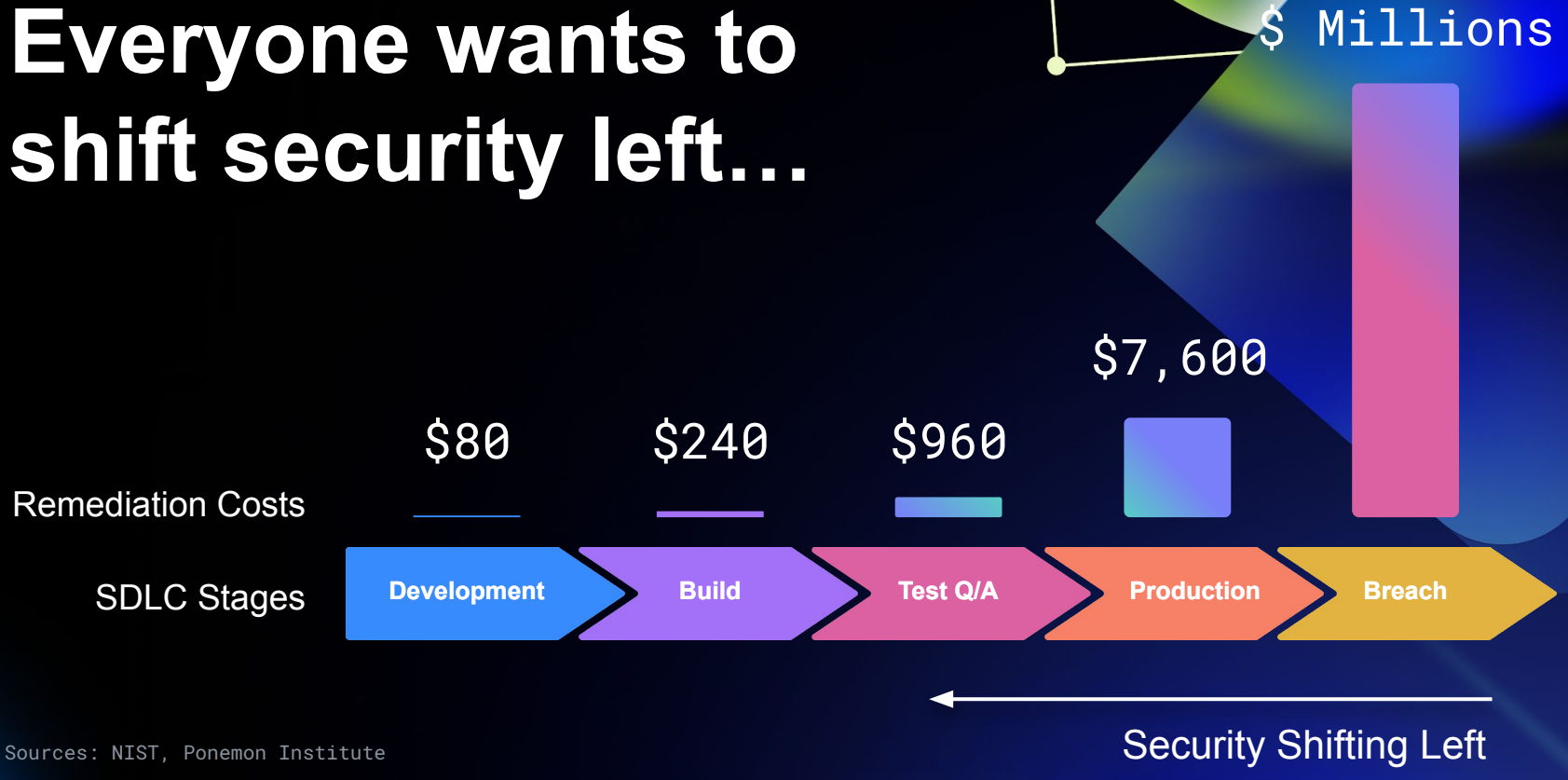
Shift left...
the right way

Christopher Harrison
@geektrainer
Senior Developer Advocate

Everyone wants to shift security left…



Sources: NIST, Ponemon Institute

Security debt is pervasive

Security debt is defined as all security flaws and vulnerabilities that remain unremediated for over one year

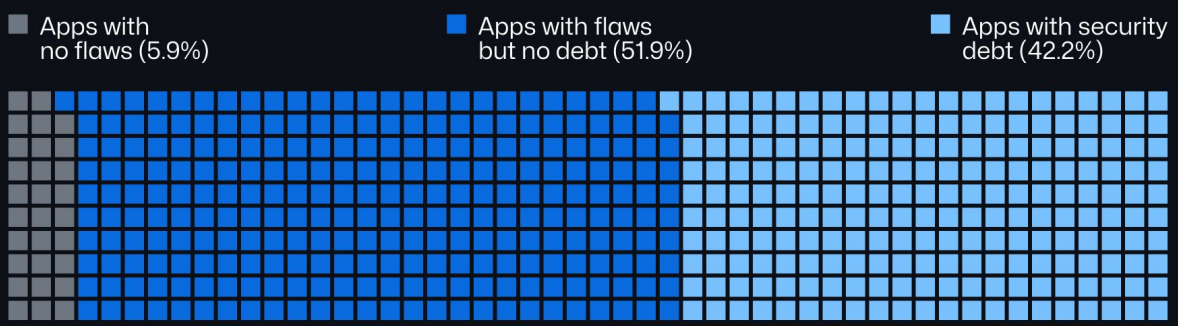


Figure 8: Prevalence of security debt across all applications greater than one year old

70.8%

of organizations have security debt

89.4%

of security debt exists in first party code

46.6%

of security flaws in the median organization will be Security Debt

Source: Veracode SOSS 2024

Flaws but no debt: Flaws exist in security backlog but <1 year old
Security debt: FLaws and vulnerabilities >1 year old

