

Driving Cultural Change

Mending the bond between
Security and Development



Brent Beer, Director Solutions Engineering

October 24, 2023

Driving Cultural Change

Mending the bond between
Security and Engineering



Brent Beer, Director Solutions Engineering

October 24, 2023

Driving Cultural Change

Mending the bond between
AppSec and Developers



Brent Beer, Director Solutions Engineering

October 24, 2023

Disney · PIXAR

BRAVE





Dev and Sec today



Problems today

- Frustration of interrupted work items / sprint planning
- Ignorance to being the source of a vulnerability
- Assuming malice (of the security team)

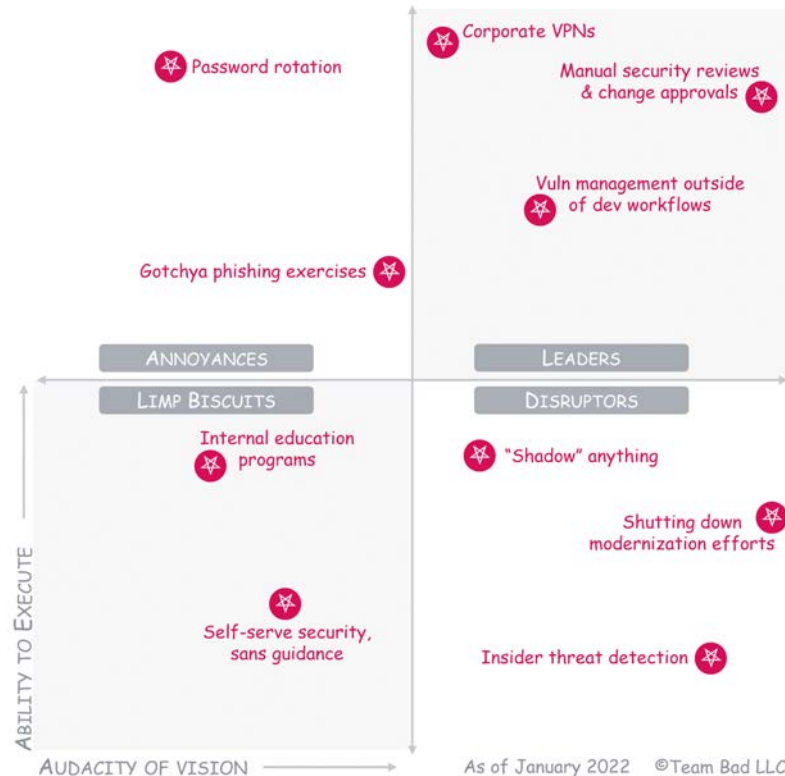




Problems today

- Security has been tasked to protect, and that's what we'll do at all costs!
- Silo of decision making
- Manual reviews will make you a hero if you catch something
- 1:many personnel problems

Figure 1: Infernal Quadrant for Security Obstructionism (SecObs)



Source: Kelly Shortridge (January 2022)



Problems today

- “And I suppose a princess just does as she’s told?”





What kind of security organization do you have?

Why are we still trying to fix this?

Shutting down modernization initiatives

Change is hard

Slow down adoption of new tech

No time to change again, we're constantly changing and we have fatigue

Not the right audience

Who built the tool that the developers are now using?
Was it developers?

