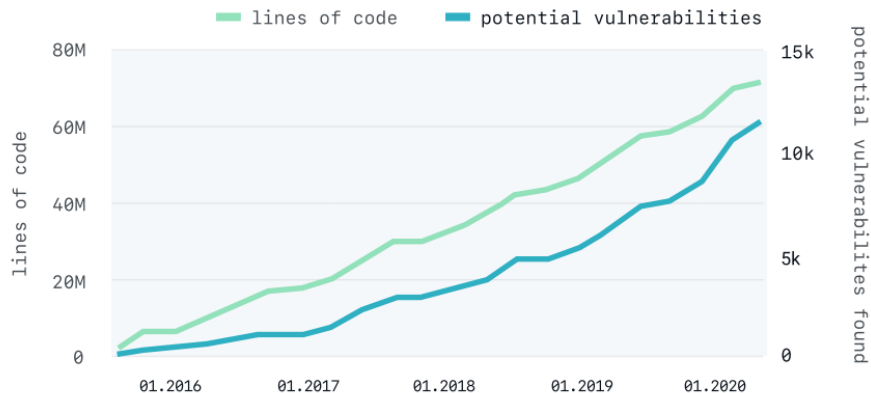




GitHub Advanced Security for Developers

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



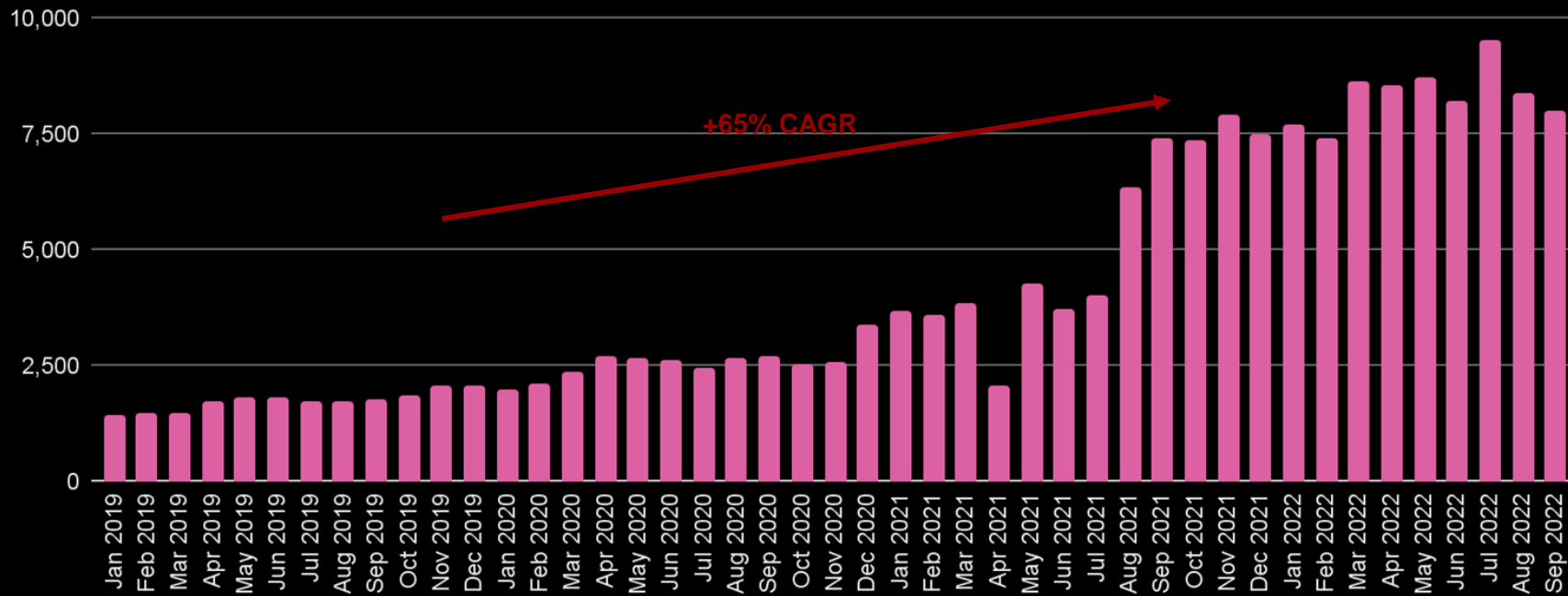
Lack of knowledge voted the main AppSec challenge



