose which files not to track from a list of templates. Learn more.

□ **Choose a license**
A license tells others what they can and can't do with your code. Learn more.

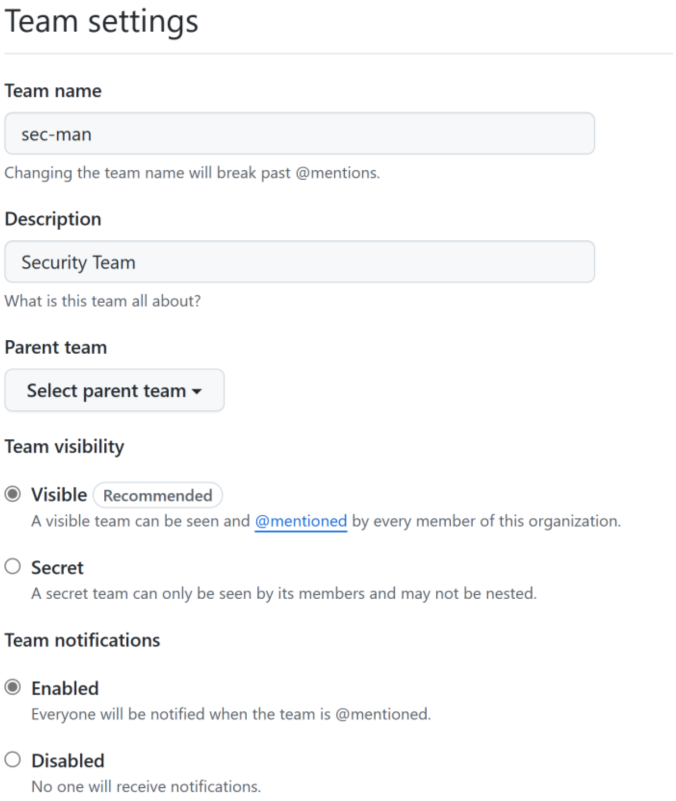
Repository base permissions



Roles

Role	Description
`Read`	Read-only access to Code and Actions. Can submit and comment on issues, pull requests, and discussions
`Triage`	Read-only permissions with the additional ability to manage issues, pull requests, discussions, assignments, and labels
`Write`	Gives write access to all parts of a repository project with the exception of the repository settings
`Maintain`	Ability to modify some settings of a repository including topics, enabling repository features, configuring merges and GitHub pages, pushing to protected branches
`Admin`	Has full administrative access to all features, settings and configurations of the repository project

GitHub Teams



Team settings

Team name

sec-man

Changing the team name will break past @mentions.

Description

Security Team

What is this team all about?

Parent team

Team visibility

● **Visible** Recommended
A visible team can be seen and @mentioned by every member of this organization.

○ **Secret**
A secret team can only be seen by its members and may not be nested.

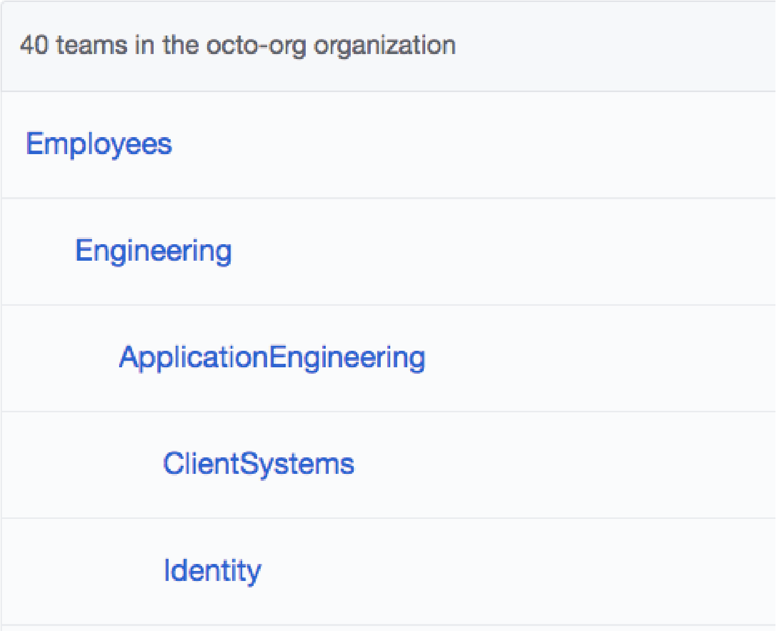
Team notifications

● **Enabled**
Everyone will be notified when the team is @mentioned.

○ **Disabled**
No one will receive notifications.

Managing GitHub Teams

- Nested teams allow you to reflect your company's hierarchy within your org
- Parents team can have more than one child
    - Child teams inherit parent's permissions
    - Children receive parent's notifications
    - Users in a child team belong also to the parent team



40 teams in the octo-org organization

Employees

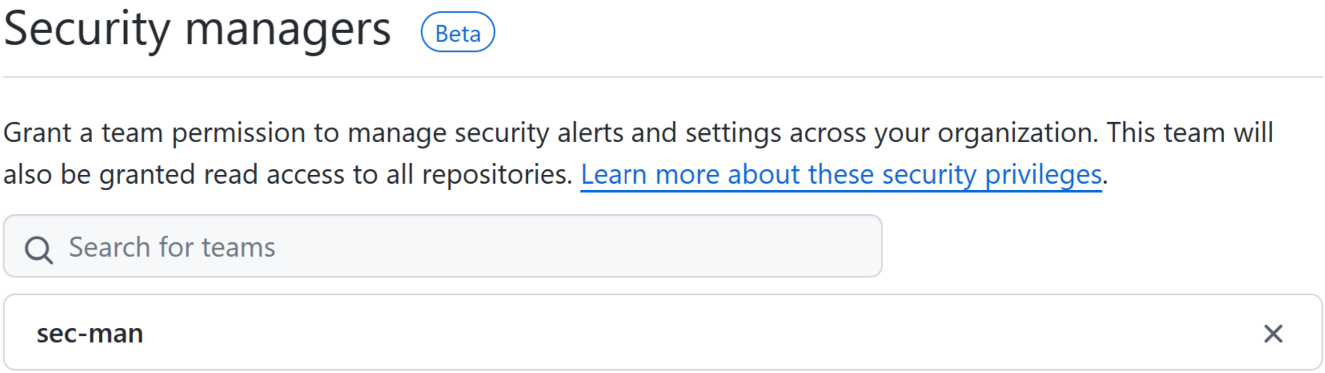
Engineering

ApplicationEngineering

ClientSystems

Identity

Managing Security Managers in your Organization



Security managers

Grant a team permission to manage security alerts and settings across your organization. This team will also be granted read access to all repositories. Learn more about these security privileges.