Agenda

How we currently do things

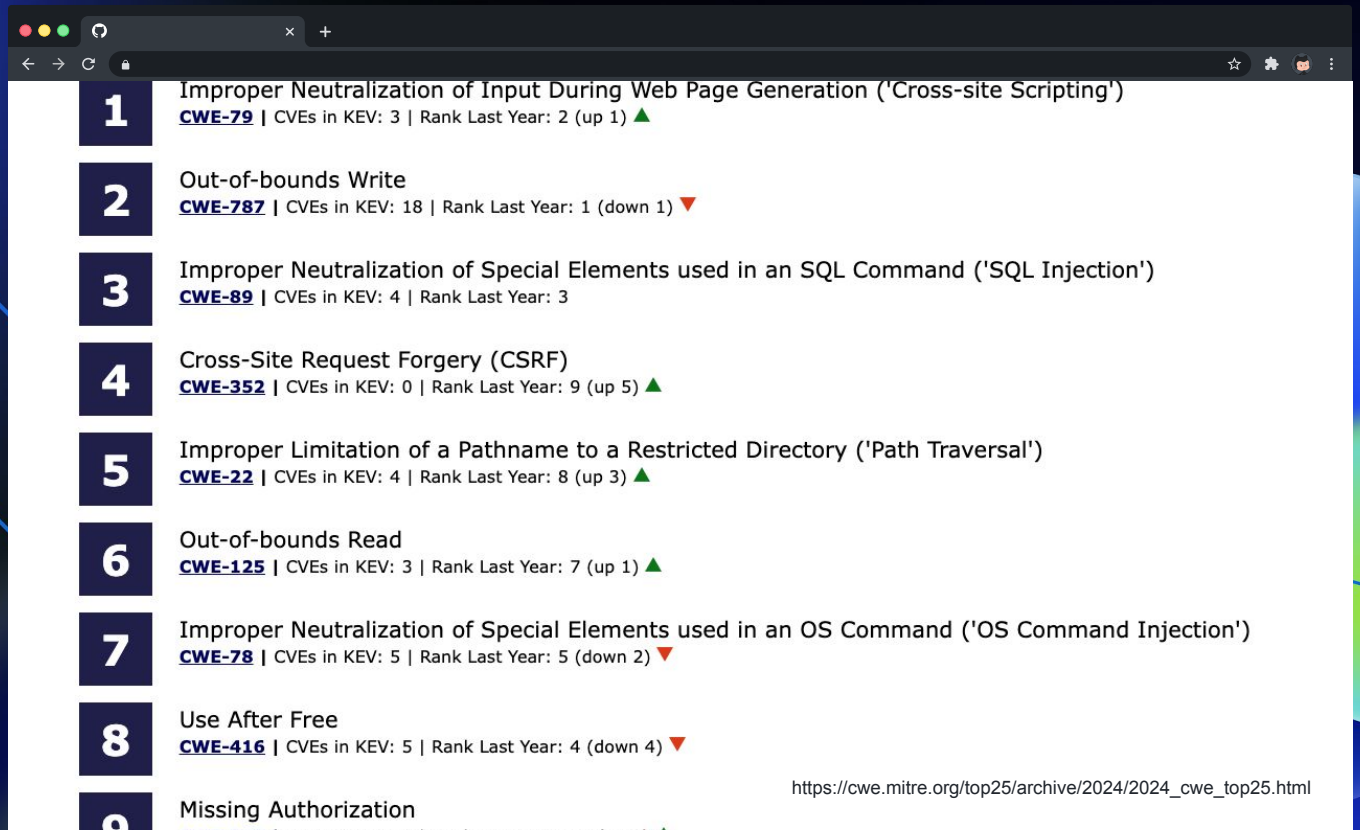
How we should do things

How we put things into action

How we currently do things

Make developers security professionals

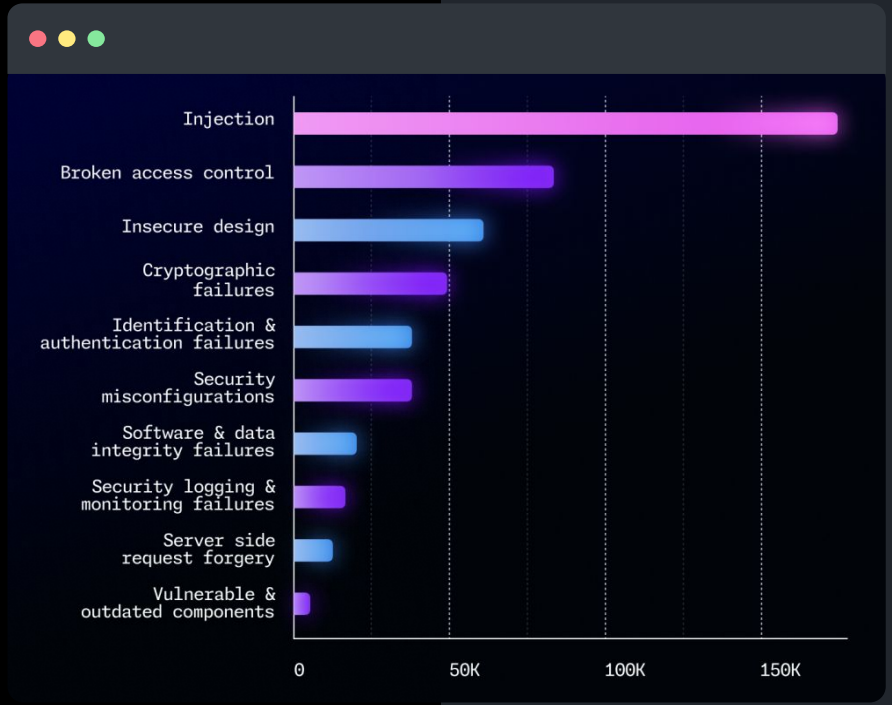
The top weaknesses are not new



1. Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')
   CWE-79 | CVEs in KEV: 3 | Rank Last Year: 2 (up 1) ▲
2. Out-of-bounds Write
   CWE-787 | CVEs in KEV: 18 | Rank Last Year: 1 (down 1) ▼
3. Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')
   CWE-89 | CVEs in KEV: 4 | Rank Last Year: 3
4. Cross-Site Request Forgery (CSRF)
   CWE-352 | CVEs in KEV: 0 | Rank Last Year: 9 (up 5) ▲
5. Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')
   CWE-22 | CVEs in KEV: 4 | Rank Last Year: 8 (up 3) ▲
6. Out-of-bounds Read
   CWE-125 | CVEs in KEV: 3 | Rank Last Year: 7 (up 1) ▲
7. Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')
   CWE-78 | CVEs in KEV: 5 | Rank Last Year: 5 (down 2) ▼
8. Use After Free
   CWE-416 | CVEs in KEV: 5 | Rank Last Year: 4 (down 4) ▼
9. Missing Authorization

https://cwe.mitre.org/top25/archive/2024/2024_cwe_top25.html

Most common types of vulnerabilities found by CodeQL in 2024



Most common vulnerability, per language

