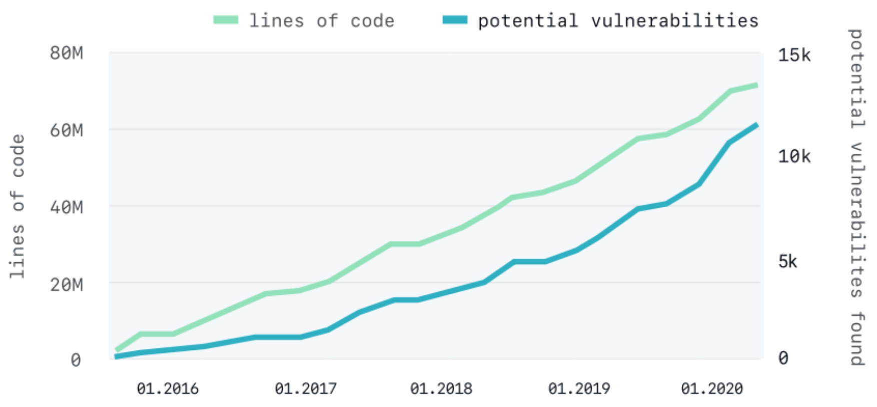
GitHub Advanced Security for Developers

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

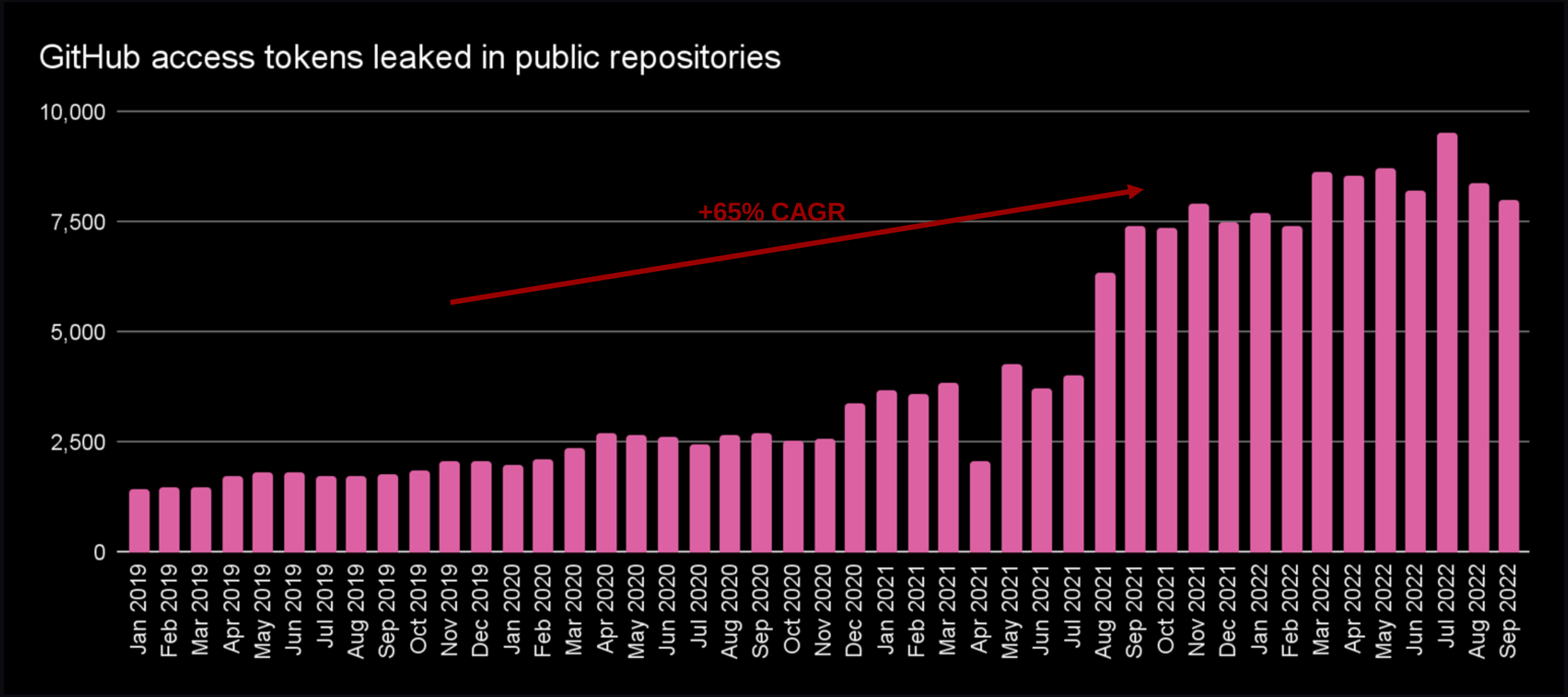
Source: Verizon Data Breach Investigations reports 2016, 2017, 2018, 2019 and 2020.