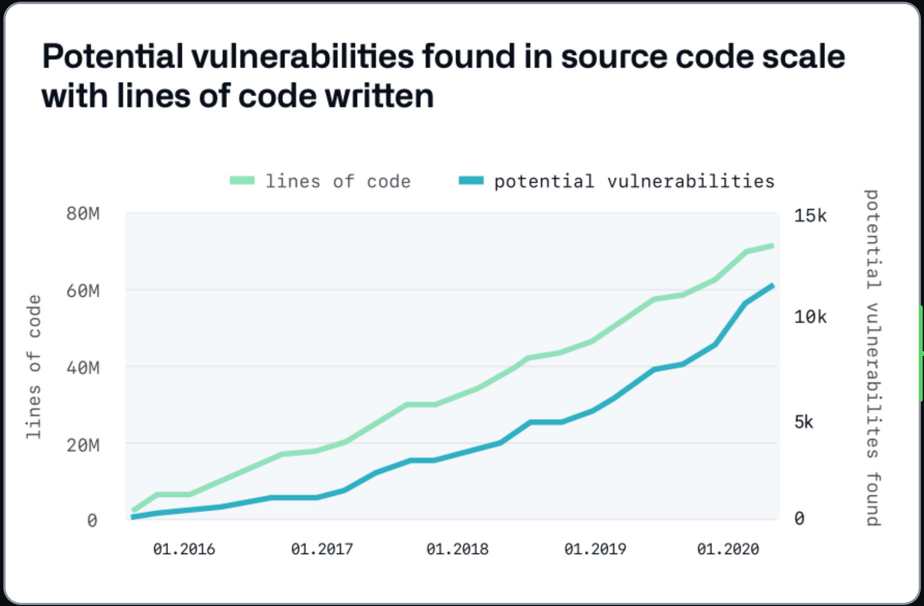
sec-man

- Security manager is an organization-level role that organization owners can assign to any team in an organization.

- It gives every member of the team permissions to view security alerts and manage settings for code security across your organization, as well as read permissions for all repositories in the organization.

Application Security

The state of AppSec



Potential vulnerabilities found in source code scale with lines of code written

Despite billions of dollars of investment…

85% of applications still contain a security issue

Code written in 2020 is just as likely to introduce a security issue as code written in 2016

Flaws in applications are consistently the #1 attack vector for breaches

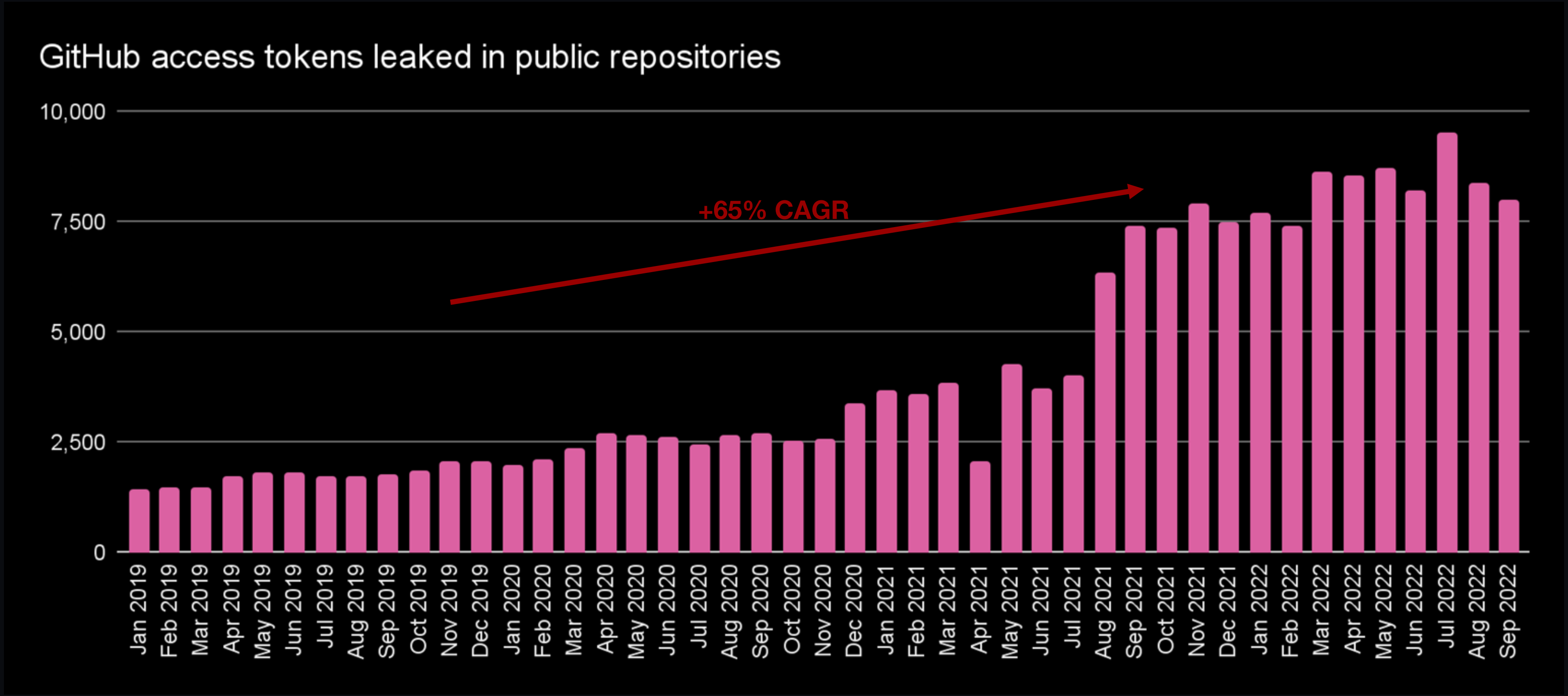
Source: Verizon Data Breach Investigations reports 2016, 2017, 2018, 2019 and 2020.

