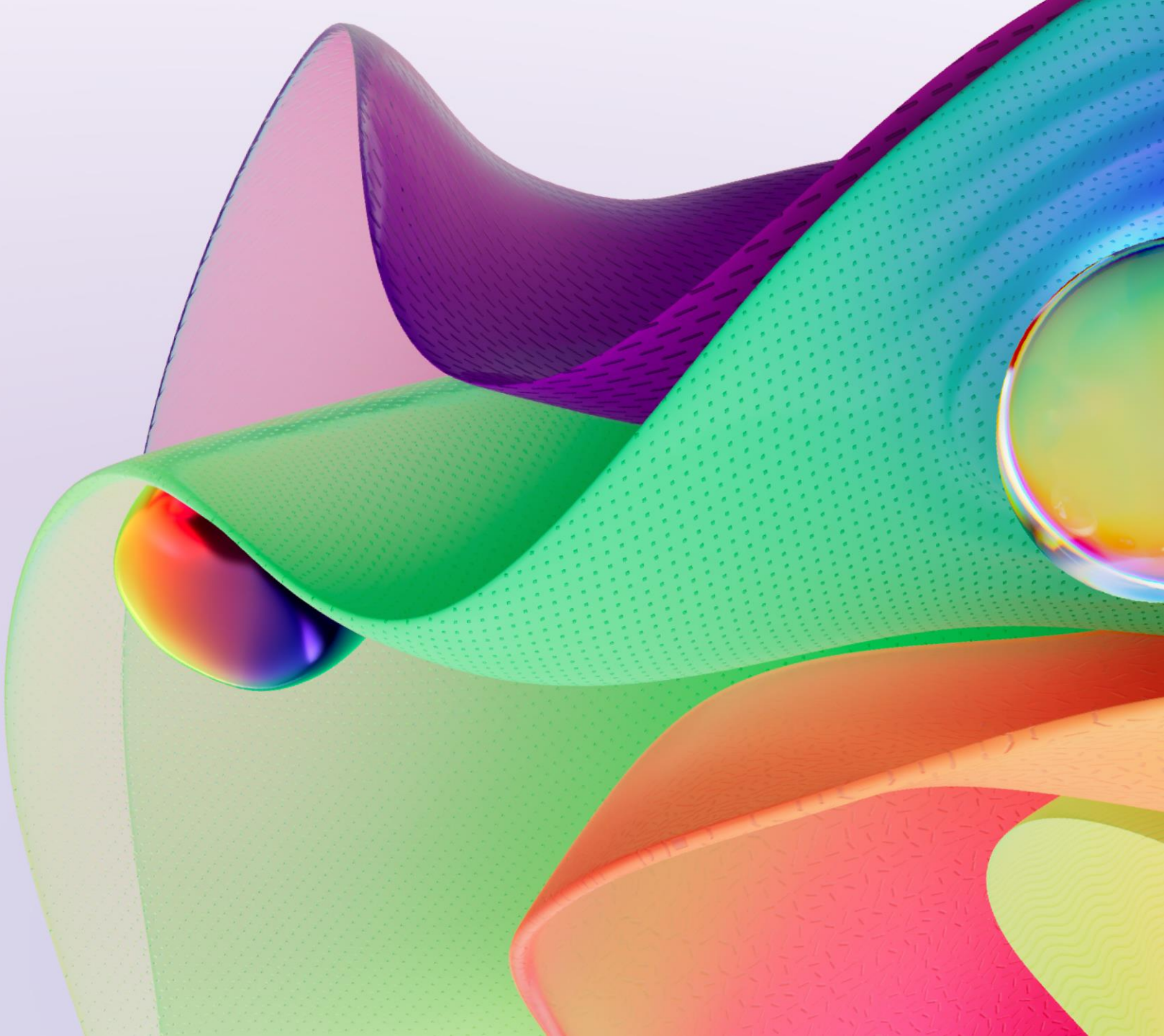
Microsoft

Microsoft AI Tour

In partnership with NVIDIA.