The state of AppSec

Is falling further behind the current state of Development

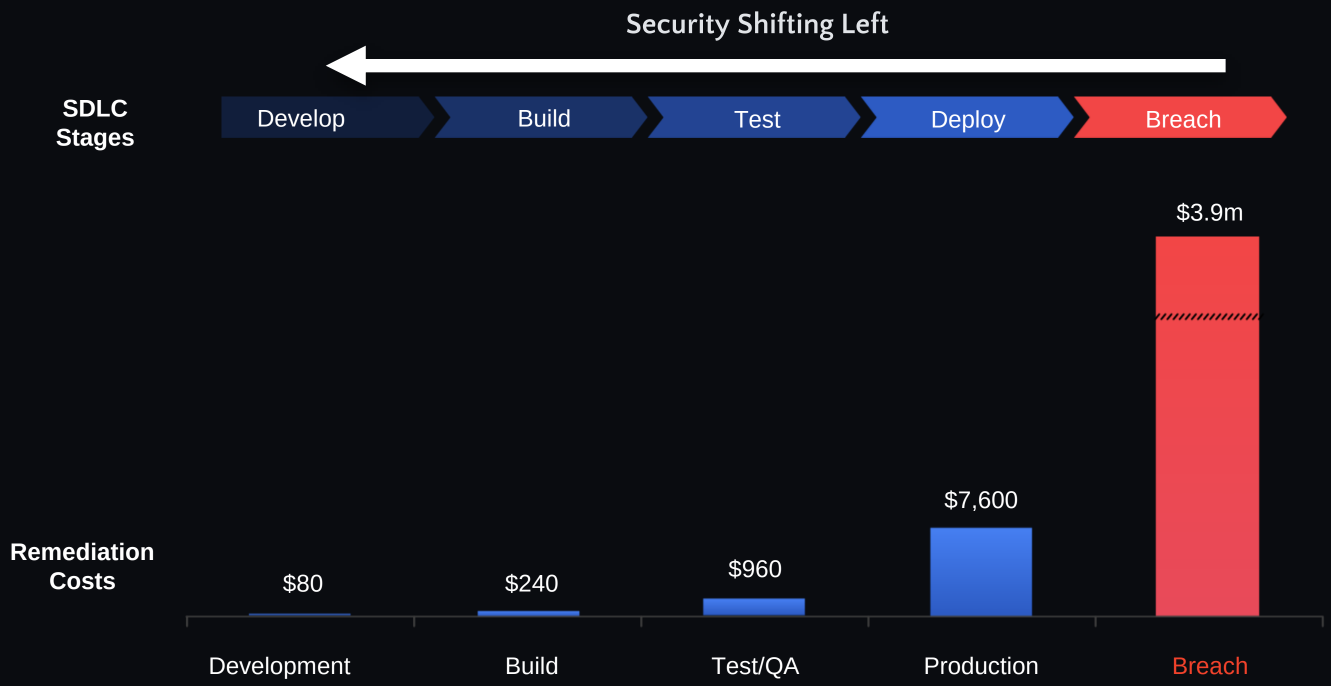
- 1:100 Security team members to developers
- Lack of knowledge voted the main AppSec challenge
- Remediation trends are stagnant