The state of AppSec

Is falling further behind the current state of Development

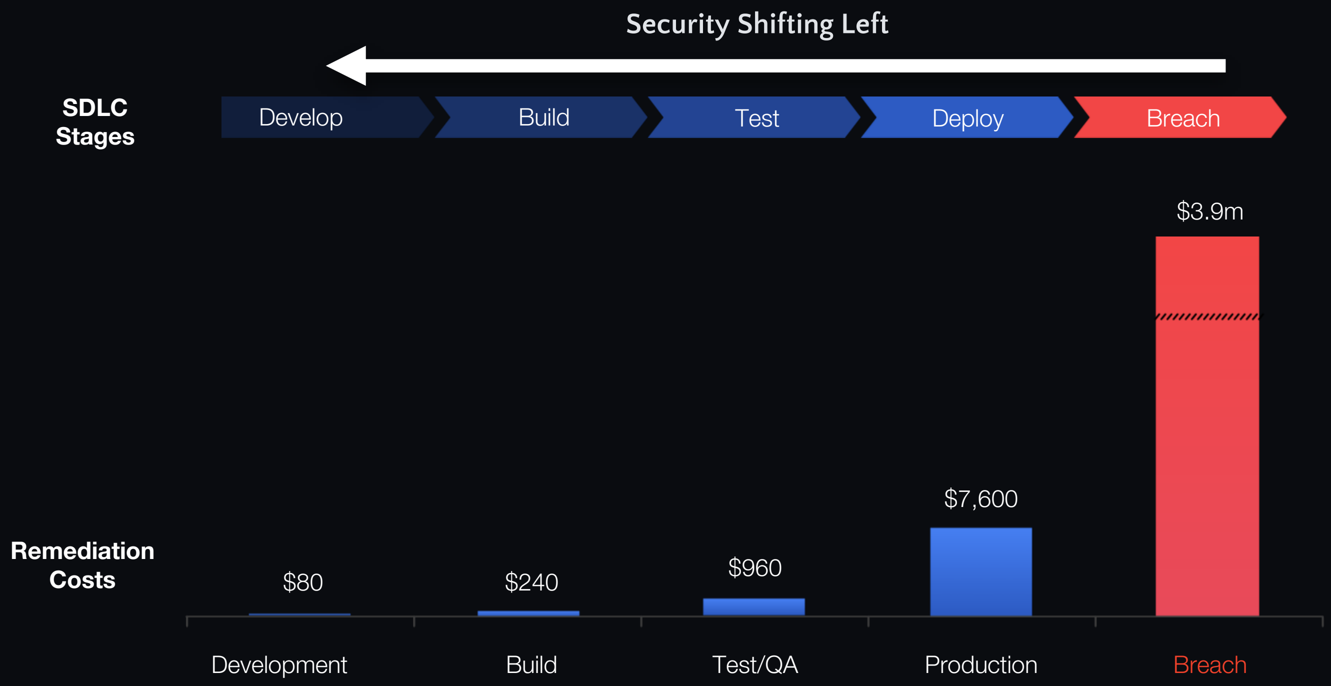
- 1:100 Security team members to developers
- Lack of knowledge voted the main AppSec challenge
- Remediation trends are stagnant