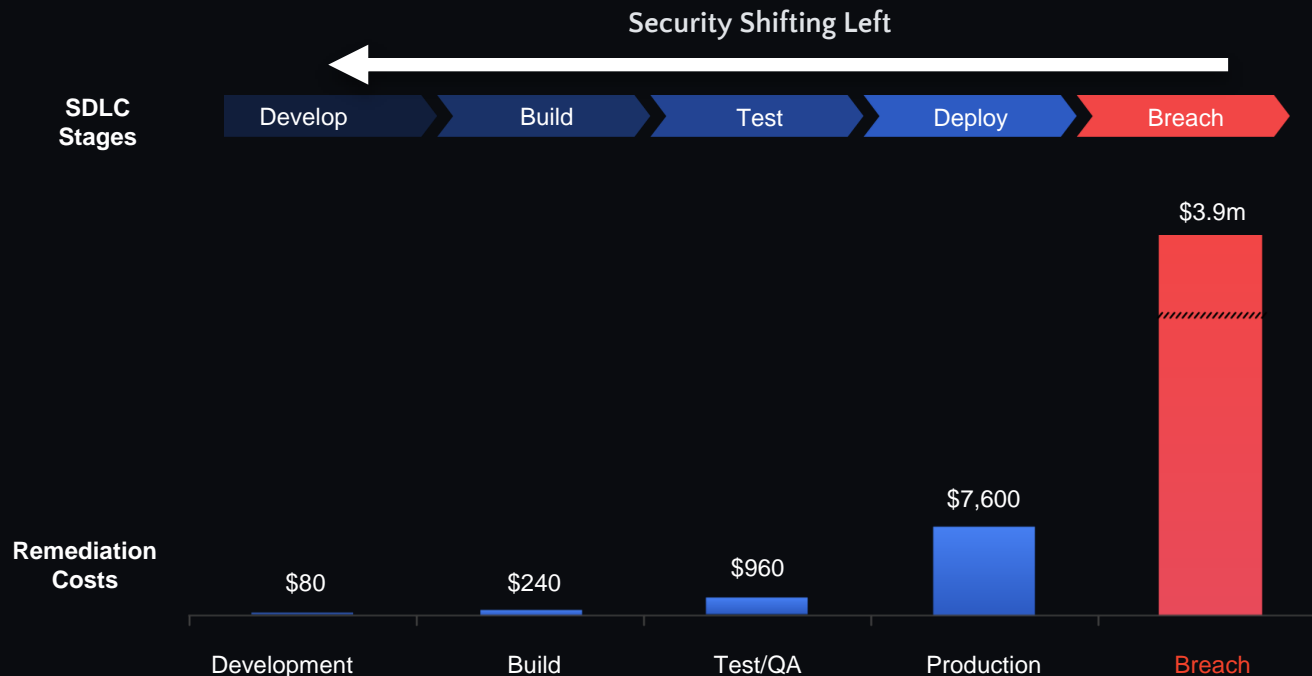
Remediation trends are stagnant

We're seeing more credential leaks than ever

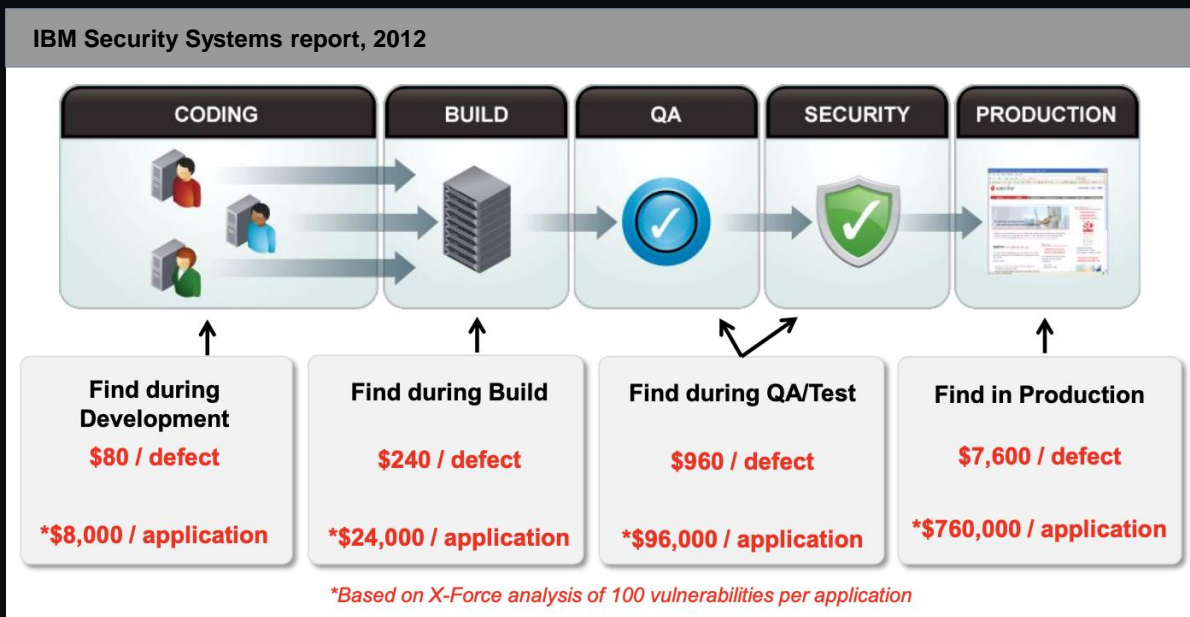