We’re seeing more credential leaks than ever



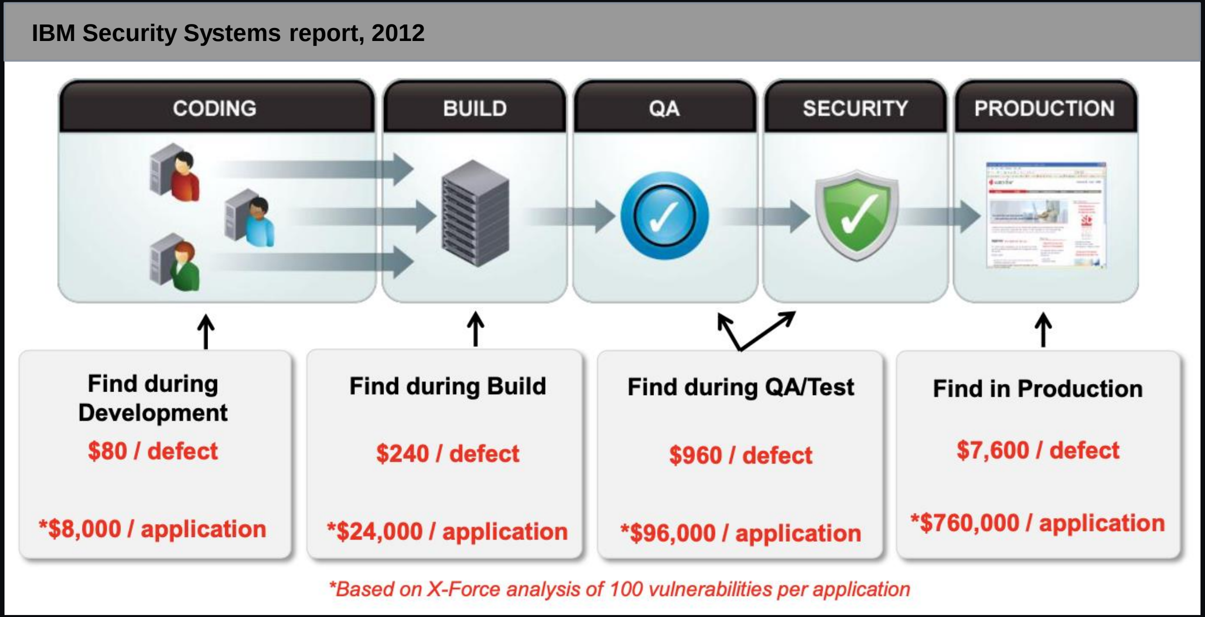
Source: GitHub data

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