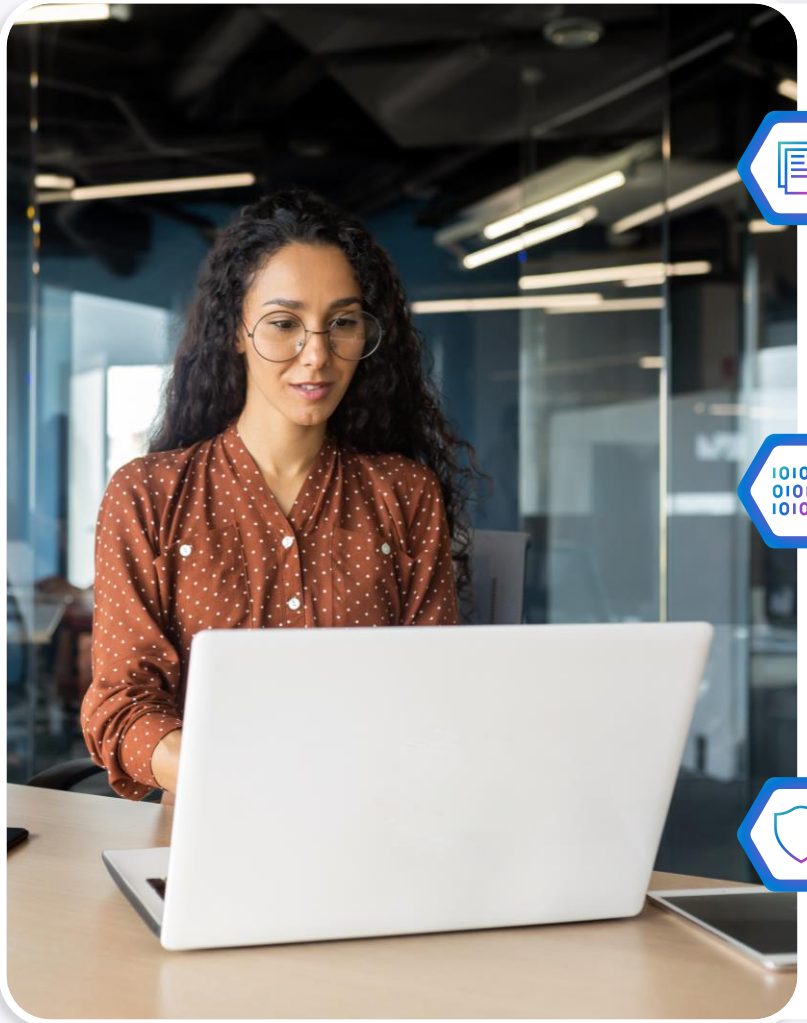


Microsoft

Future of Security with AI

Rod Trent
Senior Program Manager

Agenda



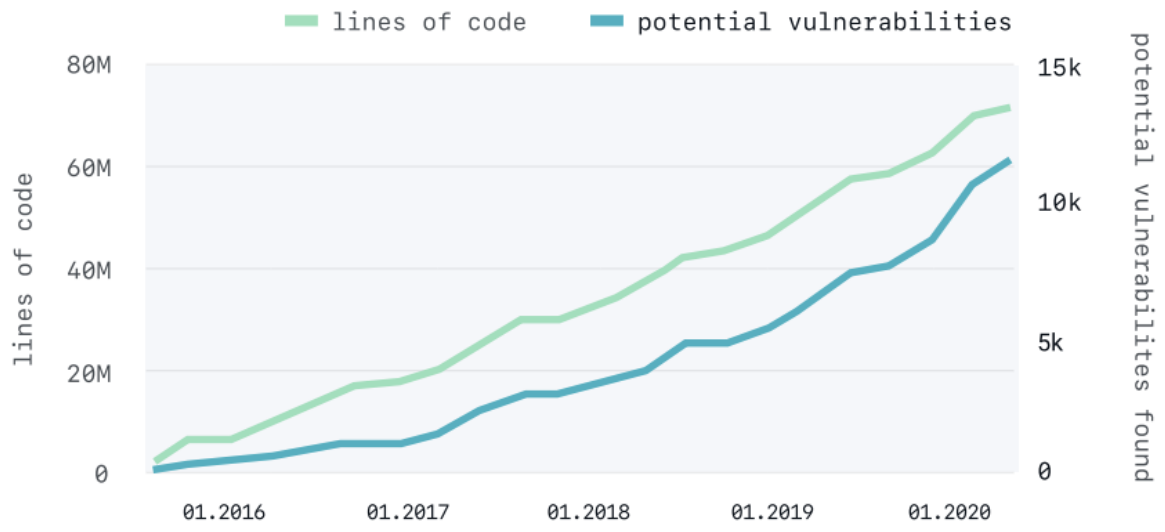
Introduction

Using GitHub Advanced Security to develop code securely

Using Security Copilot to democratize 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

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

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

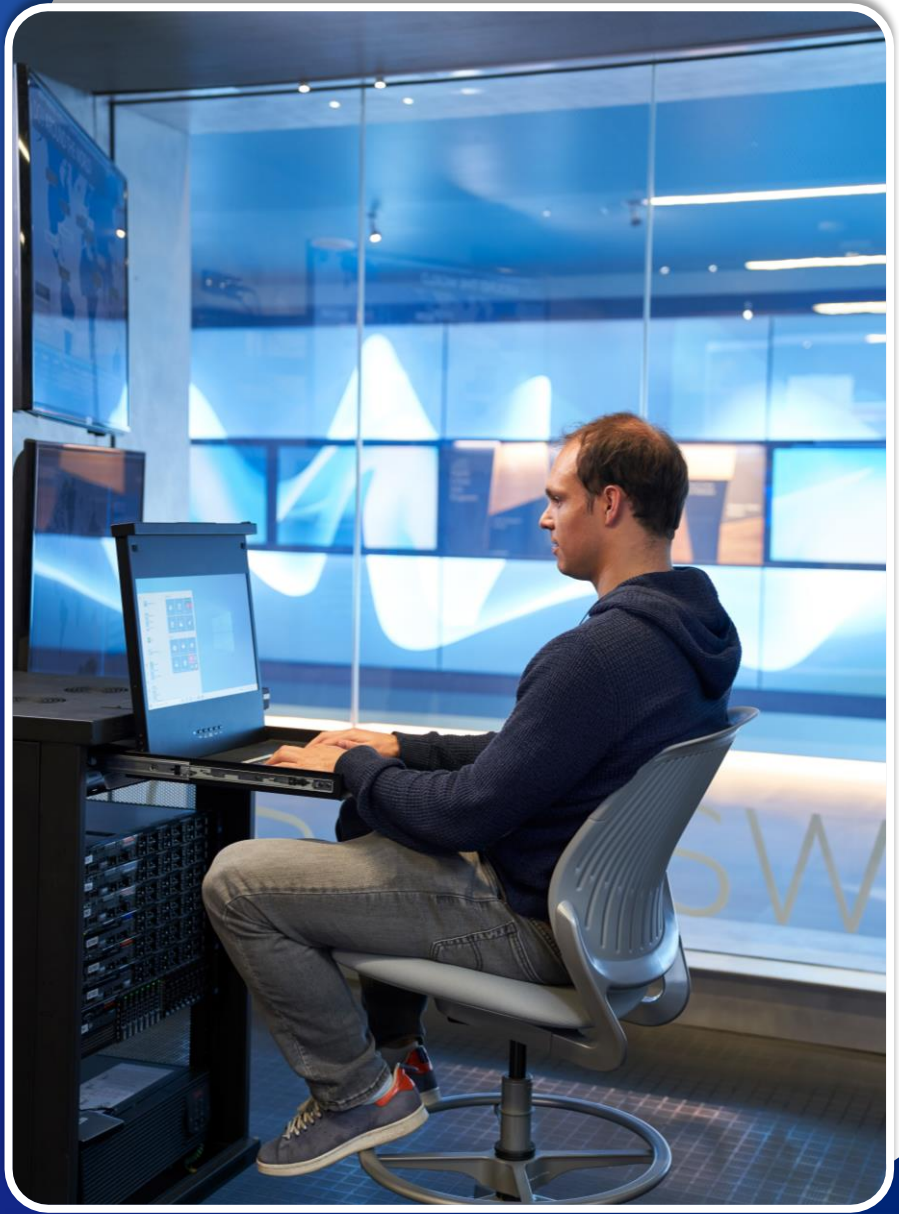
The odds are against today's security analysts