Interrupted Flow

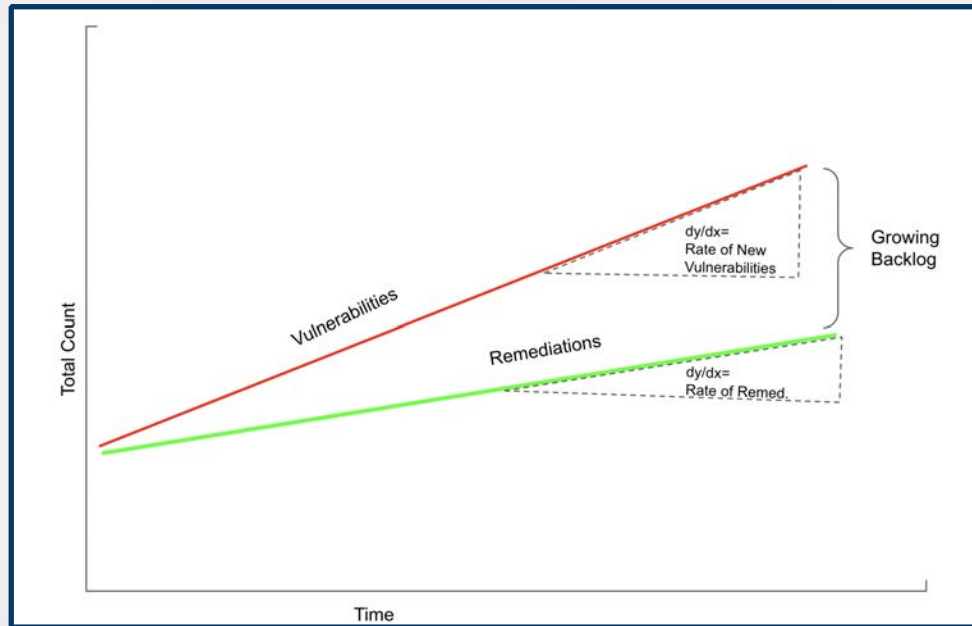
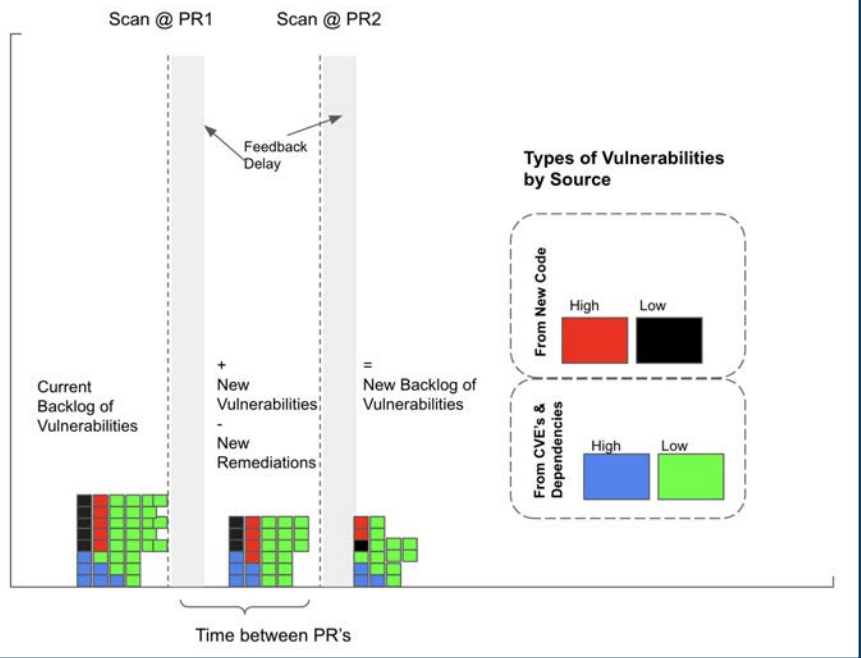
Is other work impacted because we're constantly switching or needing to create reports to talk about what's happening?

Common Space

Where are we talking about this?



Data About a Problem





What is InnerSource?



The application of open source principles to internal company software development



A journey to culturally transform towards a sharing economy



The core principles of open source and community collaboration within the walls of an organization.