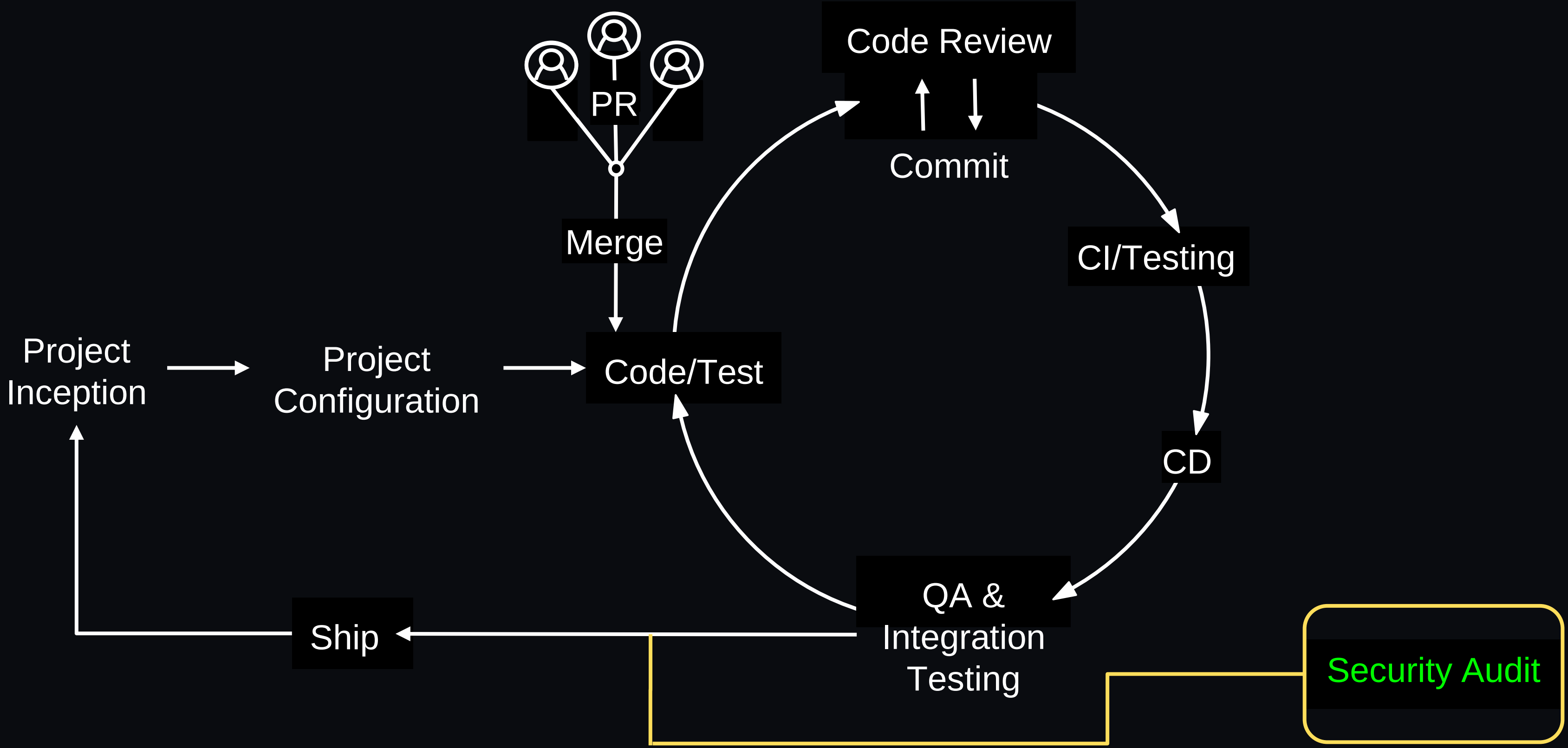
GitHub believes that making this shift requires a developer-first approach to all our security products:

- Integrate *directly* into the developer workflow.