We’re seeing more credential leaks than ever


GitHub access tokens leaked in public repositories

Source: GitHub data

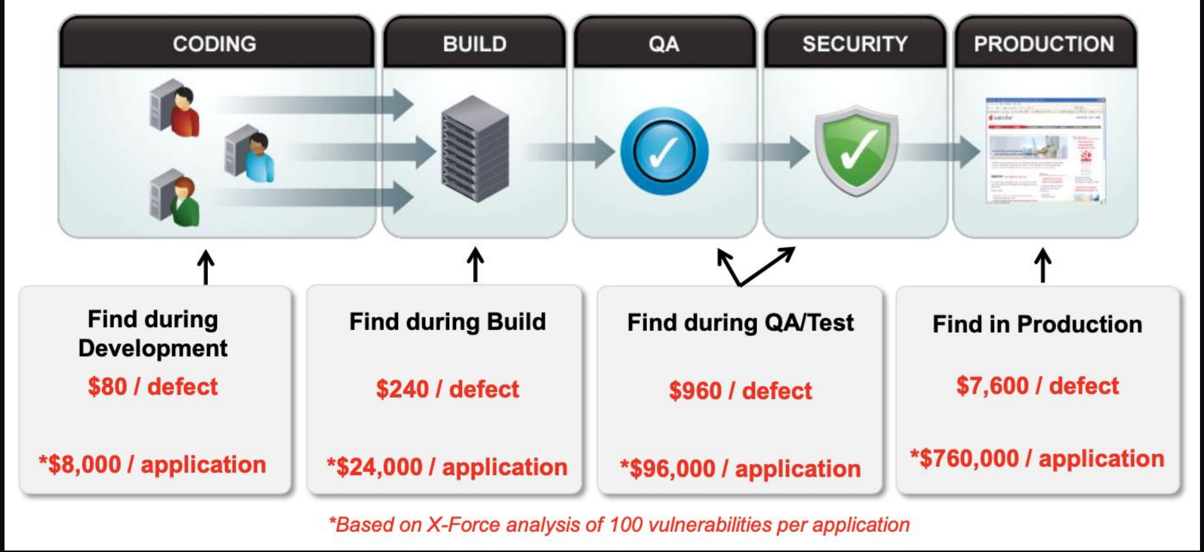
Everyone wants to shift security left…



Source: Ponemon Institute Cost of a Data Breach 2020

… but the industry has been trying to shift left for at least a decade

IBM Security Systems report, 2012

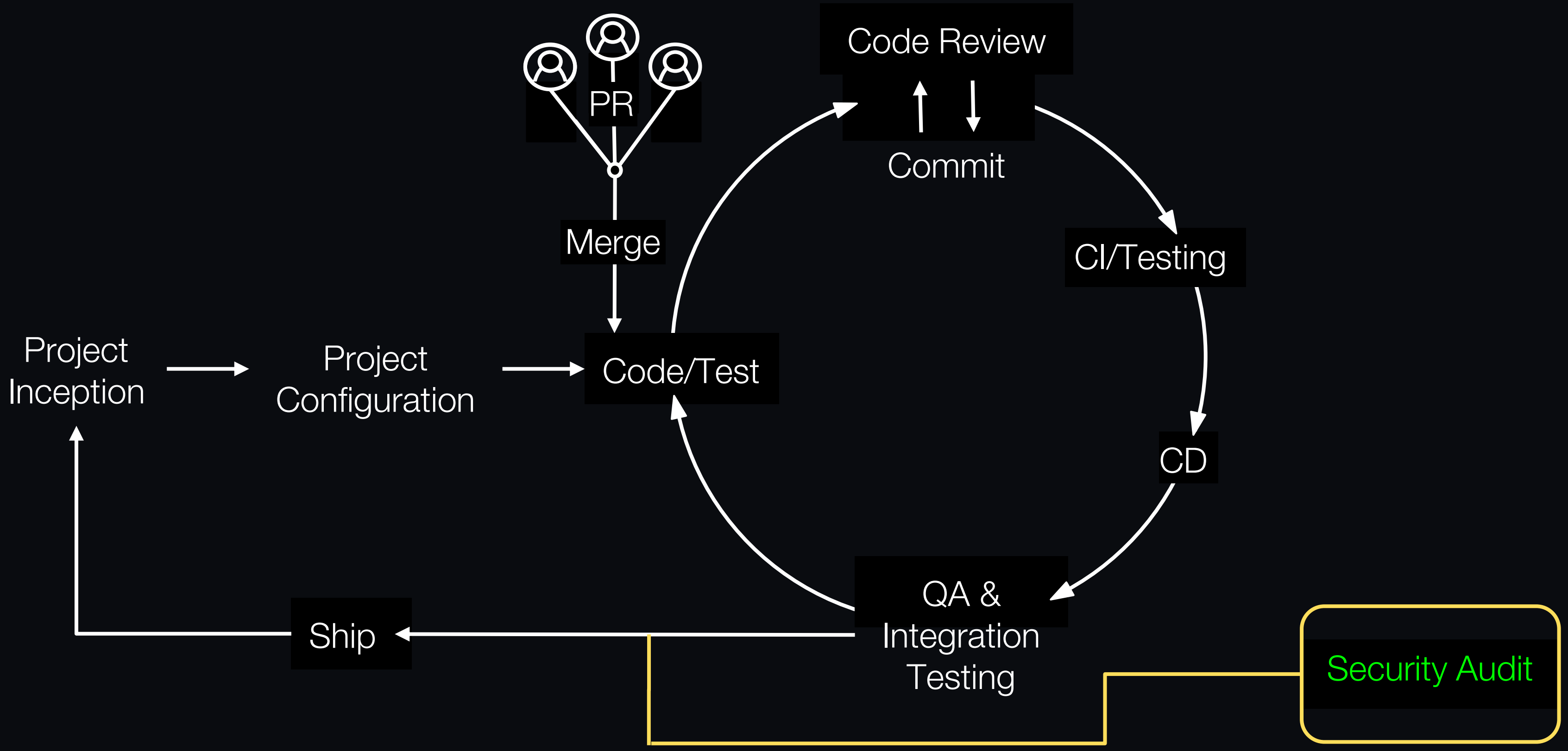


Find during Development	Find during Build	Find during QA/Test	Find in Production
$80 / defect	$240 / defect	$960 / defect	$7,600 / defect
*$8,000 / application	*$24,000 / application	*$96,000 / application	*$760,000 / application

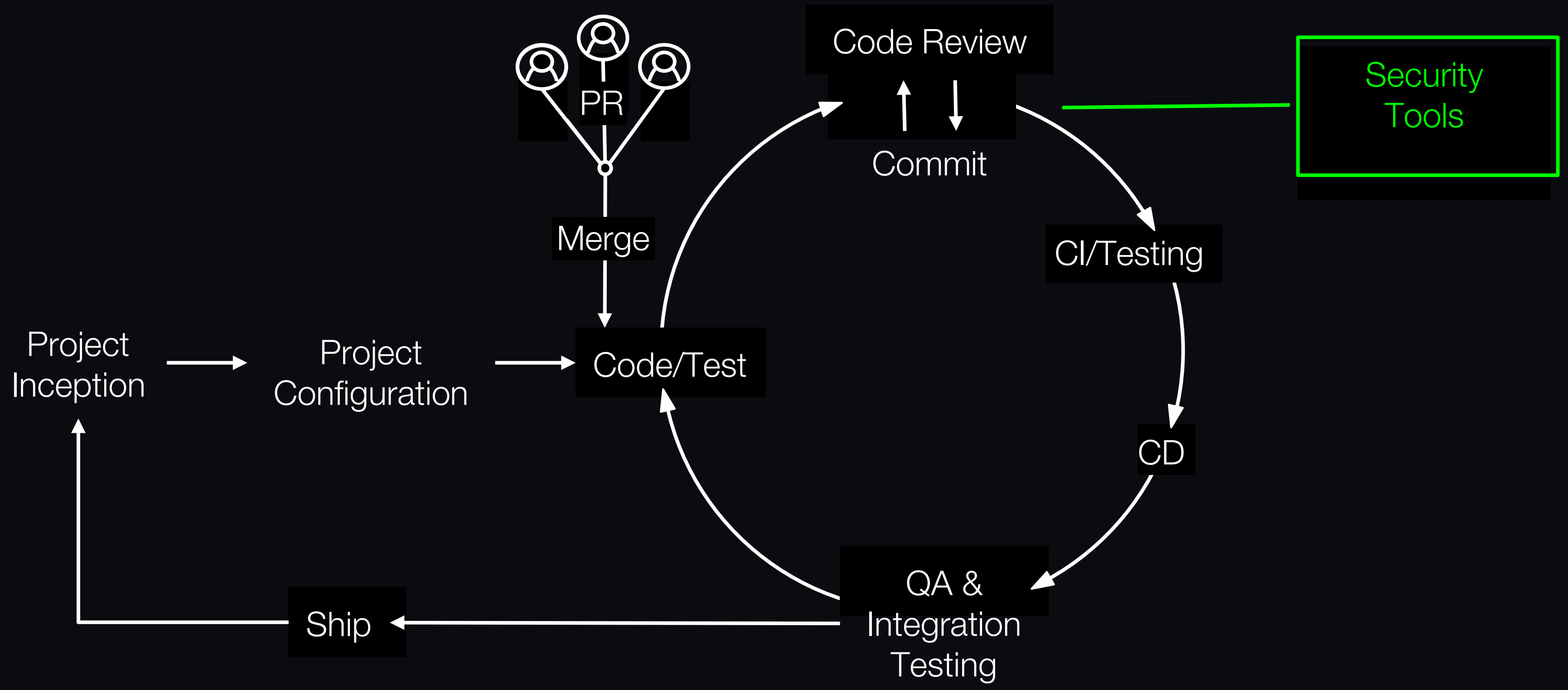
*Based on X-Force analysis of 100 vulnerabilities per application