GitHub access tokens leaked in public repositories



Everyone wants to shift security left...



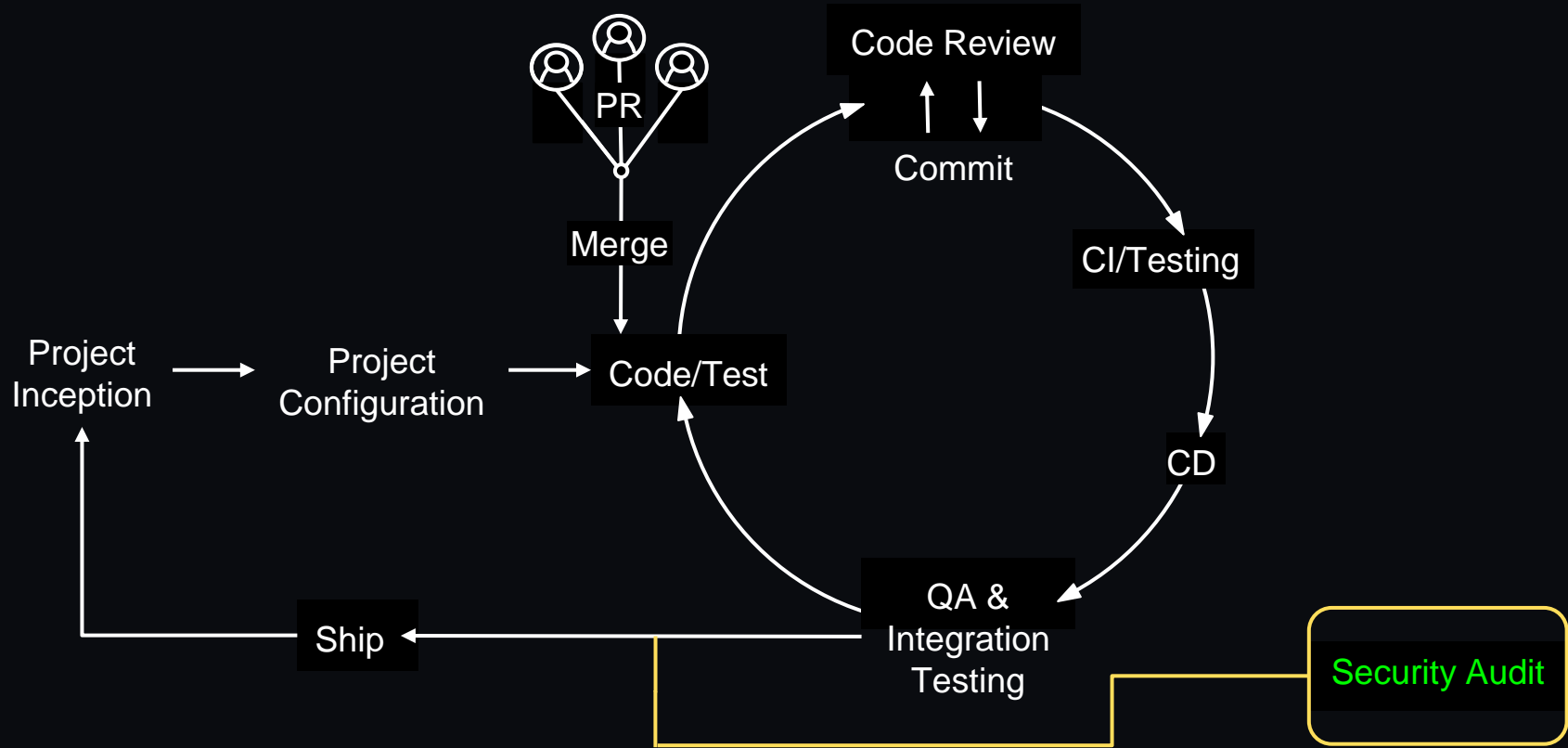
... but the industry has been trying to shift left for at least a decade