GitHub public repos
Detected by Code Scanning
Aug. 2023 - Aug. 2024

Javascript
Incomplete sanitization (XSS)

34k alerts

Python
Clear-text logging of sensitive data

32k alerts

Java
Cross-site scripting (XSS)

21k alerts

C/C++
Unsafe type casting

20k alerts

Go
Incorrect integer conversion

8k alerts

C#
Execution of arbitrary system commands

3k alerts

Hackers steal data of 2 million in SQL injection, XSS attacks

By Bill Toulas

February 6, 2024 02:00 AM 1

A threat group named 'ResumeLooters' has stolen the personal data of over two million job seekers after compromising 65 legitimate job listing and retail sites using SQL injection and cross-site scripting (XSS) attacks.

POPULAR STORIES

Zero-click Windows TCP/IP RCE impacts all systems with IPv6 enabled, patch now

Microsoft retires Windows updates causing 0x80070643 errors

Why does this still happen?

Developers are taught insecure methods

Often tutorials bypass proper security

Security training isn’t impactful

“How does this impact me?”

Difficult to see the impact of secure code

There’s no red/green process like testing

Well-meaning developers take shortcuts

“I just need this to work. I’ll fix it later.”

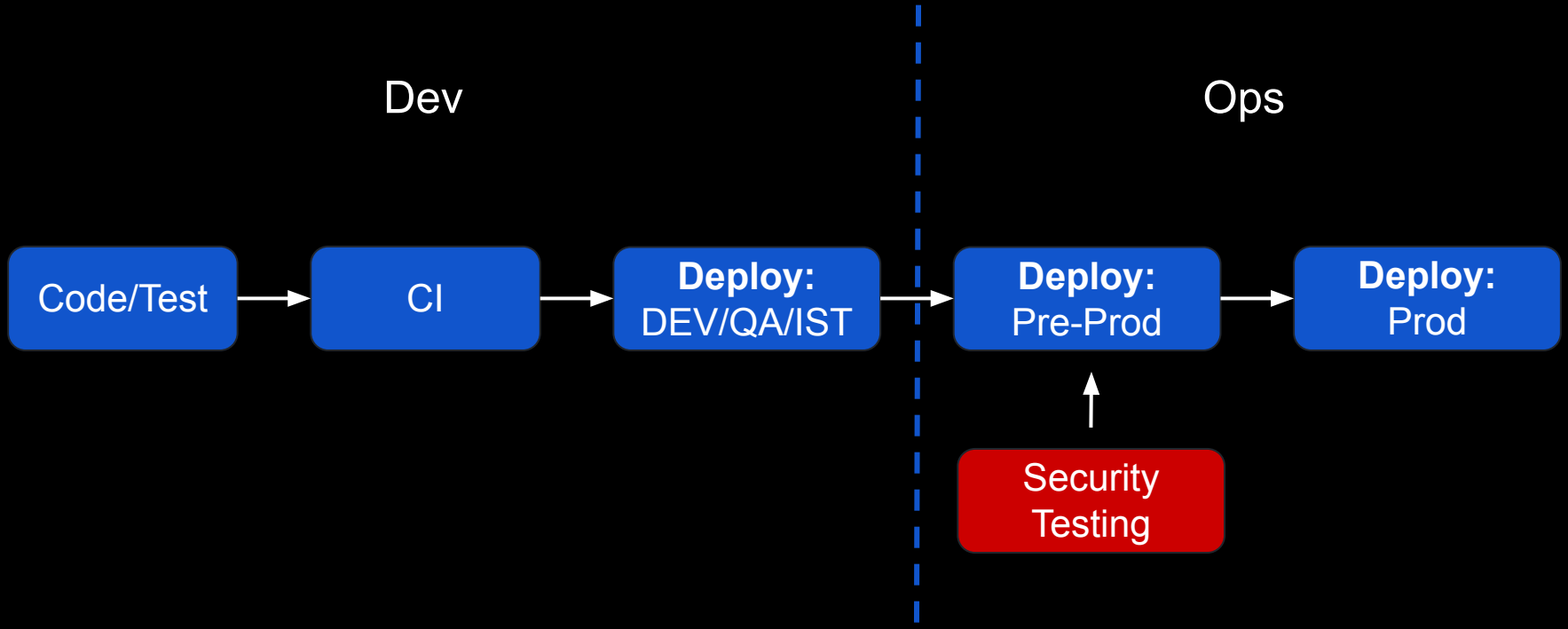
Hire security teams!