GitHub believes that making this shift requires a developer-first approach to all our security products

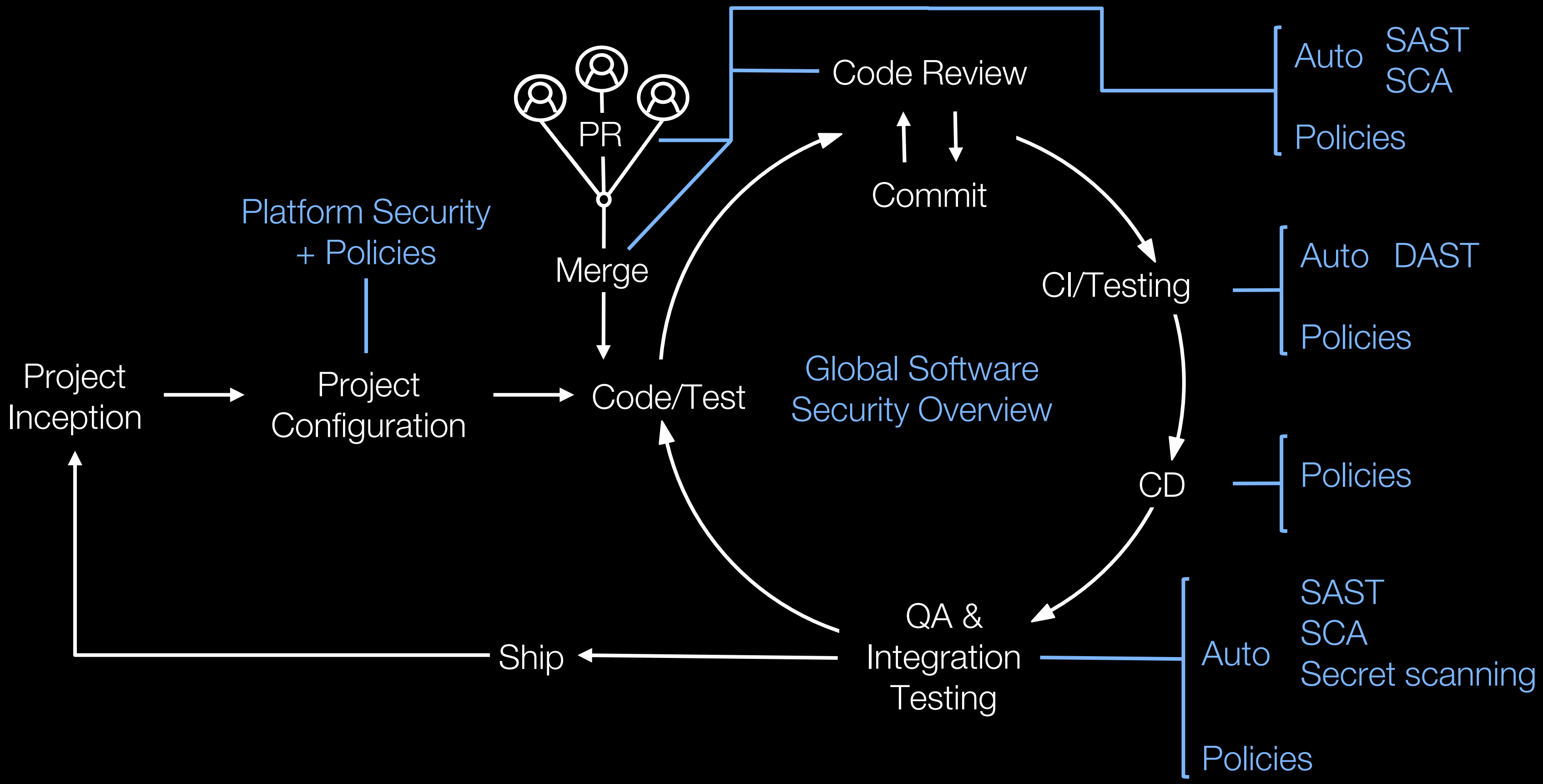
Basic Application Security scenario



Improved Application Security scenario



Application Security - Targeted state



Developer first?

We see three key aspects to being a “developer first” tool:

Integrate *directly* into the developer workflow.

Make setup and deployment fast and easy.

Produce high quality results with low numbers of false positives.

GitHub Advanced Security: Current capabilities

Dependency graph
View your dependencies

Advisory database
Canonical database of dependency vulnerabilities

Security alerts and updates
Notifications for vulnerabilities in your dependencies, and pull requests to fix them

Dependency review
Identify new dependencies and vulnerabilities in a PR

Secret scanning
Find API tokens or other secrets exposed anywhere in your git history.