4,000 Password attacks per second

72 mins Median time for an attacker to access your private data if you fall victim to a phishing email

3.5M Global shortage of skilled cybersecurity professionals

Using GitHub Advanced Security to develop code securely



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

Development lifecycle

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

Commit signing
Enforce requirement that all commits are signed

Supply Chain

Dependabot

Automatically update vulnerable and out-of-date dependencies

Automated pull requests for security & version updates

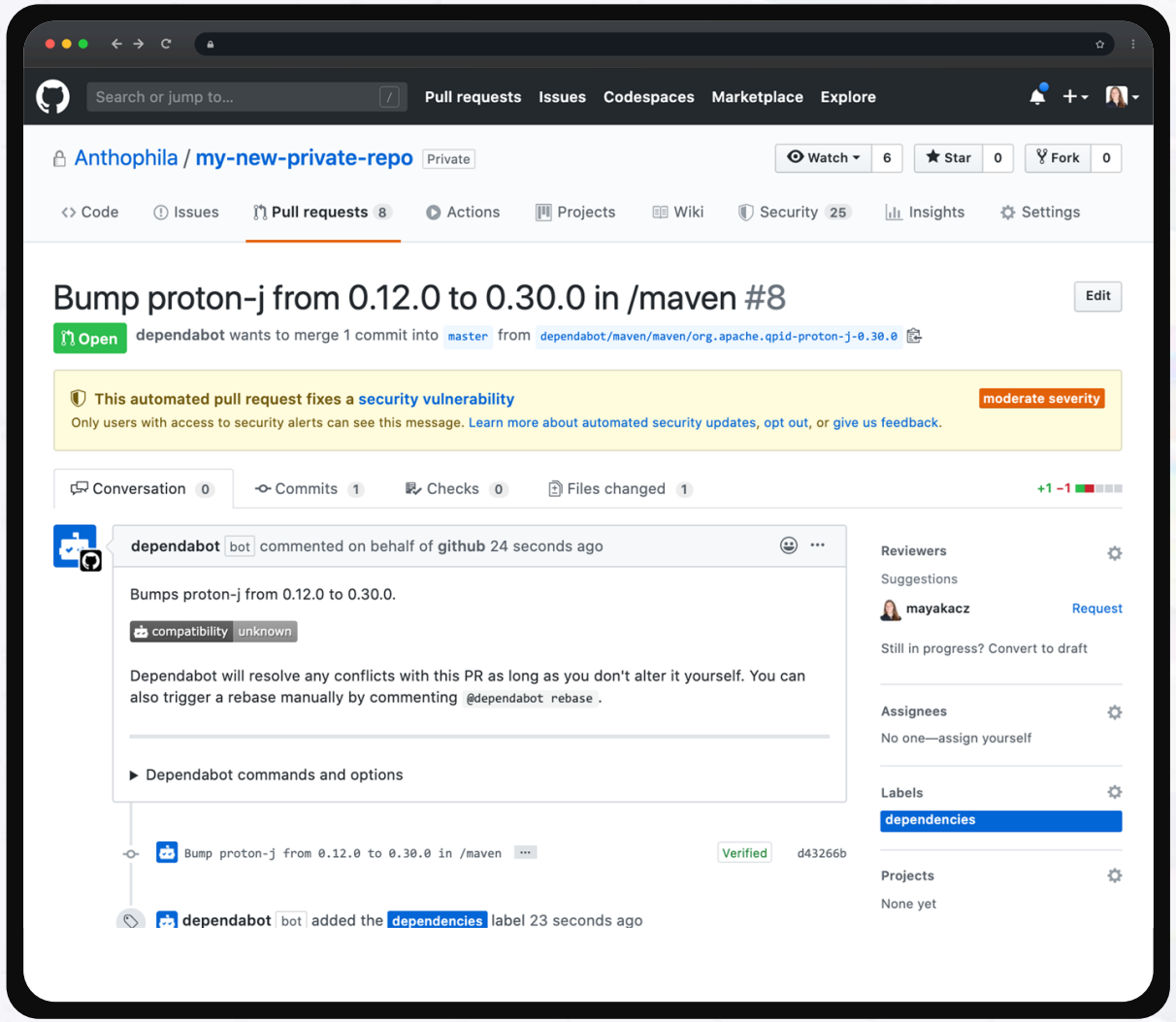
Keep your projects secure and up to date by monitoring them for vulnerable and out-of-date components. If a suggested update is found, we'll automatically open a pull request with suggested fixes.

Integrated with developer workflow

Dependabot is integrated directly into the developer workflow for a frictionless experience and faster fixes.

Rich vulnerability data

GitHub tracks vulnerabilities in packages from supported package managers using data from security researchers, maintainers, and the National Vulnerability Database—all discoverable in the GitHub Advisory Database.