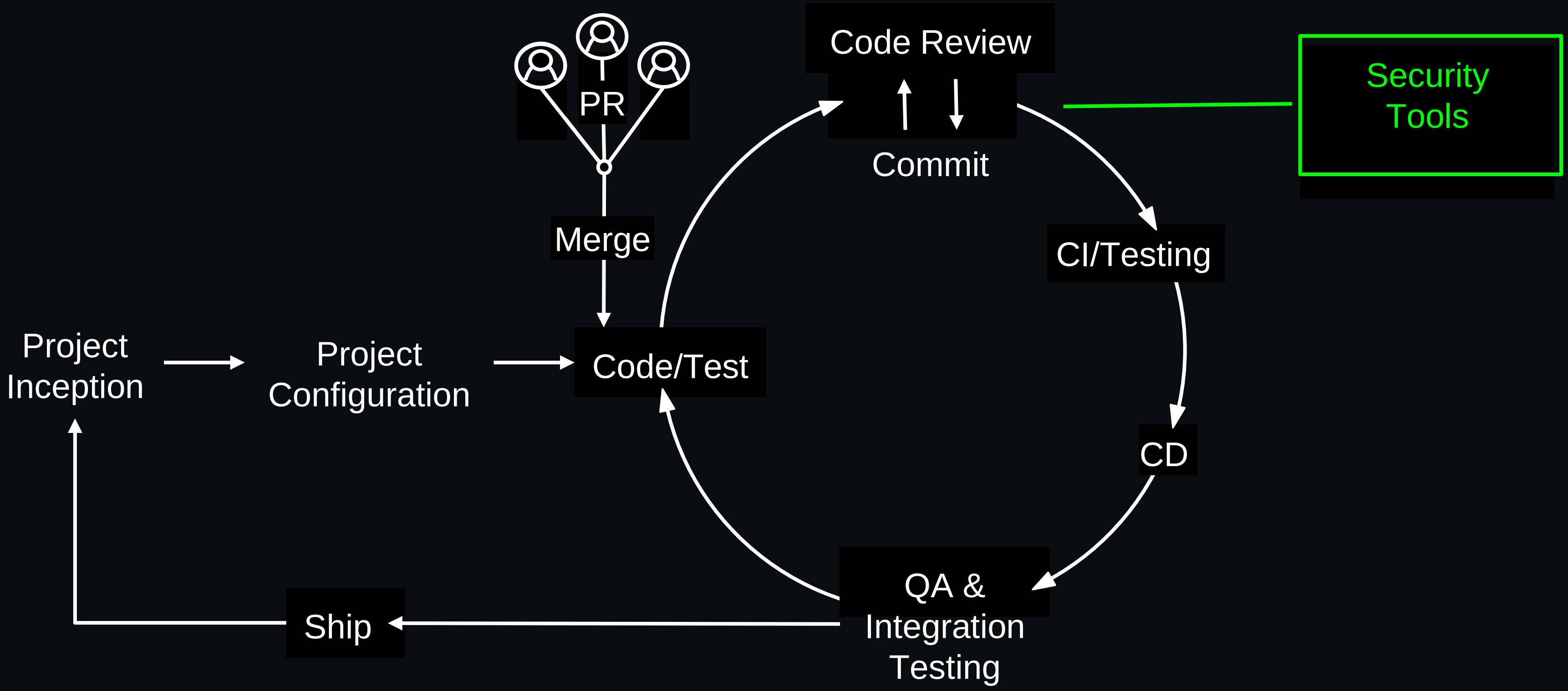
- Make setup and deployment fast and easy.

- Produce high quality results with low numbers of false positives.