This involves building an internal community, collaborative engineering workflow, and culture



The GitHub experience



A single platform
for your tools to
plug into



Collaborative tools for
inclusive
conversations



Security at
every step of
the workflow



Powered
By AI



Security

```
→ ~/my_project git:(branch_name) git push
remote: error GH009: Secrets detected! This push failed.
remote:
remote:          GITHUB PUSH PROTECTION
remote: _____
remote:  Resolve the following secrets before pushing again.
remote:
remote:  (?) Learn how to rewrite your local commit history
remote:  https://git-scm.com/book/en/v2/Git-Tools-Rewriting-History
remote:
remote:
remote: — GitHub Personal Access Token —————
remote:  locations:
remote:    - commit: c47ff8afc1ce530798ce62e064c28fe26c33c99b
remote:      path: src/config/credentials.js:12
remote:
remote:  (?) To push, remove secret from commit(s) or follow this
URL to allow the secret.
```



Secret scanning alerts / #403

Google API Key

Close as ▾

[Open](#) GitHub detected a secret on Jan 19

Secret inactive

Last verified 1 hour ago

```
AIzaSyAQfxPJiouj0DE05ZieffeBv6yft2Q
```



Remediation steps

Follow the steps below before you close this alert.

- 1 Review the Google API Key through Google to ensure that the secret has not been used for unauthorized access. [Learn more about Google tokens.](#)
- 2 Check security logs for potential breaches.
- 3 Close the alert as revoked.

Detected in 2 locations [Beta](#) [Give us feedback on issue detections](#)

🔍 Using a potential API key #144

mcantu commented on Jan 19

> README.md



GitHub opened this alert on Jan 19



Security

Code scanning alerts / #1139

Missing JWT signature check

Dismiss alert ▾

Open in main 3 days ago Tracked by #68

```
webgoat-lessons/jwt/src/main/java/org/owasp/webgoat/jwt/JWTVotesEndpoint.java:163
160     return failed(this).feedback("jwt-invalid-token").build();
161   } else {
162     try {
163       Jwt jwt = Jwts.parser().setSigningKey(JWT_PASSWORD).parse(accessToken);
```

This parser sets a JWT signing key, but the signature is not verified.

CodeQL

```
164     Claims claims = (Claims) jwt.getBody();
165     boolean isAdmin = Boolean.valueOf((String) claims.get("admin"));
166     if (!isAdmin) {
```

Tool	Rule ID	Query
CodeQL	java/missing-jwt-signature-check	View source

A JSON Web Token (JWT) consists of three parts: header, payload, and signature. The `io.jsonwebtoken:jjwt` library is one of many libraries used for working with JWTs. It offers different methods for parsing tokens like `parse`, `parseClaimsJws`, and `parsePlainTextJws`. The last two correctly verify that the JWT is properly signed. This is done by computing the signature of the combination of header and payload and comparing the locally computed signature with the signature part of the JWT.

Show more ▾

- First detected in commit on Sep 4, 2021 in configuration `.github/workflows/codeql.yml:analyze:language:java`
- Update README.md (#45) Verified 2b6cef9
webgoat-lessons/jwt/src/main/java/org/owasp/webgoat/jwt/JWTVotesEndpoint.java:163 on branch main
- Appeared in branch `refs/pull/46/merge` on Oct 26, 2021 in configuration `.github/workflows/codeql.yml:analyze:language:java`
Commit 660bbd37 (language: java)
- leftrighthigh closed this as false positive on Oct 20, 2022
Validated
- leftrighthigh reopened this on Oct 20, 2022