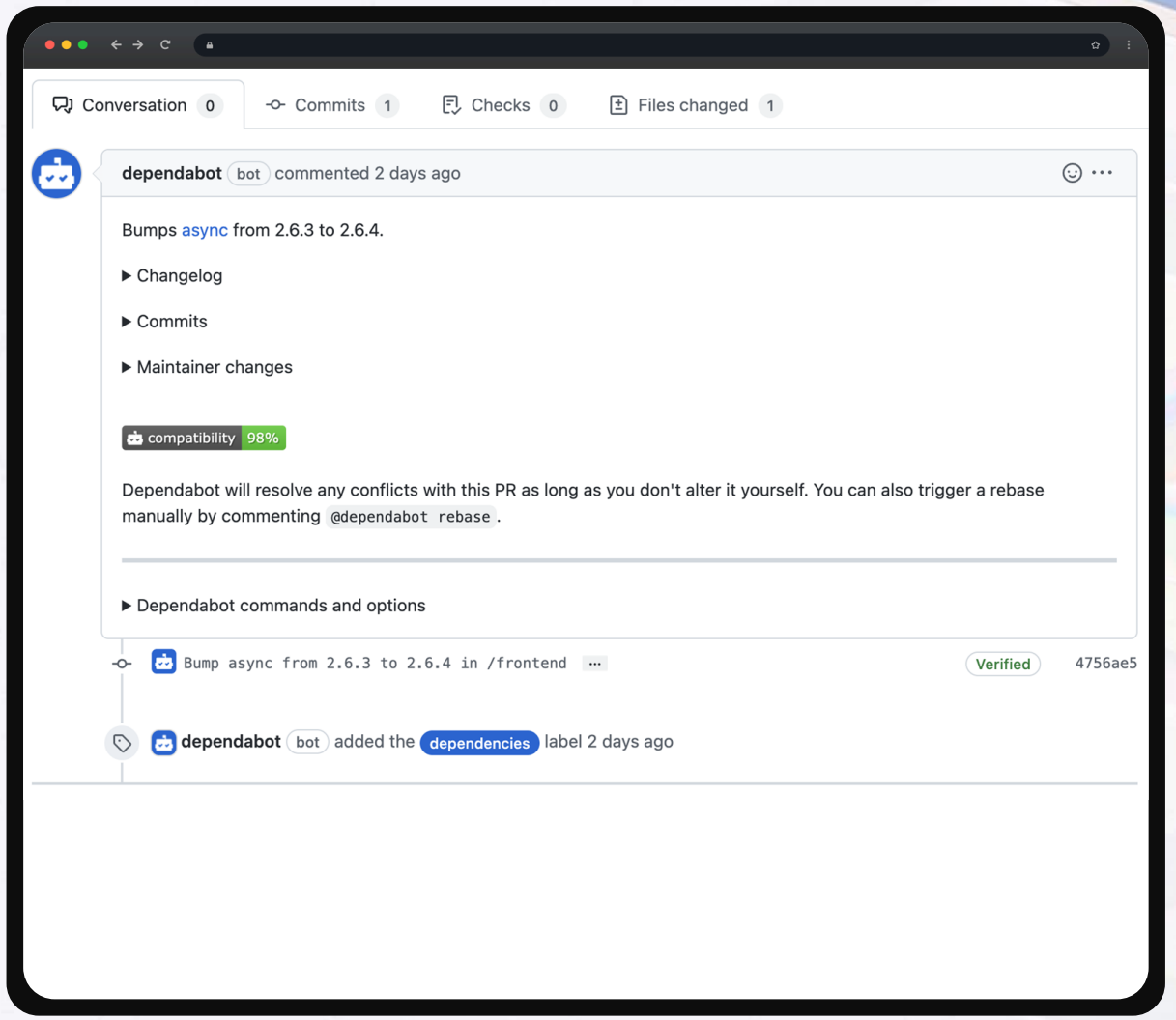
Dependabot

Automatically raise alerts when vulnerable dependencies are detected.

Automatically open pull requests to fix dependency vulnerabilities.

Notify the appropriate people about the vulnerability.

Rate the compatibility of a vulnerability patch.



Secret Scanning

Secret scanning

Find and manage hard-coded secrets

Identifies secrets as early as possible

Finds secrets (including Azure secrets) the moment they are pushed to GitHub and immediately notifies developers when they are found.