The Dev-Sec Divide

Developers

Security

DevOps Pipeline



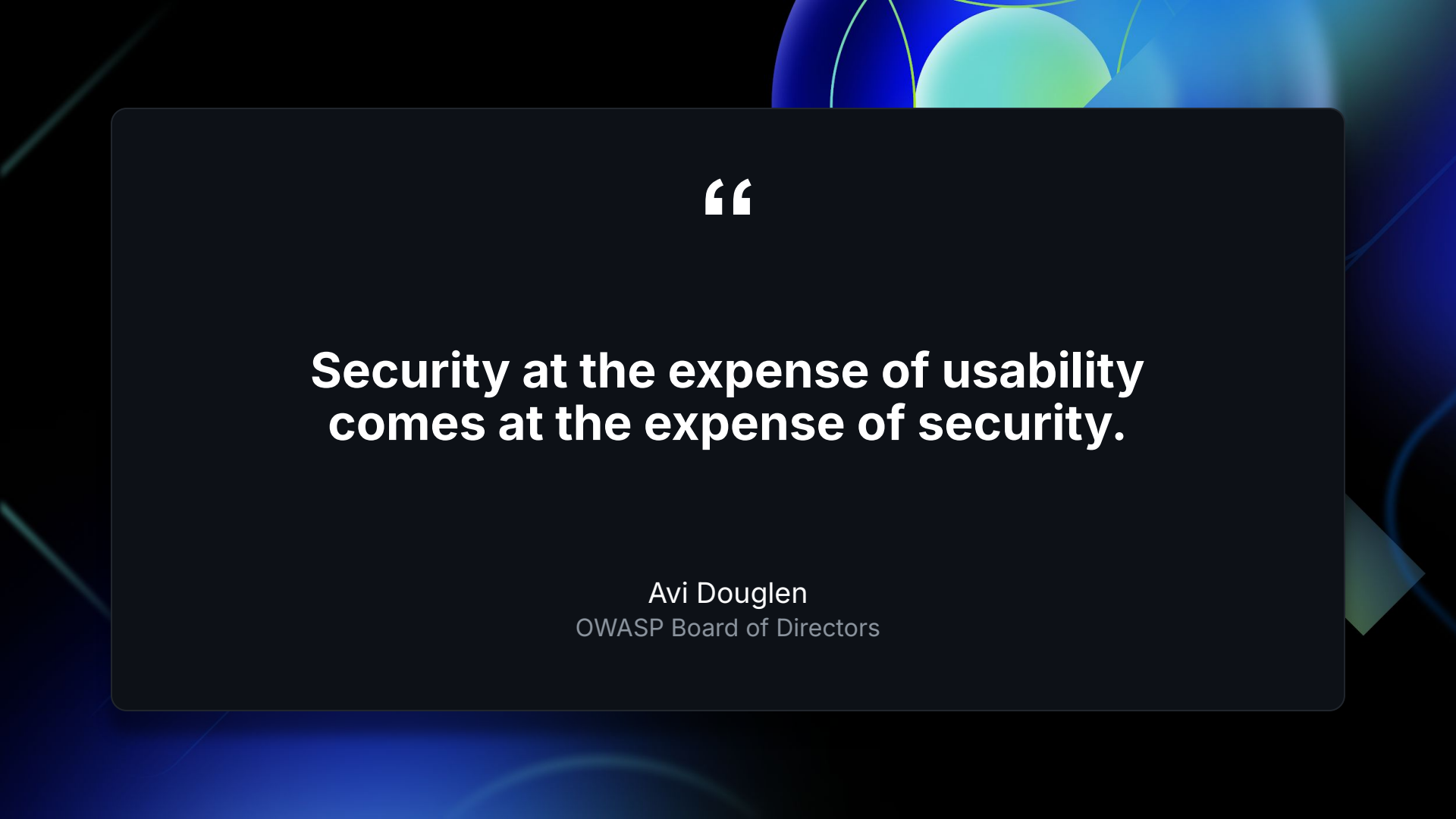
Separate teams with (what feels like) separate missions

Always outnumbered (and overworked)

Traditionally involved late in the process

Unfairly criticized for “misaligned objectives”

BUY MORE TOOLS!!


“
Security at the expense of usability comes at the expense of security.
Avi Douglen
OWASP Board of Directors


CRAFT

New tools don’t magically deliver new culture

Shiny objects are quickly forgotten

Wrong tools for the job

Wrong tools for the culture

How we should do things

But first… Let's define DevOps and DevSecOps

Key focus areas

Assume best intent
Nobody is checking in unsecure code on purpose

We’re a team
Everyone has different roles and strengths; let’s play to those

Review the output
Give feedback on the code and the process, not the humans involved

Automate automation
Streamline the process as much as possible

How do we do the thing?

Take a swiss cheese approach

The code we write

A moment for our overloaded devs

Remember:
good DevOps is key to good DevSecOps

Supporting developers

Code smells
Teach developers to involve security early when they're working on something which "smells" interesting

Inline upskilling
Show developers on **their code** where problems arise and how to best avoid them

Documentation
Devs want to do the right thing; they just don't know how

Team work
We're all on the same team with the same goals

The role of GenAI


It looks like you're trying to create a login page. Would you like help?