Code scanning
Static analysis of every git push, integrated into the developer workflow and powered by CodeQL

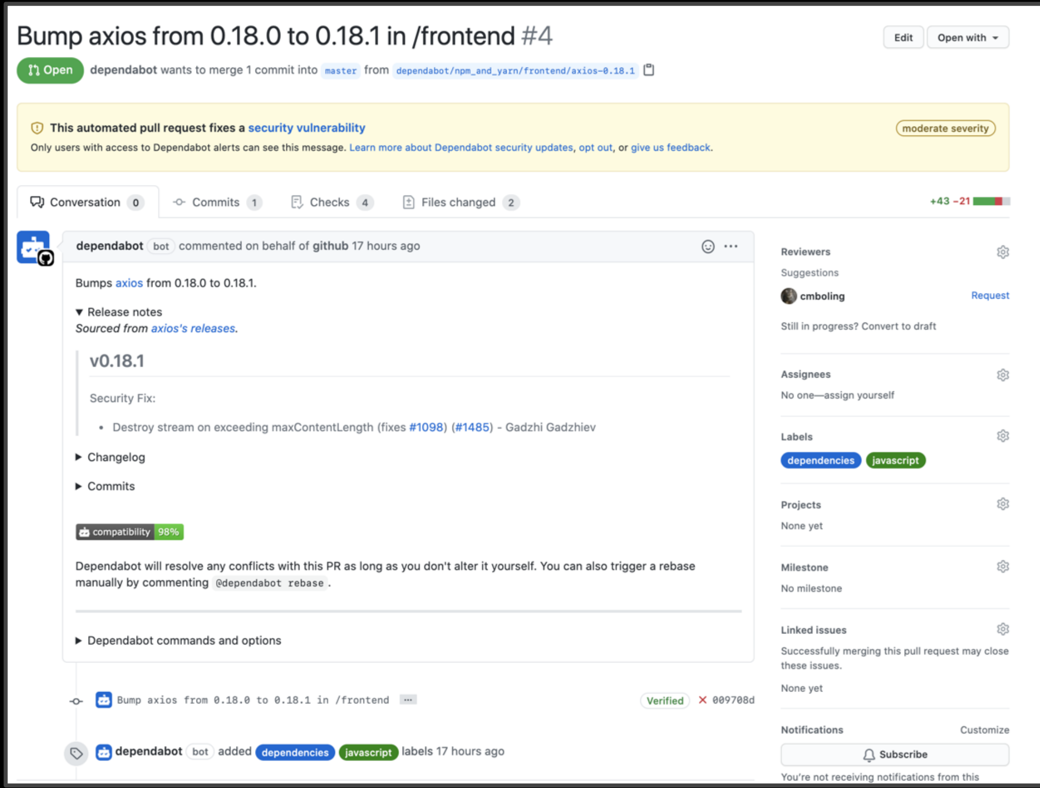
Branch protection
Enforce requirement for pushing to a branch or merging PRs

Commit signing
Enforce requirement that all commits are signed

Security overview
View security results of all kinds acros your organization

Dependabot

- Developers (and others!) notified by an alert when new vulnerable dependencies are detected.
- Automatically open pull requests to fix dependency vulnerabilities.
- Supports dependency review within PRs to prevent adding known vulnerable dependencies.



Bump axios from 0.18.0 to 0.18.1 in /frontend #4

dependabot wants to merge 1 commit into `master` from `dependabot/npm_and_yarn/frontend/axios-0.18.1`

This automated pull request fixes a security vulnerability

moderate severity

Only users with access to Dependabot alerts can see this message. Learn more about Dependabot security updates, opt out, or give us feedback.

dependabot bot commented on behalf of github 17 hours ago

Bumps axios from 0.18.0 to 0.18.1.

▼ Release notes

Sourced from axios's releases.

> v0.18.1
>
> Security Fix:
>
> - Destroy stream on exceeding maxContentLength (fixes #1098) (#1485) - Gadzhi Gadzhiev

▶ Changelog

▶ Commits

compatibility 98%

Dependabot will resolve any conflicts with this PR as long as you don't alter it yourself. You can also trigger a rebase manually by commenting `@dependabot rebase`.

▶ Dependabot commands and options

`Bump axios from 0.18.0 to 0.18.1 in /frontend` … Verified ✕ 009708d

dependabot bot added dependencies javascript labels 17 hours ago

Reviewers

Suggestions

cmboling Request

Still in progress? Convert to draft

Assignees

No one—assign yourself

Labels

dependencies javascript

Projects

None yet

Milestone

No milestone

Linked issues

Successfully merging this pull request may close these issues.