Severity

High

Affected branches

- main 2
- BrightnBubbly-patch-1
- rafskov-demo

Tags

security

Weaknesses

CWE-347



Security

Security

- Risk**
- Coverage
- Overview Private beta

- Metrics
- Coverage Private beta
 - Secret scanning Beta

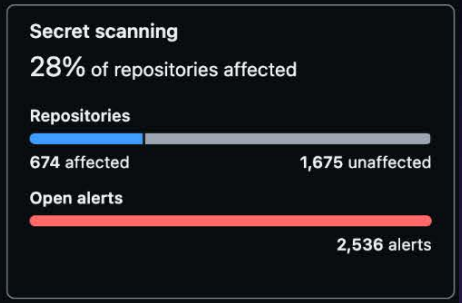
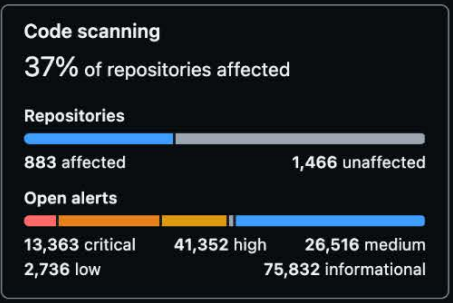
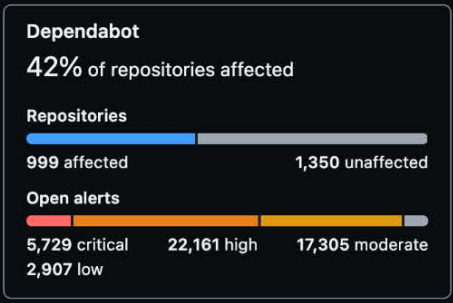
- Alerts
- Dependabot 5,000+
 - Code scanning 5,000+
 - Secret scanning 2,581

Security risk

[Give feedback](#)