Community of secret scanning partners

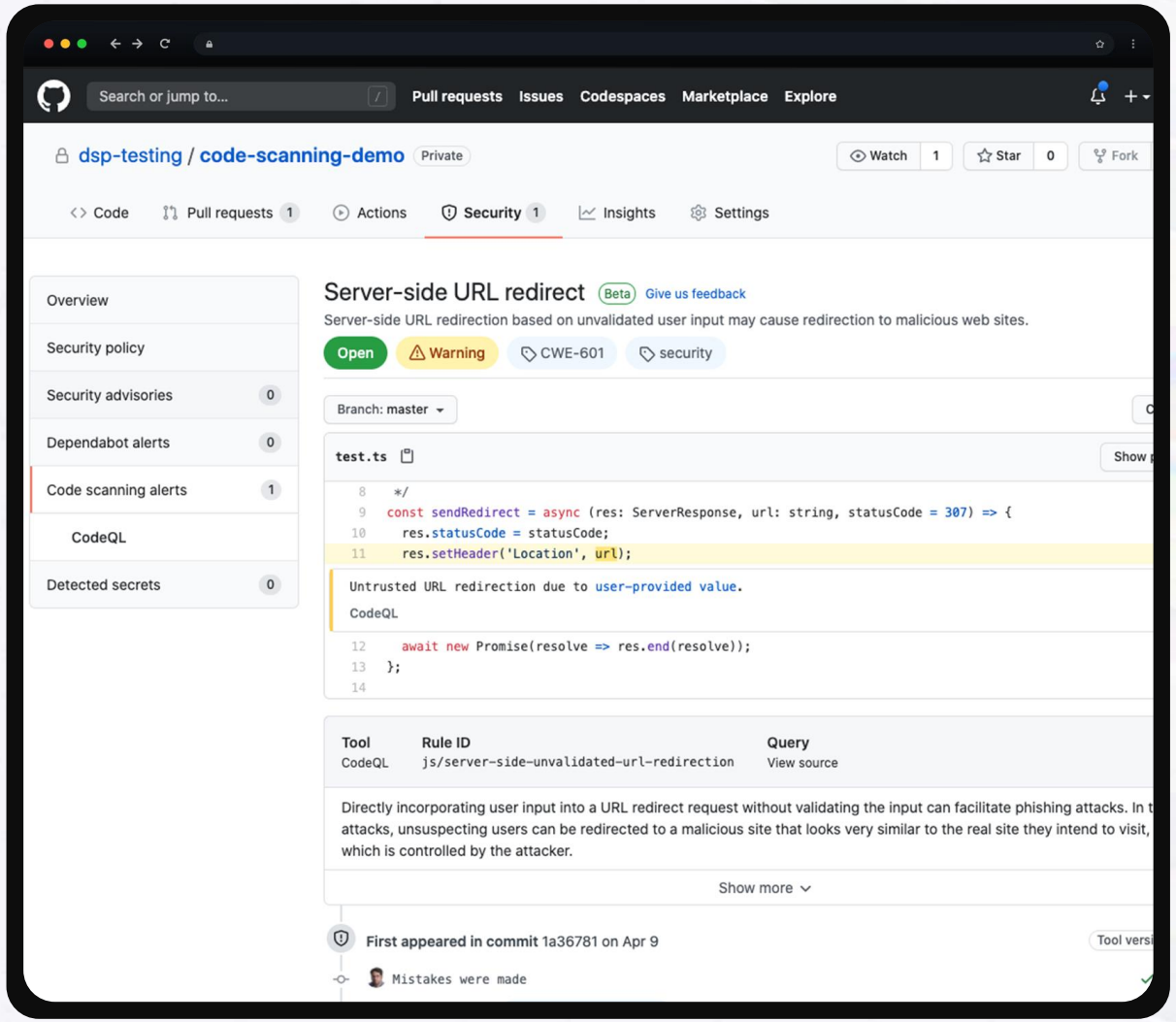
For every commit made to your repository and its full git history, we'll look for secret formats from secret scanning partners.

Define custom patterns

Scan for patterns that are internal to your organization across your repositories.

Supports both public and private repos

Secret scanning watches both public and private repos for potential secret vulnerabilities.



octodemo / leftrightleft-beat-bot

Type / to search

Code | Issues | Pull requests 1 | Actions | Projects | Wiki | Security 6 | Insights | Settings

leftrightleft-beat-bot

Internal

generated from octodemo/beat-bot

Edit Pins | Watch 6 | Fork 0 | Star 0

main | 2 Branches | 0 Tags | Go to file | Add file | Code

leftrightleft Initial commit

b74757d · 1 minute ago | 1 Commits

.github/workflows	Initial commit	1 minute ago
front-end	Initial commit	1 minute ago
src	Initial commit	1 minute ago
test	Initial commit	1 minute ago
.env	Initial commit	1 minute ago
.gitignore	Initial commit	1 minute ago
Dockerfile	Initial commit	1 minute ago
README.md	Initial commit	1 minute ago
entrypoint.sh	Initial commit	1 minute ago
requirements.txt	Initial commit	1 minute ago

README

About

No description, website, or topics provided.