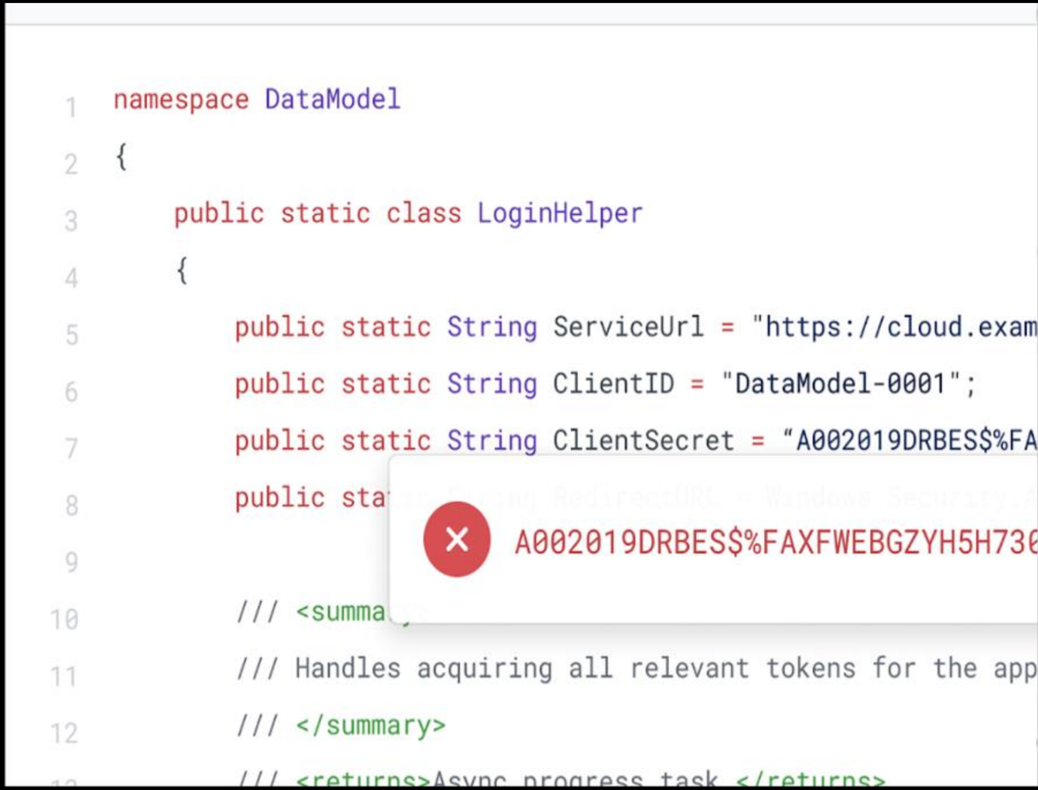
None yet

Notifications Customize

You're not receiving notifications from this

Secret scanning

- Identify secrets across your entire git history with high accuracy.
- Push protection - prevent secrets from being pushed to GitHub.
- Developers (and others!) notified by an alert if secrets are pushed.
- Automated revocation for public repositories, private repositories include a review workflow.



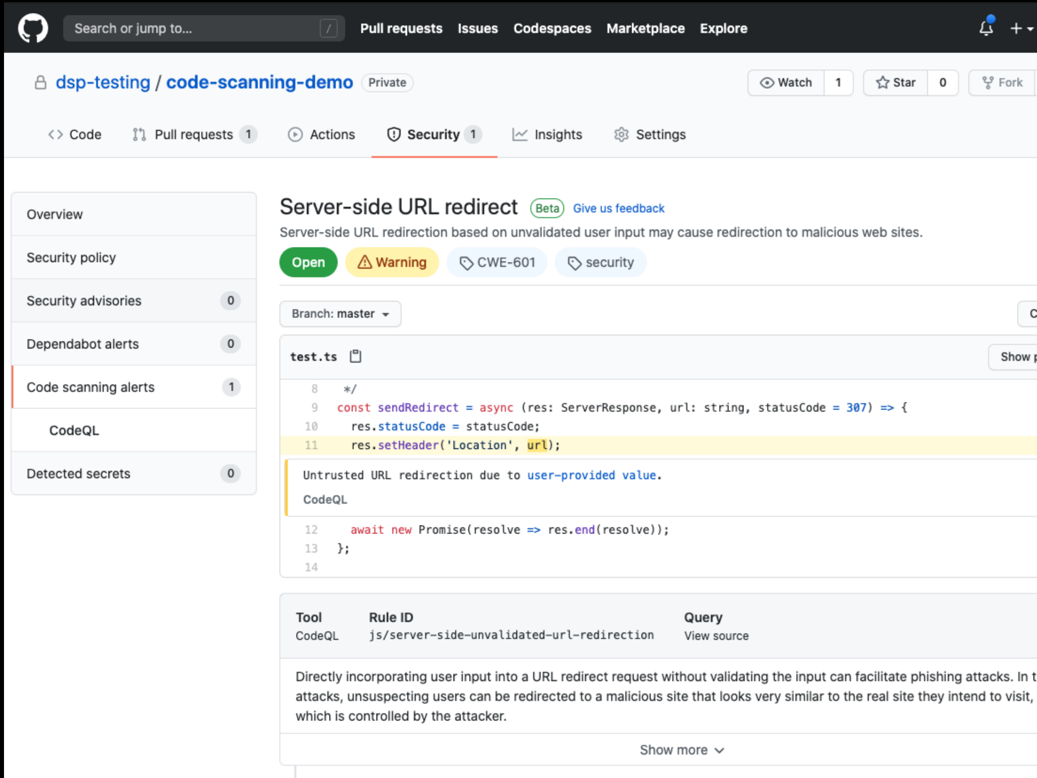
```
namespace DataModel
{
    public static class LoginHelper
    {
        public static String ServiceUrl = "https://cloud.exam
        public static String ClientID = "DataModel-0001";
        public static String ClientSecret = “A002019DRBES$%FA
        public sta

        /// <summa
        /// Handles acquiring all relevant tokens for the app
        /// </summary>
        /// <returns>Async progress task </returns>
```

✕ A002019DRBES$%FAXFWEBGZYH5H73

Code scanning

- Find vulnerabilities before they are merged into the code base with automated CodeQL scans
- Integrate results directly into the developer workflow
- Run custom queries and the community-powered GitHub query set
- Extensible, with support for other SAST tools



Server-side URL redirect Beta Give us feedback

Server-side URL redirection based on unvalidated user input may cause redirection to malicious web sites.

test.ts

```
  */
const sendRedirect = async (res: ServerResponse, url: string, statusCode = 307) => {
  res.statusCode = statusCode;
  res.setHeader('Location', url);
```

Untrusted URL redirection due to user-provided value.