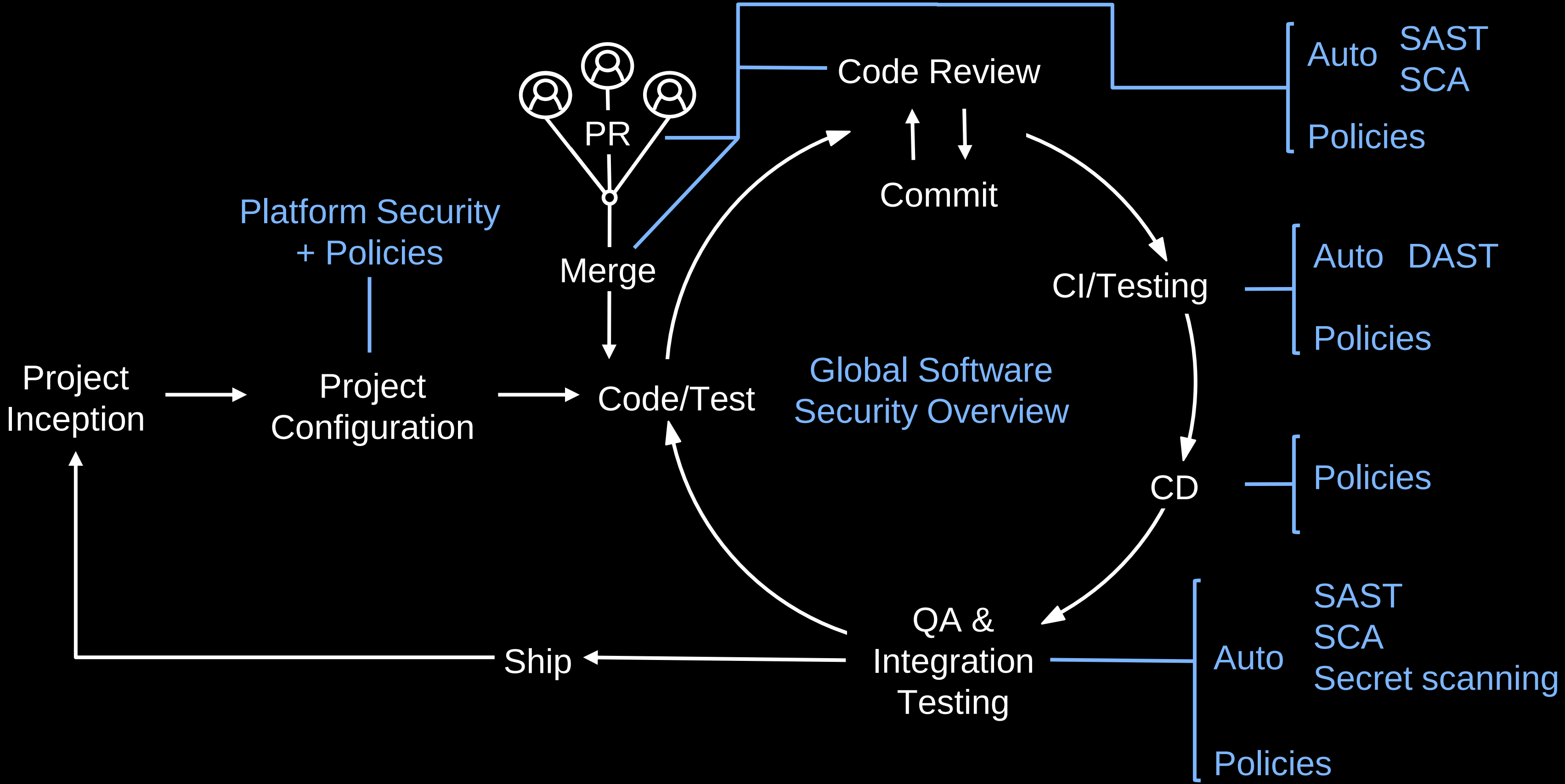
Basic Application Security scenario



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

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

code

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

platform

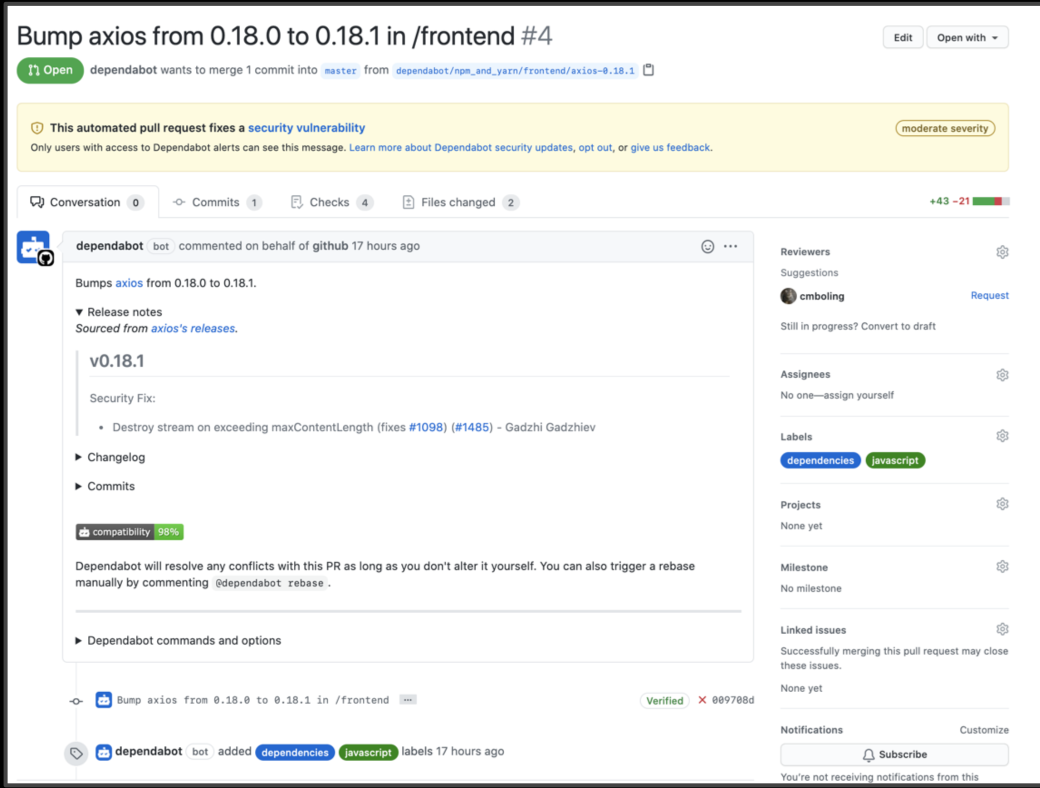
Branch protection
Enforce requirement for pushing to a branch or merging PRs

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.



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

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

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.