How GenAI can help developers

Explain code

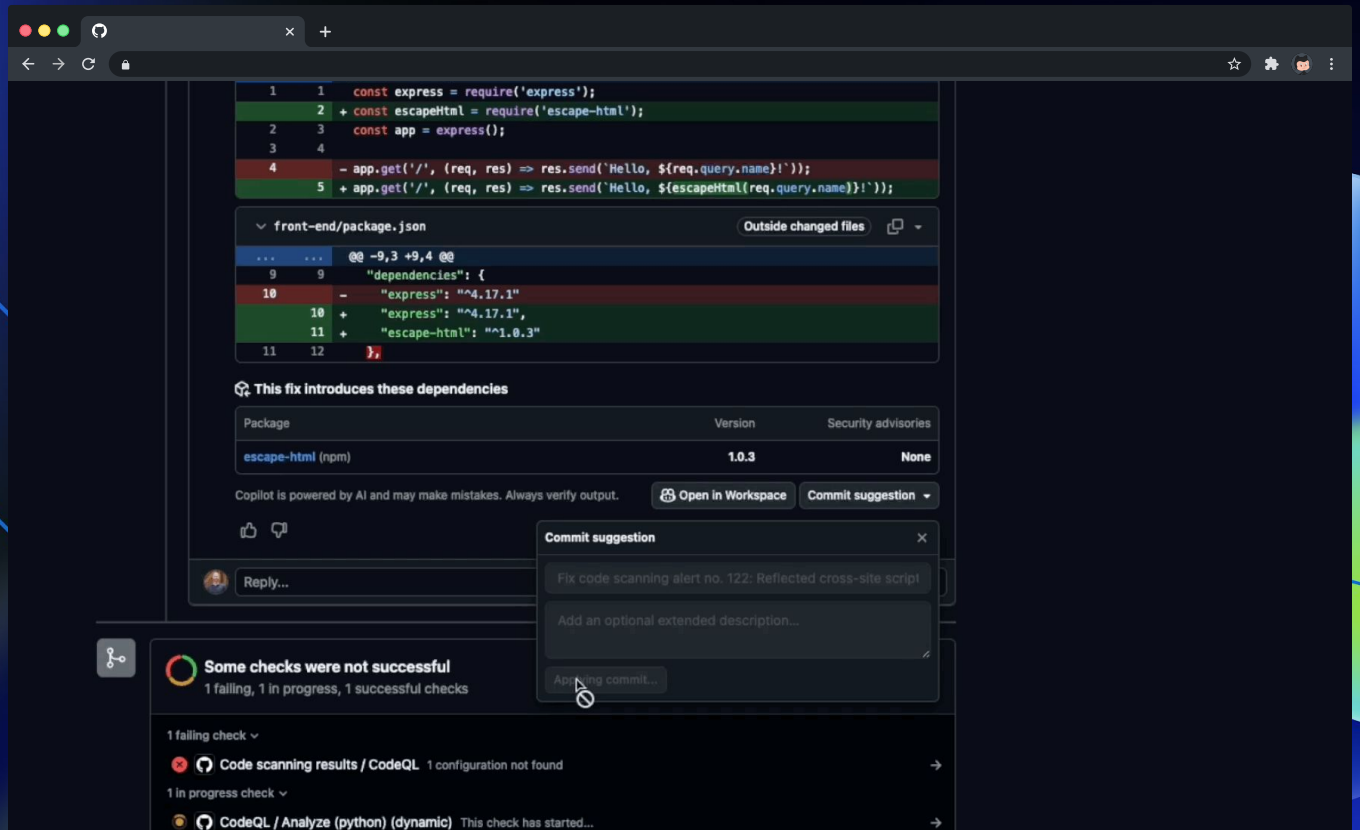
Ask your assistant if there are any potential issues and how to resolve them

Create code

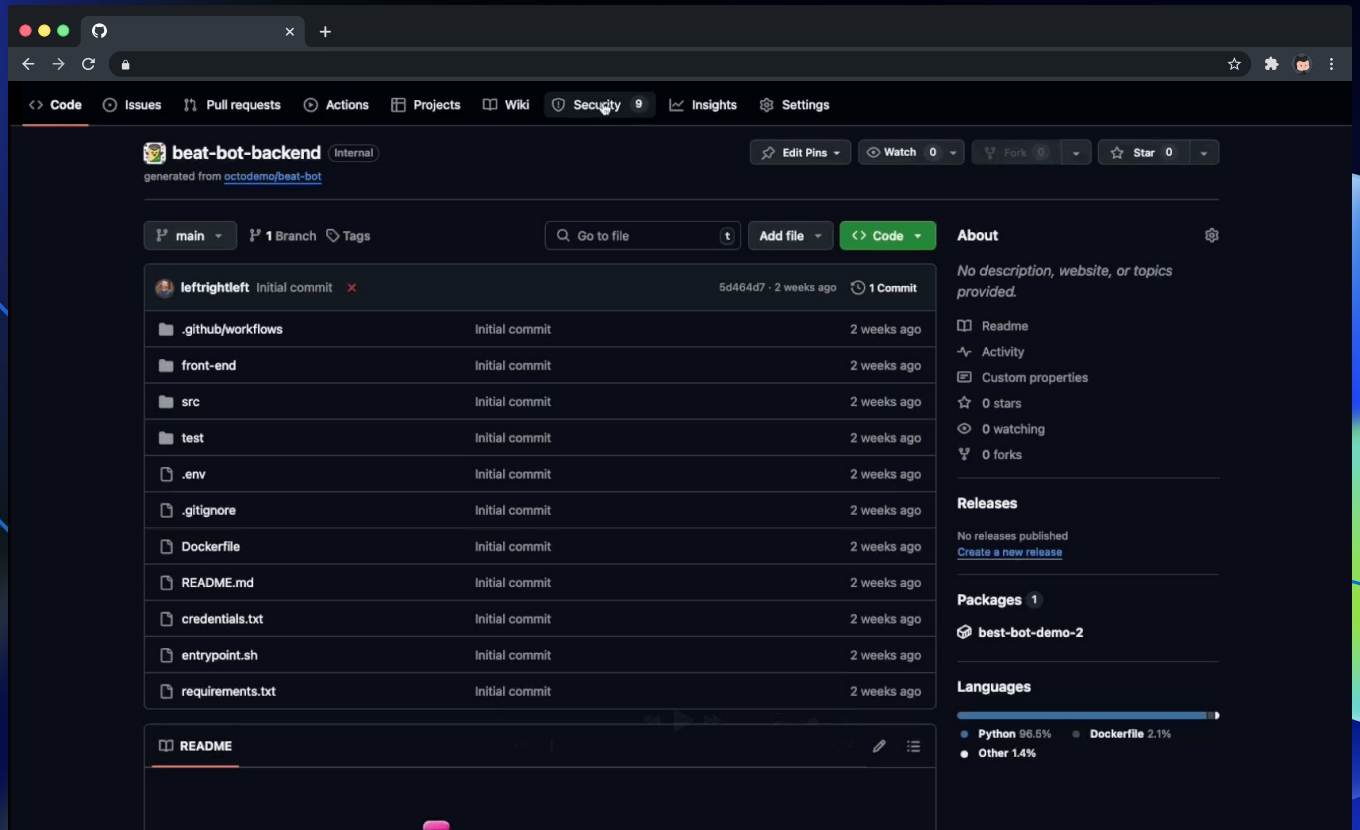
AI assistants can generate secure code either by design or as prompted

What if the code is already written?