Alert counts for security features in repositories across the organization

Q archived:false × Teams ▾ Export CSV



2,349 Active 22 Archived Sort by: Recently updated ▾



Security

Summary

Jobs

❌ dependency-review

Run details

🕒 Usage

📄 Workflow file

dependency-review

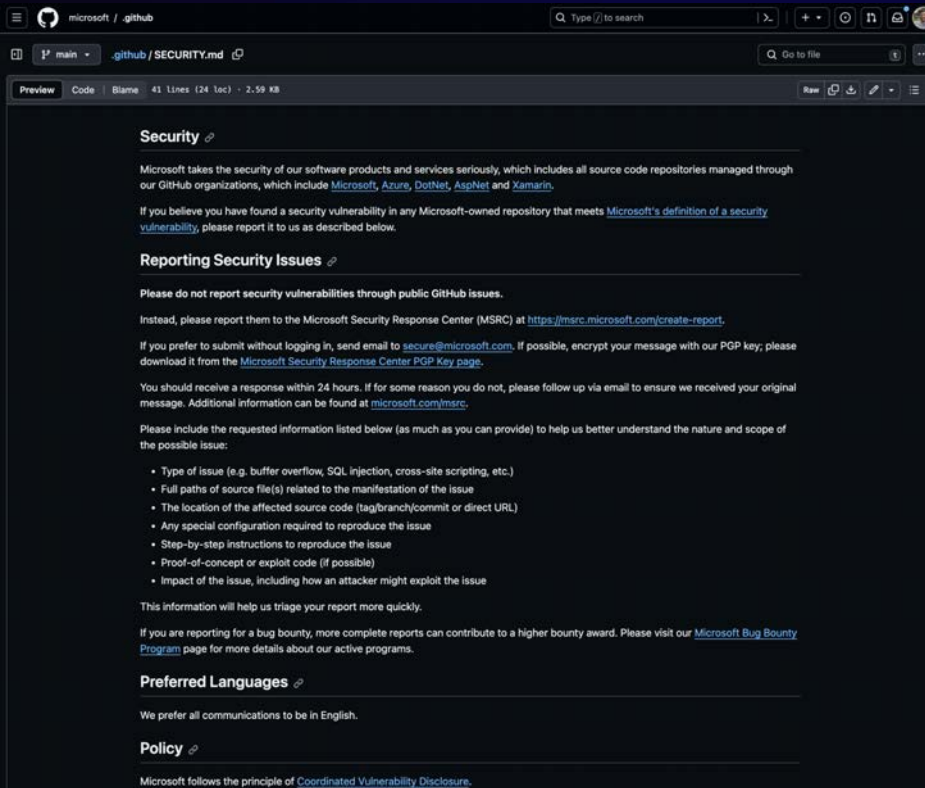
failed on Dec 16, 2022 in 7s

🔍 Search logs



- > ✅ Set up job 2s
- > ✅ Checkout Repository 1s
- ▼ ❌ Dependency Review 2s
 - 1 ▶ Run actions/dependency-review-action@v3
 - 7 ▼ Vulnerabilities
 - 8 package.json » parse-server@4.10.19 - Parse Server is vulnerable to Prototype Pollution via Cloud Code Webhooks (high severity)
 - 9 ↪ <https://github.com/advisories/GHSA-93vw-8fm5-p2jf>
 - 10 target/package.json » parse-server@4.10.19 - Parse Server is vulnerable to Prototype Pollution via Cloud Code Webhooks (high severity)
 - 11 ↪ <https://github.com/advisories/GHSA-93vw-8fm5-p2jf>
 - 12 **Error:** Dependency review detected vulnerable packages.
 - 13 ▶ Licenses
 - 22 ▶ Dependency Changes
 - 295
 - 296
 - > ✅ Post Checkout Repository 0s
 - > ✅ Complete job 0s

Collaboration



Security

Microsoft takes the security of our software products and services seriously, which includes all source code repositories managed through our GitHub organizations, which include [Microsoft](#), [Azure](#), [DotNet](#), [AspNet](#) and [Xamarin](#).

If you believe you have found a security vulnerability in any Microsoft-owned repository that meets [Microsoft's definition of a security vulnerability](#), please report it to us as described below.

Reporting Security Issues

Please do not report security vulnerabilities through public GitHub issues.

Instead, please report them to the Microsoft Security Response Center (MSRC) at <https://msrc.microsoft.com/create-report>.

If you prefer to submit without logging in, send email to secure@microsoft.com. If possible, encrypt your message with our PGP key; please download it from the [Microsoft Security Response Center PGP Key page](#).

You should receive a response within 24 hours. If for some reason you do not, please follow up via email to ensure we received your original message. Additional information can be found at microsoft.com/msrc.

Please include the requested information listed below (as much as you can provide) to help us better understand the nature and scope of the possible issue:

- Type of issue (e.g. buffer overflow, SQL injection, cross-site scripting, etc.)
- Full paths of source file(s) related to the manifestation of the issue
- The location of the affected source code (tag/branch/commit or direct URL)
- Any special configuration required to reproduce the issue
- Step-by-step instructions to reproduce the issue
- Proof-of-concept or exploit code (if possible)
- Impact of the issue, including how an attacker might exploit the issue

This information will help us triage your report more quickly.