CodeQL

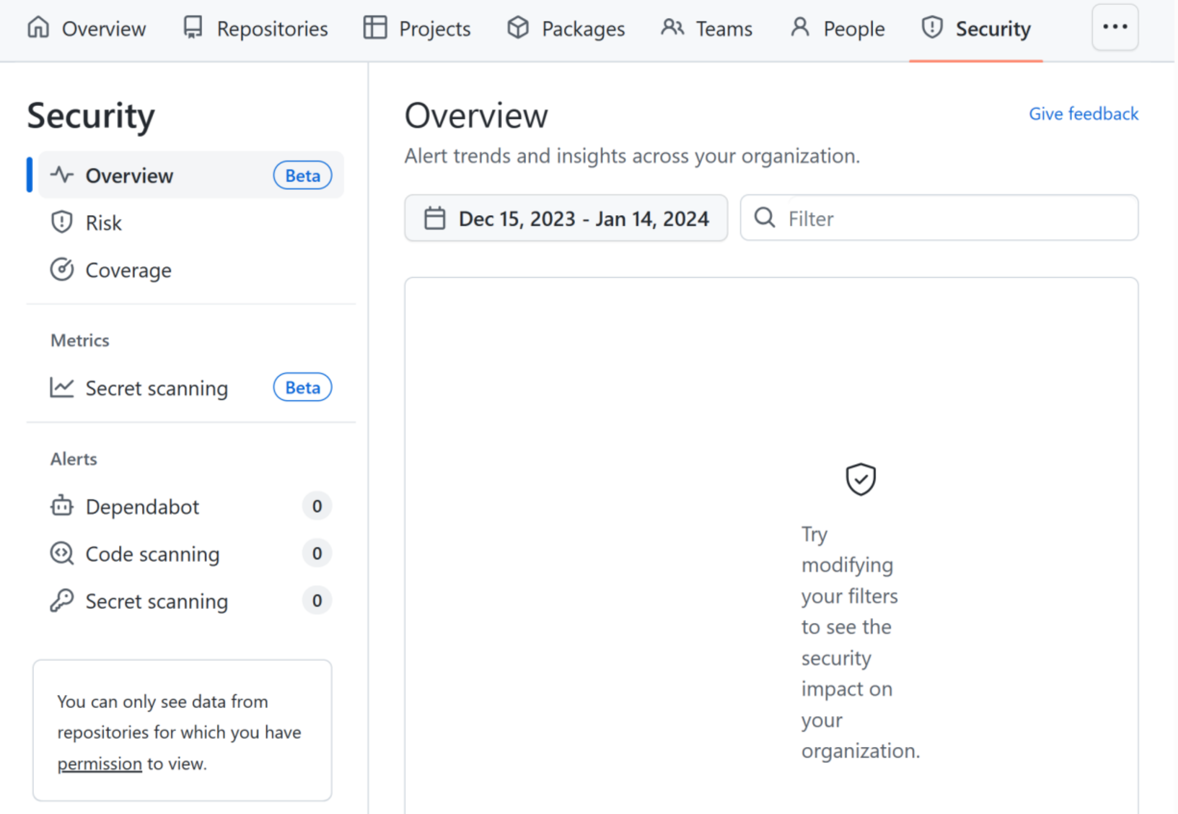
```
  await new Promise(resolve => res.end(resolve));
};
```

Tool	Rule ID	Query
CodeQL	js/server-side-unvalidated-url-redirection	View source

Directly incorporating user input into a URL redirect request without validating the input can facilitate phishing attacks. In t
attacks, unsuspecting users can be redirected to a malicious site that looks very similar to the real site they intend to visit,
which is controlled by the attacker.

Show more

Reviewing Alerts



Security

- Overview Beta
- Risk
- Coverage

Metrics

- Secret scanning Beta

Alerts

Dependabot	0
Code scanning	0
Secret scanning	0

You can only see data from repositories for which you have permission to view.

Overview

Give feedback

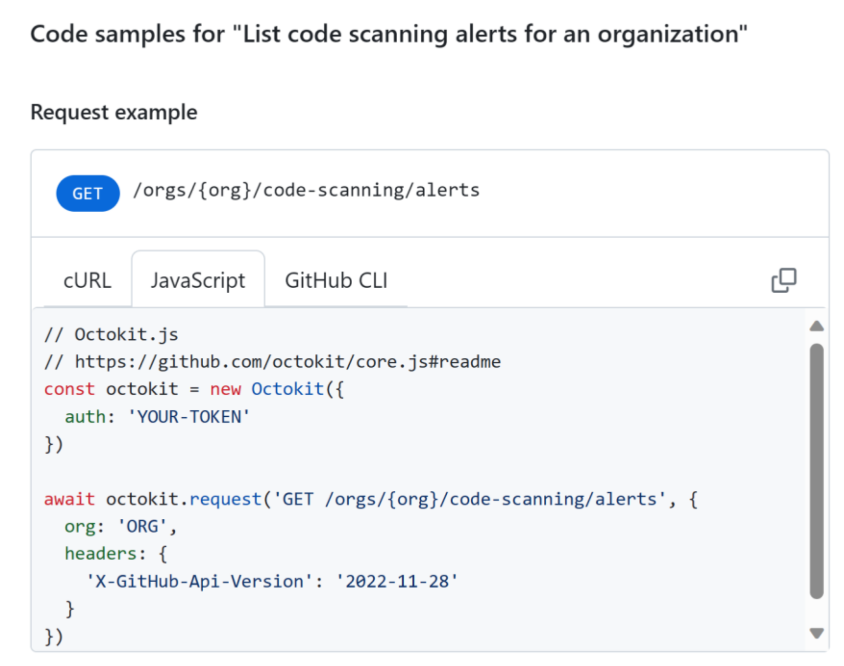
Alert trends and insights across your organization.

Dec 15, 2023 - Jan 14, 2024

Filter

Try modifying your filters to see the security impact on your organization.

Monitoring and responding to alerts



Code samples for "List code scanning alerts for an organization"

Request example

GET /orgs/{org}/code-scanning/alerts

```
// Octokit.js
// https://github.com/octokit/core.js#readme
const octokit = new Octokit({
  auth: 'YOUR-TOKEN'
})

await octokit.request('GET /orgs/{org}/code-scanning/alerts', {
  org: 'ORG',
  headers: {
    'X-GitHub-Api-Version': '2022-11-28'
  }
})
```

Q&A

Resources and Examples

- Code security documentation - GitHub Docs
  Build security into your GitHub workflow.
- https://docs.github.com/en/rest/code-scanning
  Use the REST API to retrieve and update code scanning alerts from a repository.
- Removing sensitive data from a repository
  Remove unwanted files from a repository's history
- https://github.com/advanced-security/advanced-security-material
  A place for resources to help you understand and use GitHub Advanced Security (GHAS)
- https://github.com/advanced-security/policy-as-code
  Example application which uses the GHAS APIs to create policy engine using GitHub Actions.
- https://github.com/github/ghas-jira-integration
  A project showing how to integrate GitHub Advanced Security with JIRA.