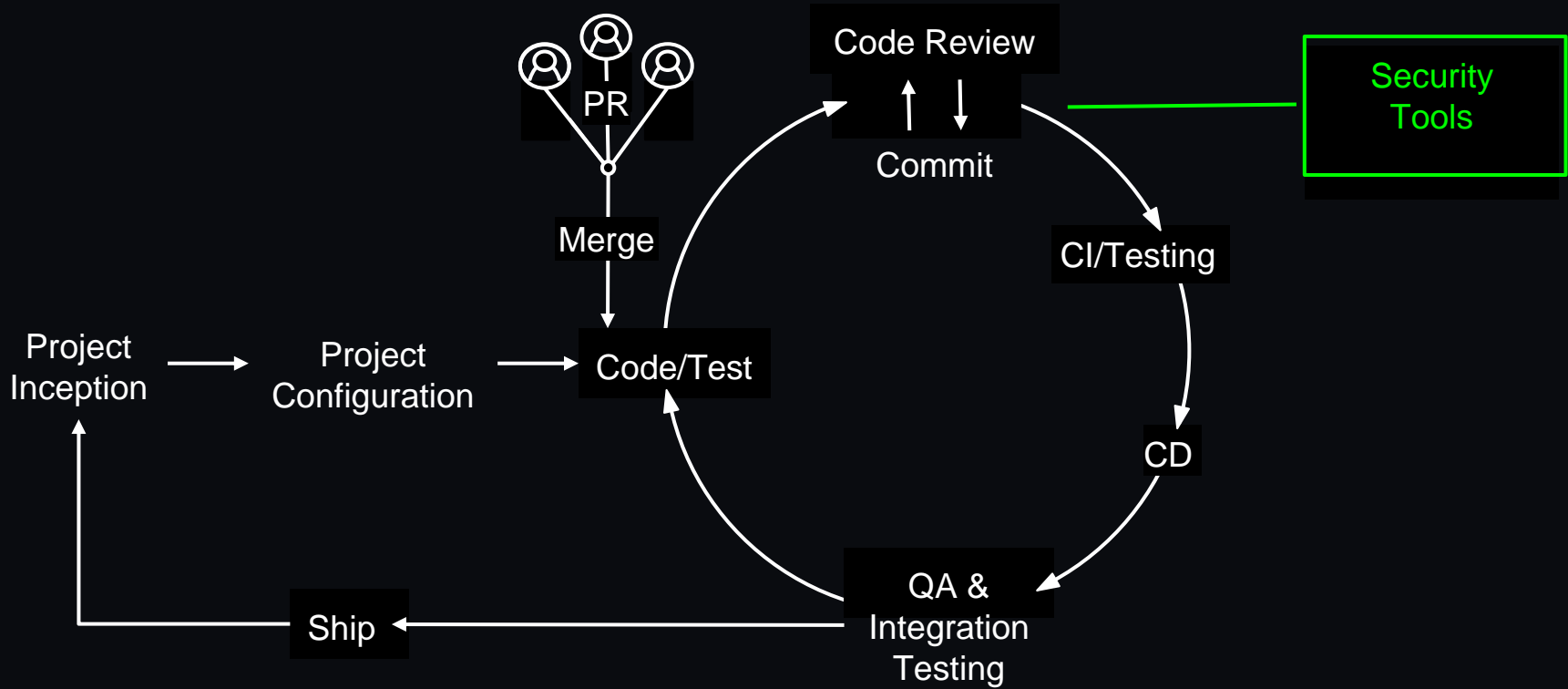
GitHub believes that making this shift requires a developer-first approach to all our security products:

- Integrate *directly* into the developer workflow.
- Make setup and deployment fast and easy.
- Produce high quality results with low numbers of false positives.

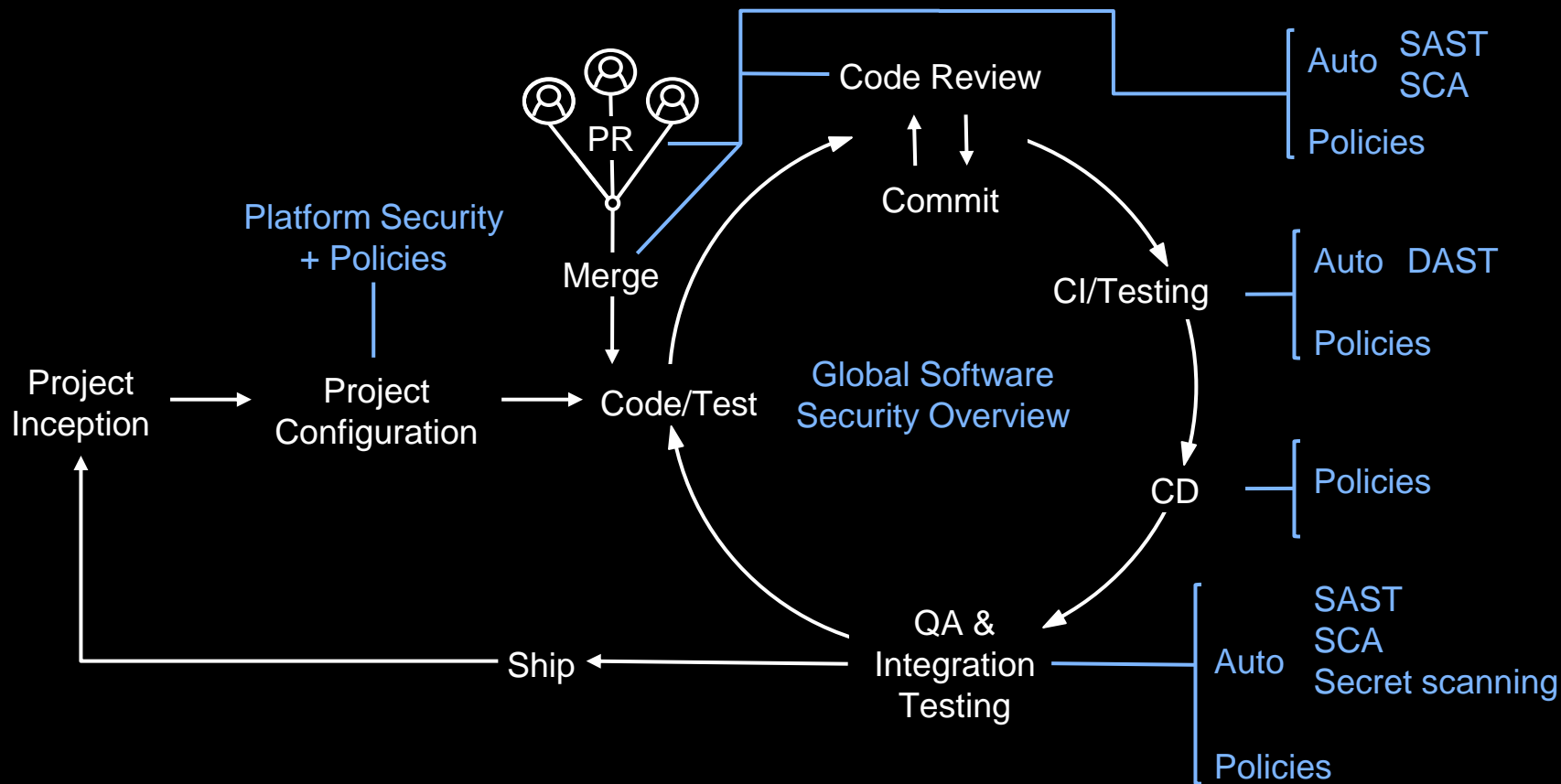
Basic Application Security scenario



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



supply
chain



code



platform

• **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
across 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.

The screenshot shows a GitHub pull request titled "Bump axios from 0.18.0 to 0.18.1 in /frontend #4". The pull request is created by the Dependabot bot and targets the master branch. A yellow banner at the top states: "This automated pull request fixes a security vulnerability" with a link to learn more about Dependabot security updates. The pull request description includes the following details:


- 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:** 99%
- 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**

The pull request is reviewed by [cmbolling](#) and is currently in progress. The right sidebar shows the pull request details, including the assignees (none), labels (dependencies, javascript), projects (none yet), milestone (none), linked issues (none), and notifications (subscribe).

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.

```
1 namespace DataModel
2 {
3     public static class LoginHelper
4     {
5         public static String ServiceUrl = "https://cloud.example.com";
6         public static String ClientID = "DataModel-0001";
7         public static String ClientSecret = "A002019DRBES$%FA
8         public static string RedirectURL = "Windows Security, A
9
10        /// <summary>
11        /// Handles acquiring all relevant tokens for the app
12        /// </summary>
13        /// <returns> Asynchronous task </returns>
14        /// </summary>
15        /// Handles acquiring all relevant tokens for the app
```



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