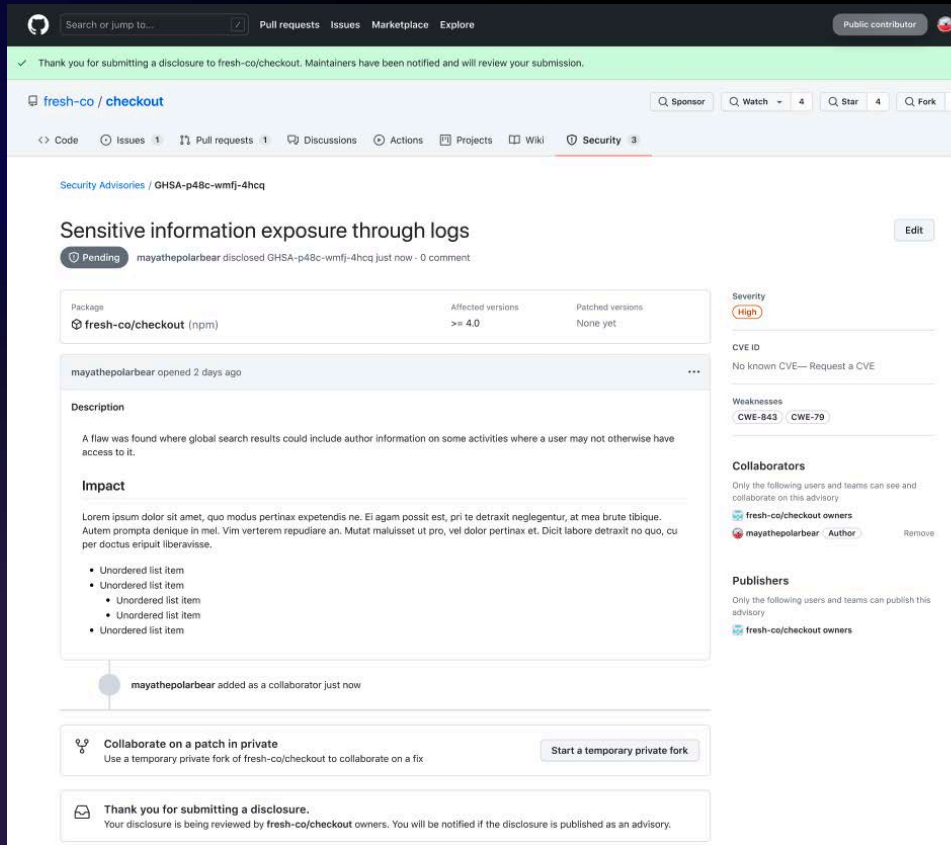
If you are reporting for a bug bounty, more complete reports can contribute to a higher bounty award. Please visit our [Microsoft Bug Bounty Program](#) page for more details about our active programs.

Preferred Languages

We prefer all communications to be in English.

Policy

Microsoft follows the principle of [Coordinated Vulnerability Disclosure](#).



Thank you for submitting a disclosure to fresh-co/checkout. Maintainers have been notified and will review your submission.

fresh-co / checkout

Security Advisories / GHSA-p48c-wmfj-4hcq

Sensitive information exposure through logs

Pending mayathepolarbear disclosed GHSA-p48c-wmfj-4hcq just now · 0 comment

Package	Affected versions	Patched versions
fresh-co/checkout (npm)	>= 4.0	None yet

mayathepolarbear opened 2 days ago

Description

A flaw was found where global search results could include author information on some activities where a user may not otherwise have access to it.

Impact

Lorem ipsum dolor sit amet, quo modus pertinax expetendis ne. Ei agam possit est, pri te detrahit neglegentur, at mea brute tibi que. Autem prompta denique in mel. Vim verterem repudiare an. Mutat malisset ut pro, vel dolor pertinax et. Dicit labore detrahit no quo, cu per doctus eriput liberavisse.

- Unordered list item
- Unordered list item
 - Unordered list item
 - Unordered list item
- Unordered list item

mayathepolarbear added as a collaborator just now

Collaborate on a patch in private
Use a temporary private fork of fresh-co/checkout to collaborate on a fix [Start a temporary private fork](#)

Thank you for submitting a disclosure.
Your disclosure is being reviewed by fresh-co/checkout owners. You will be notified if the disclosure is published as an advisory.

Severity
High

CVE ID
No known CVE— Request a CVE

Weaknesses
CWE-843 CWE-79

Collaborators
Only the following users and teams can see and collaborate on this advisory.
fresh-co/checkout owners
mayathepolarbear Author Remove

Publishers
Only the following users and teams can publish this advisory.
fresh-co/checkout owners



Productivity



github-actions bot commented on Mar 28, 2022



Heads up [@jhutchings1](#) [@erinhav](#) [@exvuma](#) - the "dependabot" label was applied to this issue.



[greysteil](#) added the [security advisory database](#) label on Mar 28, 2022



github-actions bot commented on Mar 28, 2022



Heads up [@jhutchings1](#) [@KateCatlin](#) - the "security-advisory-database" label was applied to this issue.