Readme

Activity

0 stars

6 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

Languages

Python 96.7% | Dockerfile 1.9% | Other 1.4%

Code Scanning

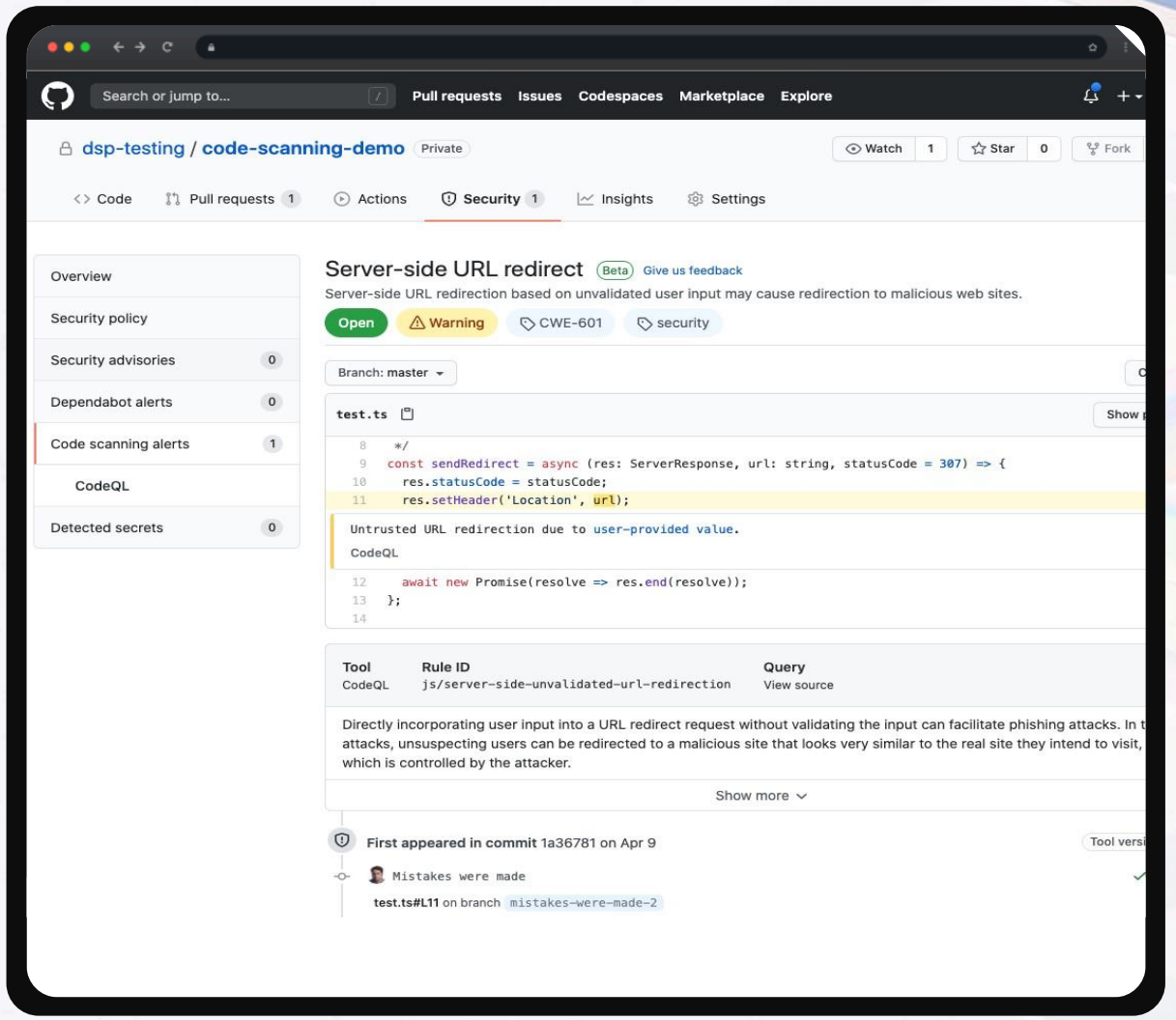
Code Scanning

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.



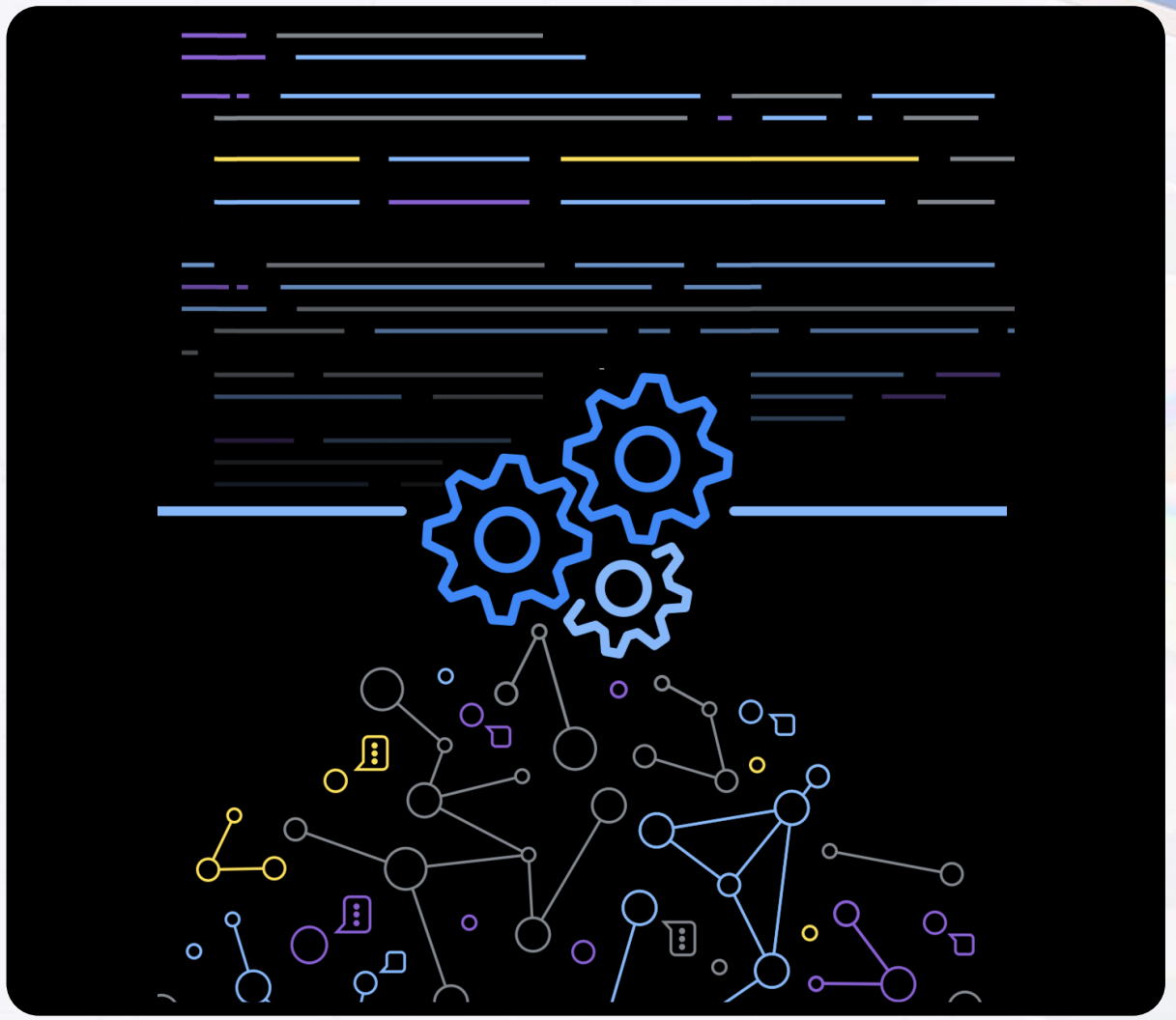
CodeQL: A revolutionary semantic code engine

Advanced code analysis engine based on 13 years of research by a 30-person team from Oxford University.

Allows you to query your code's logic to find vulnerabilities.

Queries can be quickly customized to adapt to your specific threat topology.

Community-driven query set powers every project with a world-class security team.



Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks. Learn more about diff comparisons here.

base: main ← compare: new-api-endpoints ✓ **Able to merge.** These branches can be automatically merged.

Add a title

New api endpoints

Add a description

Write | Preview

Add your description here...