Autofix in the pull request



Autofix for existing alerts



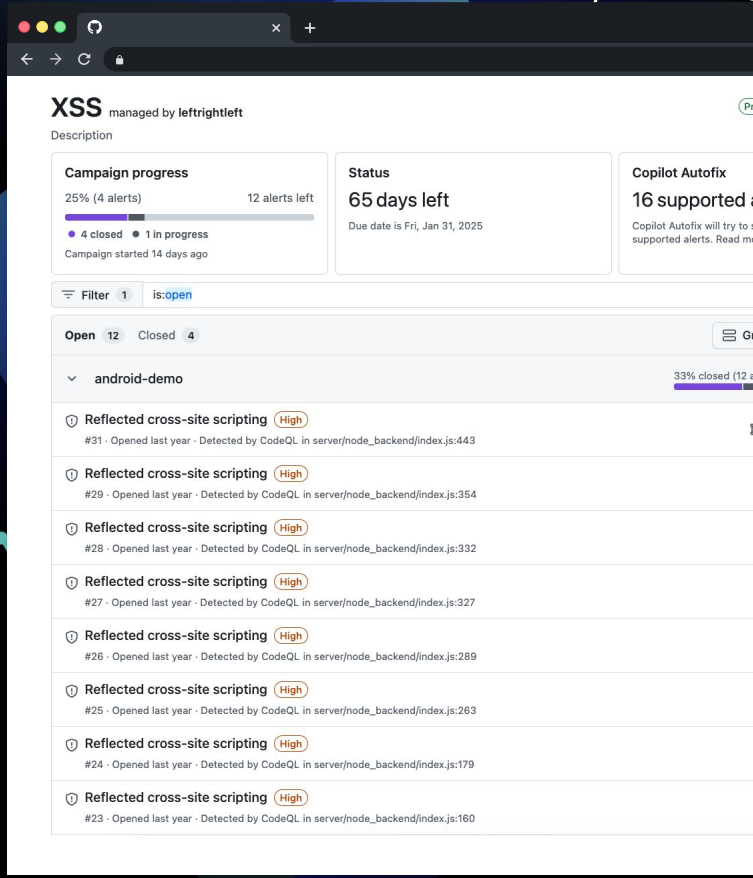
Campaigns

Define the highest-risk problems

Security teams define the subset of vulnerabilities to resolve

Track the resolution of issues

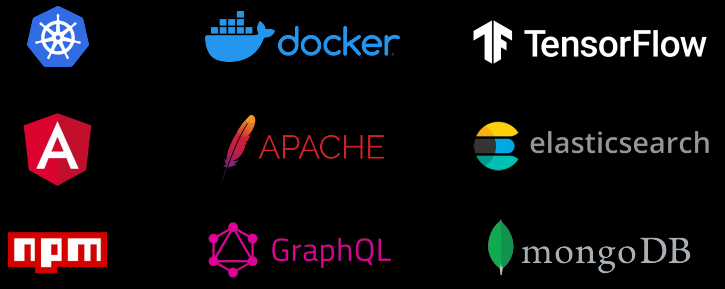
Visibility into the state of remediation