[github-actions](#) bot added the [requested-1-5](#) label on Mar 28, 2022





[exvuma](#) added this to **Dependabot GHES 3.3 Private Beta** on Apr 6, 2022


Multi-checkout: add discount label, switch summary ordering, add purchase hydro event #294365

Edit

<> Code ▾

 Open gwwar wants to merge 4 commits into `master` from `multicheckout-feedback` 

 Conversation 5

 Commits 4

 Checks 154

 Files changed 3

+132 -63 



gwwar commented 4 hours ago · edited ▾

Reviewers: Please read carefully. By approving, you support the deployment and mitigation plans as well as the code change. If anything is unclear or missing, please ask for updates.

Context

Part of [github/octogrowth#2779](#)

What are you trying to accomplish?



Changes here:

Reviewers



github/gitcoin-reviewers is a code owner ✓



gitcoin-reviewers  

Still in progress? [Convert to draft](#)

Assignees



gwwar

Labels



Powered by AI

Remediation



Faster Remediation

Quickly re-write vulnerable code with suggested fixes



Context Aware Suggestions

Developers gain a deeper understanding of findings from security tools

GITHUB COPILOT

lefrightleft

CodeQL Identified a SQL injection in this file. Can you help me create a fix for this issue?

GitHub Copilot

Certainly! One way to fix this issue is to use parameterized queries instead of concatenating strings. Here's an example of how you can modify the code to use parameterized queries:

```
author = request.args.get('author')
read = bool(request.args.get('read'))

if name:
    cursor.execute(
        "SELECT * FROM books WHERE name LIKE %s",
        ('%' + name + '%',)
    )
    books = [Book(*row) for row in cursor]

elif author:
    cursor.execute(
        "SELECT * FROM books WHERE author LIKE %s",
        ('%' + author + '%',)
    )
    books = [Book(*row) for row in cursor]

else:
    cursor.execute("SELECT name, author, read FROM books")
    books = [Book(*row) for row in cursor]

return render_template('books.html', books=books)
```

In this modified code, we use %s as a placeholder for the search term, and pass the search term as a tuple as the second argument to the execute method. This way, the search term is properly escaped and prevents SQL injection attacks.

How can I test that my fix for the SQL injection issue is working correctly?

Ask Copilot a question or type '/' for topics

routes.py — advanced-security-pytho

```
routes.py
from server.models import Book

@flaskapp.route('/')
def index():
    name = request.args.get('name')
    author = request.args.get('author')
    read = bool(request.args.get('read'))

    if name:
        cursor.execute(
            "SELECT * FROM books WHERE name LIKE '%s'",
            ('%' + name + '%',)
        )
        books = [Book(*row) for row in cursor]

    elif author:
        cursor.execute(
            "SELECT * FROM books WHERE author LIKE '%s'",
            ('%' + author + '%',)
        )
        books = [Book(*row) for row in cursor]

    else:
        cursor.execute("SELECT name, author, read FROM books")
        books = [Book(*row) for row in cursor]

    return render_template('books.html', books=books)
```

Security Alert

This SQL query depends on a user-provided value.

SQL query built from user-cont

If a database query (such as a SQL or NoSQL query) is built from user-provided input, it can be used to run malicious database queries.

This also includes using the TextClause class in the [SQLAlchemy] which is used to represent a literal SQL fragment and is inserted into the query.

Recommendation

Most database connector libraries offer a way of safely embedding user-provided input into prepared statements.

Example

Ln 22, C



Powered by AI

Streamline Incident Response

Leverage Copilot to **build** Custom CodeQL Queries, simplifying program analysis for Security Researchers

```
quick-query.ql x
Quick Queries > quick-query.ql > {} quick-query
1  import java
2
3  // Find all classes that extend java.util.List
4
5  select ""
```





Powered by AI

Prevention

GitHub Copilot enables developers to reach that **magical flow state**—including delivering fast, accurate vulnerability prevention right in the editor. This, combined with GitHub Advanced Security provides an end-to-end **seamless experience** for developers to secure their code.