Markdown is supported | Paste, drop, or click to add files

Create pull request

Remember, contributions to this repository should follow our GitHub Community Guidelines.

Reviewers

No reviews

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

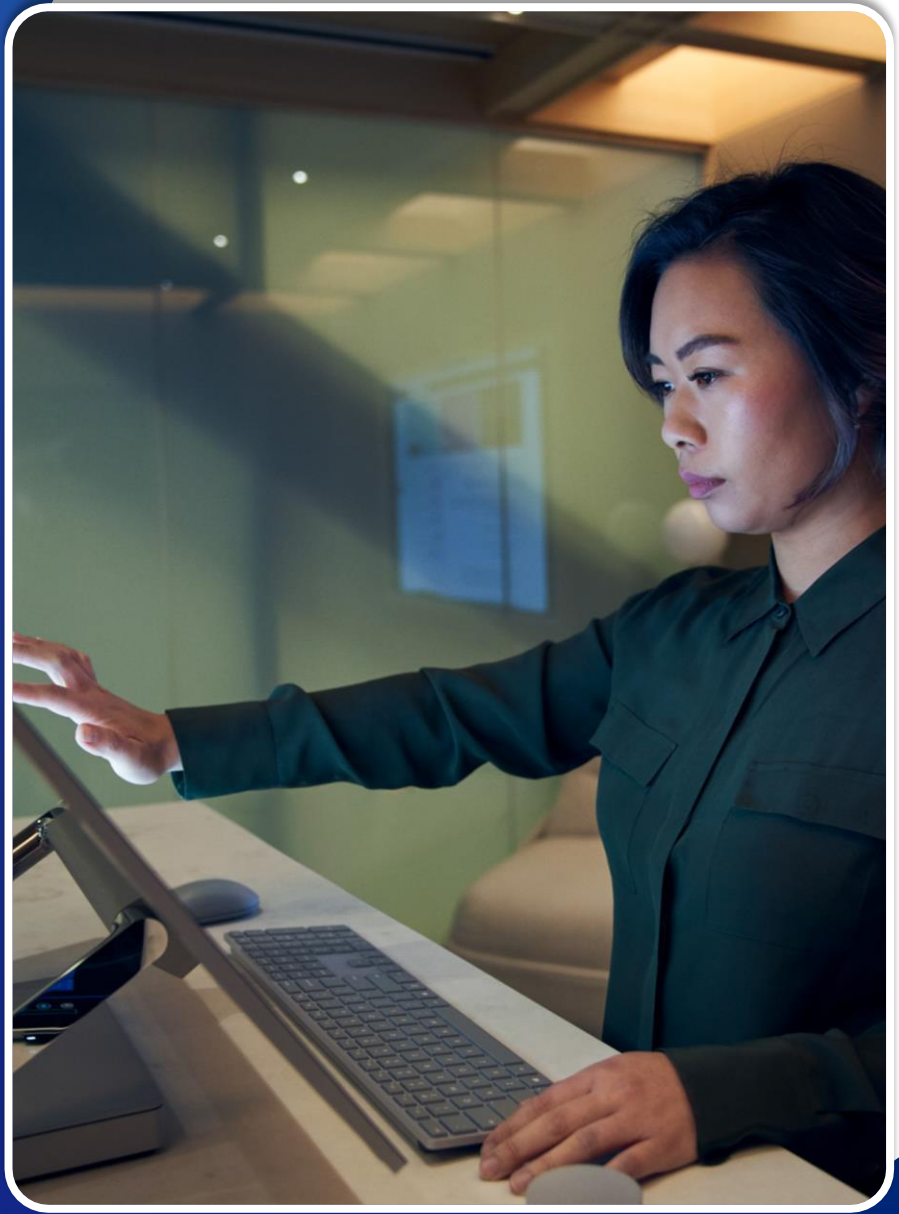
Use Closing keywords in the description to automatically close issues