The screenshot displays the GitHub Code Scanning interface for a repository named 'dsp-testing / code-scanning-demo'. The left sidebar shows navigation options: Overview, Security policy, Security advisories (0), Dependabot alerts (0), Code scanning alerts (1), CodeQL, and Detected secrets (0). The main content area is titled 'Server-side URL redirect' with a 'Beta' label and a 'Give us feedback' link. Below the title, a description states: 'Server-side URL redirection based on unvalidated user input may cause redirection to malicious web sites.' There are three status buttons: 'Open' (green), 'Warning' (yellow), and 'CWE-601' (blue). A 'security' tag is also present. The code snippet is from 'test.ts' on the 'master' branch, showing a function 'sendRedirect' that sets a 'Location' header based on user input. A CodeQL alert is triggered at line 11, stating 'Untrusted URL redirection due to user-provided value.' Below the code, a table lists the tool (CodeQL), rule ID (js/server-side-unvalidated-url-redirection), and query (View source). The description of the rule explains that directly incorporating user input into a URL redirect request without validation can facilitate phishing attacks.

Search or jump to... Pull requests Issues Codespaces Marketplace Explore

dsp-testing / code-scanning-demo Private Watch 1 Star 0 Fork

<> Code Pull requests 1 Actions Security 1 Insights Settings

Overview

Security policy

Security advisories 0

Dependabot alerts 0

Code scanning alerts 1

CodeQL

Detected secrets 0

Server-side URL redirect (Beta) Give us feedback

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

Open Warning CWE-601 security

Branch: master

test.ts

```
8 */
9 const sendRedirect = async (res: ServerResponse, url: string, statusCode = 307) => {
10   res.statusCode = statusCode;
11   res.setHeader('Location', url);
12   await new Promise(resolve => res.end(resolve));
13 };
14
```

Untrusted URL redirection due to user-provided value.

CodeQL

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 these 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

Overview

Repositories

Projects

Packages

Teams

People

Security

Security

Overview

Risk

Coverage

Metrics

Secret scanning

Alerts

Dependabot

Code scanning

Secret scanning

You can only see data from repositories for which you have [permission](#) to view.

Overview

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

cURL

JavaScript

GitHub CLI

```
// 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