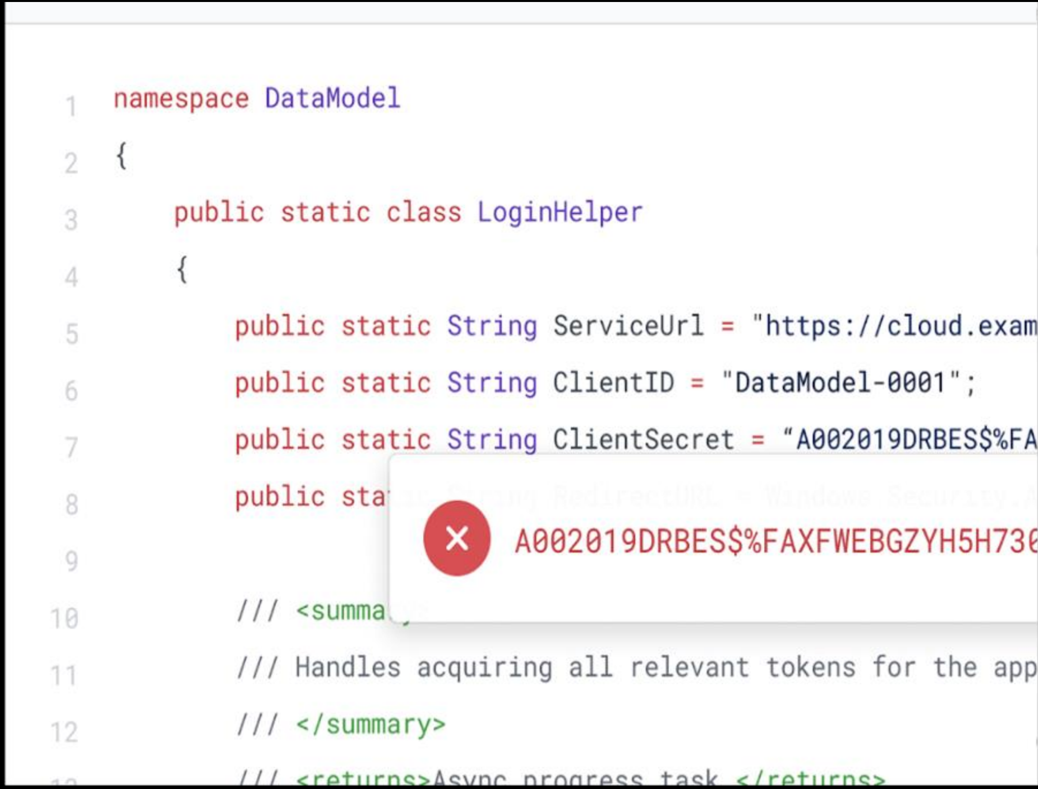
None yet

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.



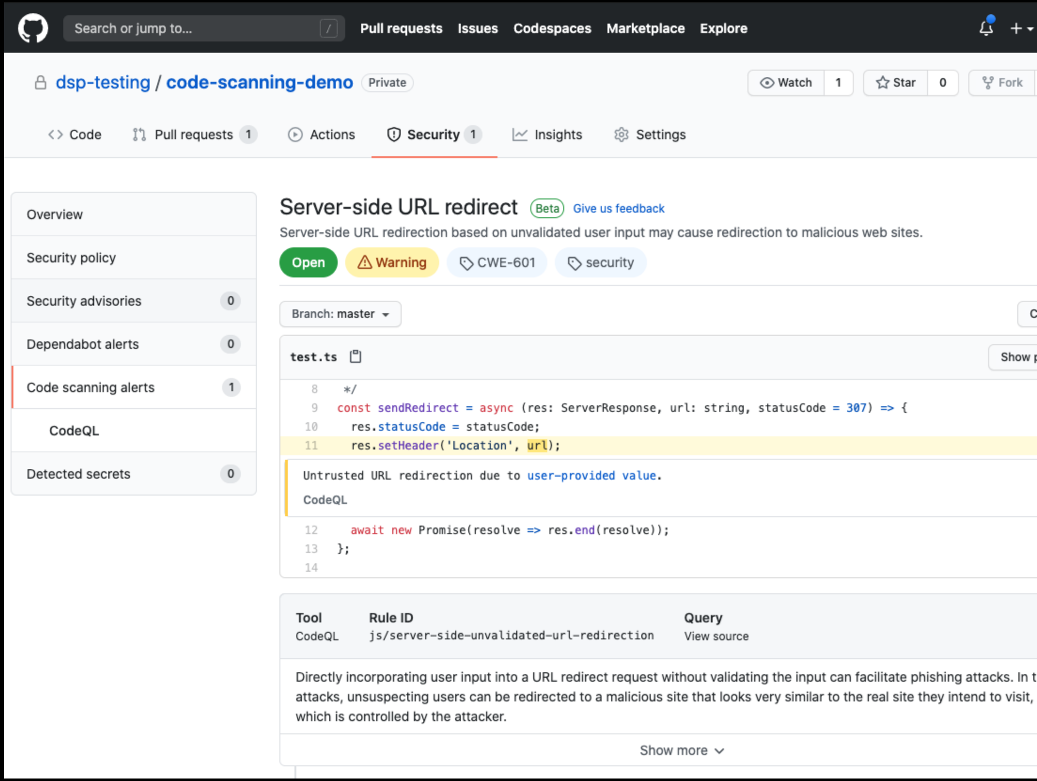
```
namespace DataModel
{
    public static class LoginHelper
    {
        public static String ServiceUrl = "https://cloud.exam
        public static String ClientID = "DataModel-0001";
        public static String ClientSecret = “A002019DRBES$%FA
        public sta

        /// <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



dsp-testing / code-scanning-demo Private

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.