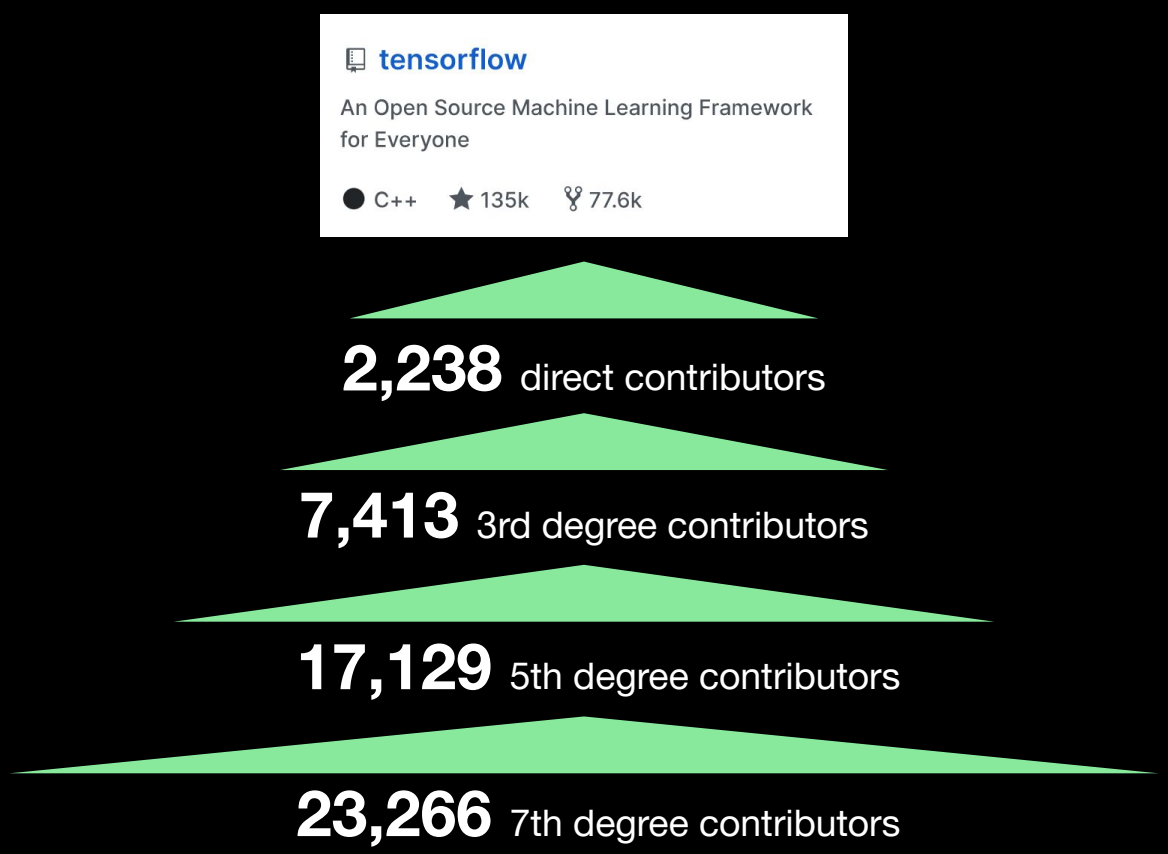


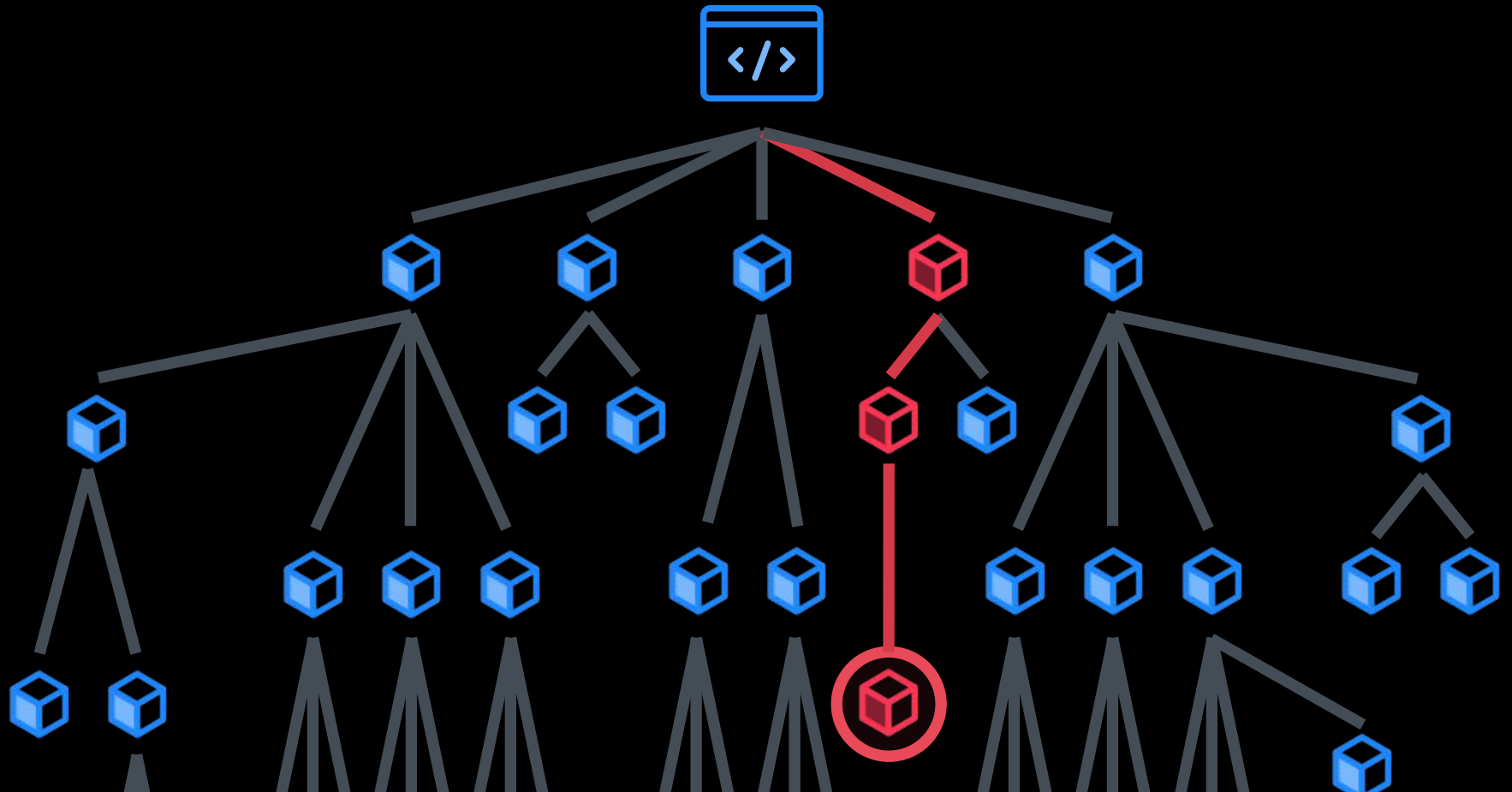
The code we use

90%

of code in new applications is open source




tensorflow
An Open Source Machine Learning Framework for Everyone
C++
135k
77.6k
2,238 direct contributors
7,413 3rd degree contributors
17,129 5th degree contributors
23,266 7th degree contributors



Staying current

“Bump the version”

Only possible if you have good DevOps processes

Time to rewrite?