AI-based vulnerability prevention

Blocks insecure coding patterns in real-time to make GitHub Copilot suggestions more secure.



Secure suggestions

Insecure coding patterns are quickly blocked and replaced by alternative suggestions.



Incredibly fast

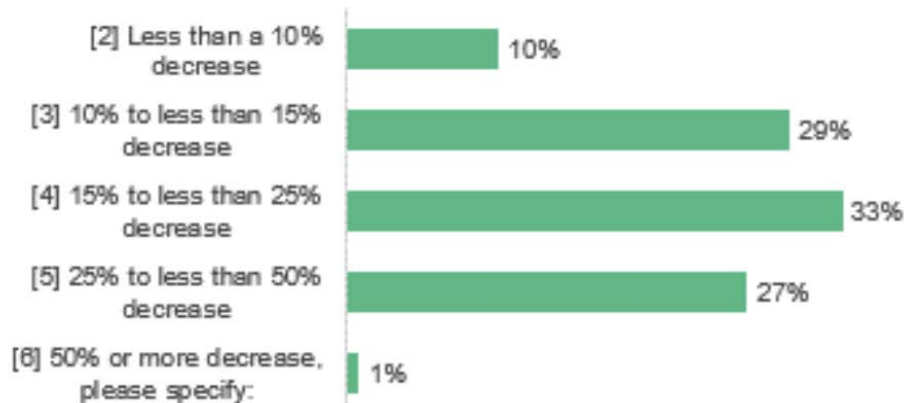
GitHub Copilot runs advanced AI models on powerful compute resources



Data

Proving the Solution

“By what percentage did GitHub decrease the number of security flaws for your organization?”



Base: 109 – GitHub Enterprise Cloud users who said it reduced the number of security flaws at their organization

Source: A commissioned study conducted by Forrester Consulting on behalf of GitHub, August 2022



Data

Proving the Solution

<p>3. Reduce the vulnerability backlog</p> <p>by applying productivity gains to increase remediation.</p>	<p>Some Teams report fewer vulnerabilities from one year ago while others have a stable backlog.</p>	<p>Reducing the vulnerability backlog is an indicator that code scanning and remediation efforts are translating to reduced security risk</p>	<p>Combining the two improvements from above, 414,000 hours is reduced to 110,000 hours to handle the same pace-of-change and the same number of vulnerabilities.</p> <p>For the same effort/delay experienced today by Devs, it would be possible to remediate 4 times the number of vulnerabilities!</p> <p>(Increasing the rate of remediation is the only way vulnerability backlogs can be reduced.)</p>
<p>4. Improve Predictability of PR's</p>	<p>We did not collect direct data about how vulnerability scanning & remediation affect the PR-Merge process, but teams are merging PR's on a daily basis, so even an hour or two of impact from scanning & remediation is significant.</p>	<p>In Microsoft's experience, Efficient, Predictable PR's encourage Collaboration, sharing, and cross-team contributions.</p>	<p>Faster remediation leads to faster, more predictable PR merges:</p> <ul style="list-style-type: none">- Scanning leads to fewer surprises, less delay- Reduced backlog means contributors and consumers take on less risk.



Data

Proving the Solution

<p>5. Lower the effort required to collaborate and spur “Innersourcing”</p>	<p>Uncertainty and Inefficiency slows collaboration and undermines the goals of finding Innovations and Efficiencies.</p>	<p>Easy, Simple, Predictable Pull Requests are proven precursors to Innersourcing collaboration metrics showing contributions from non-maintainers.</p>	<p>Increase in % total PRs originating from non-maintainers</p> <p>Increase % merged PRs originating from non-maintainers</p> <p>Reduction in redundant functionality and redundant code through more sharing and collaboration.</p>
------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



Summary



Communicate

- Ask to be included
- Be curious
- Be social
- Have a location to communicate



Coordinate

- Discuss when things may happen



Collaborate

- Work together and find a way
- Work in a common space



Thank you