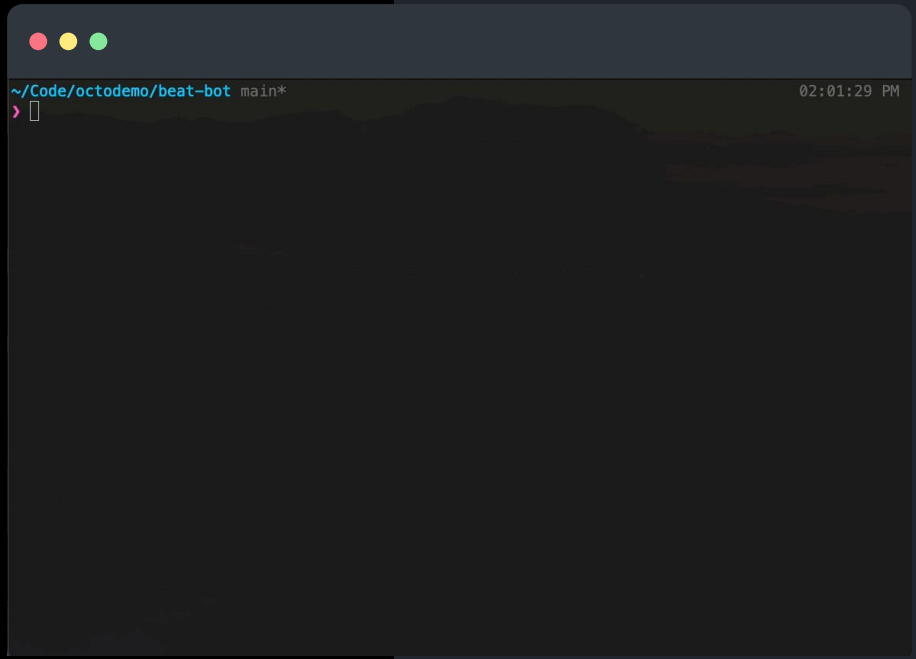
Longer the wait, the more expensive the fix

Keys in code

Pop quiz:
How many times per month are AWS credentials written to public GitHub repos?

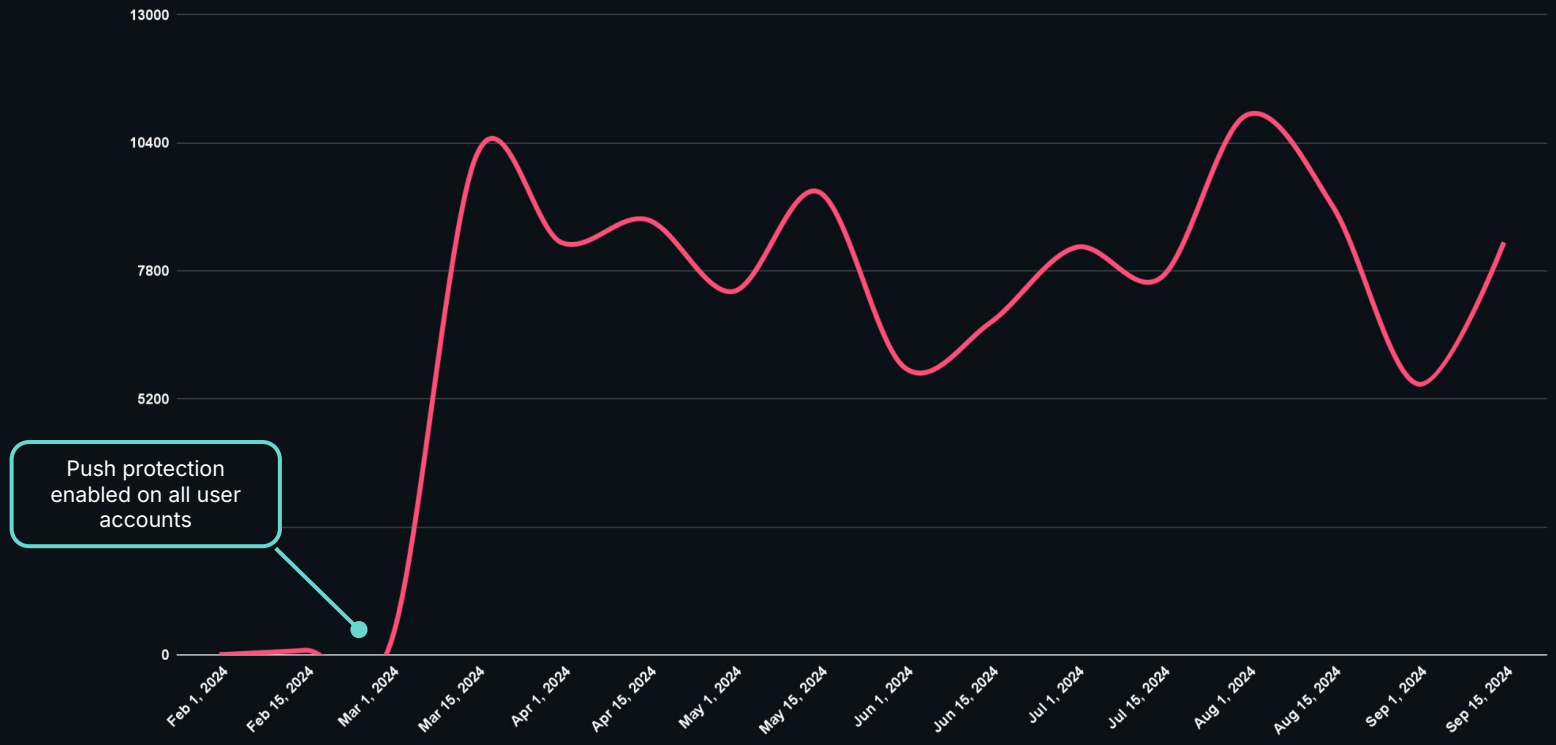
Scan for secrets and block on push

- Block new credentials from being written to GitHub
- Provides a bypass mechanism



Push Protection Blocks

GitHub public repos, Feb. 2024 - Aug. 2024



Credential considerations

Key management

How easy is it for your devs to do the right thing?

Mitigation plan

What's your plan for rotating and removing keys?

Where do we go from here?

Use the right tools

Play to people’s strengths

Have empathy

Thank you