Helpful resources

GitHub Community Guidelines

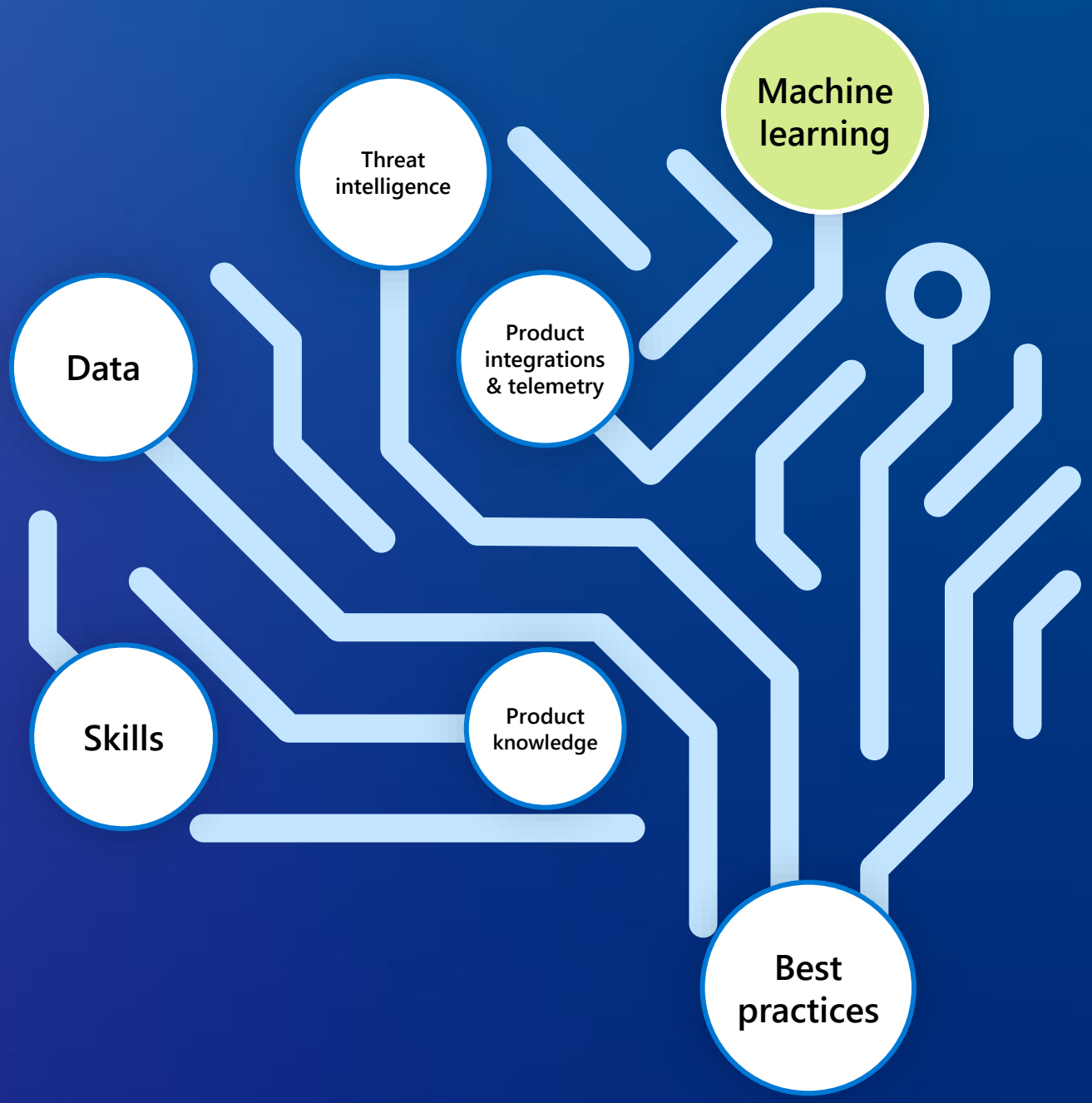
3 commits | 2 files changed | 1 contributor

Using Security Copilot to democratize security for developers

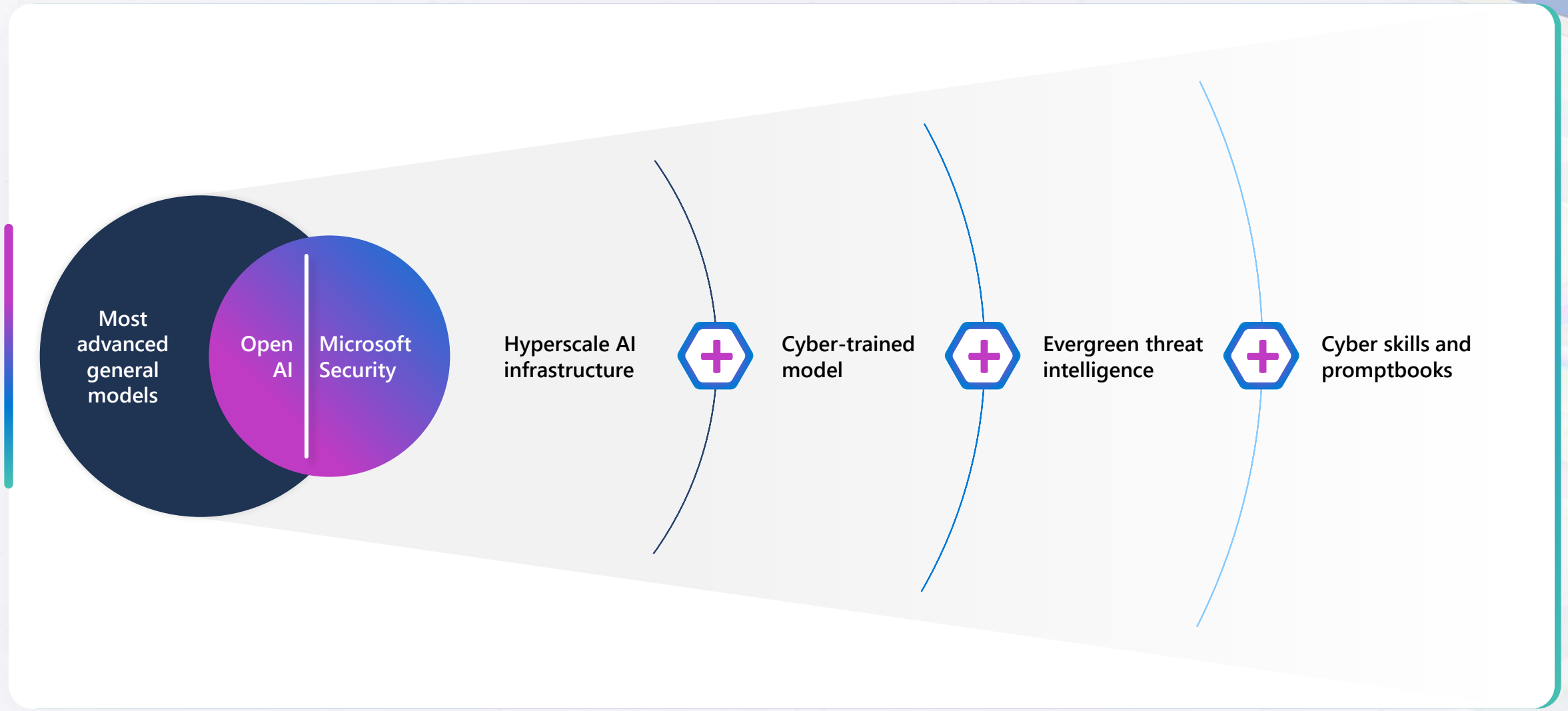


Microsoft Security Copilot

The first generative AI security product that empowers security and IT teams to defend at machine speed and scale



The Microsoft Security Copilot advantage

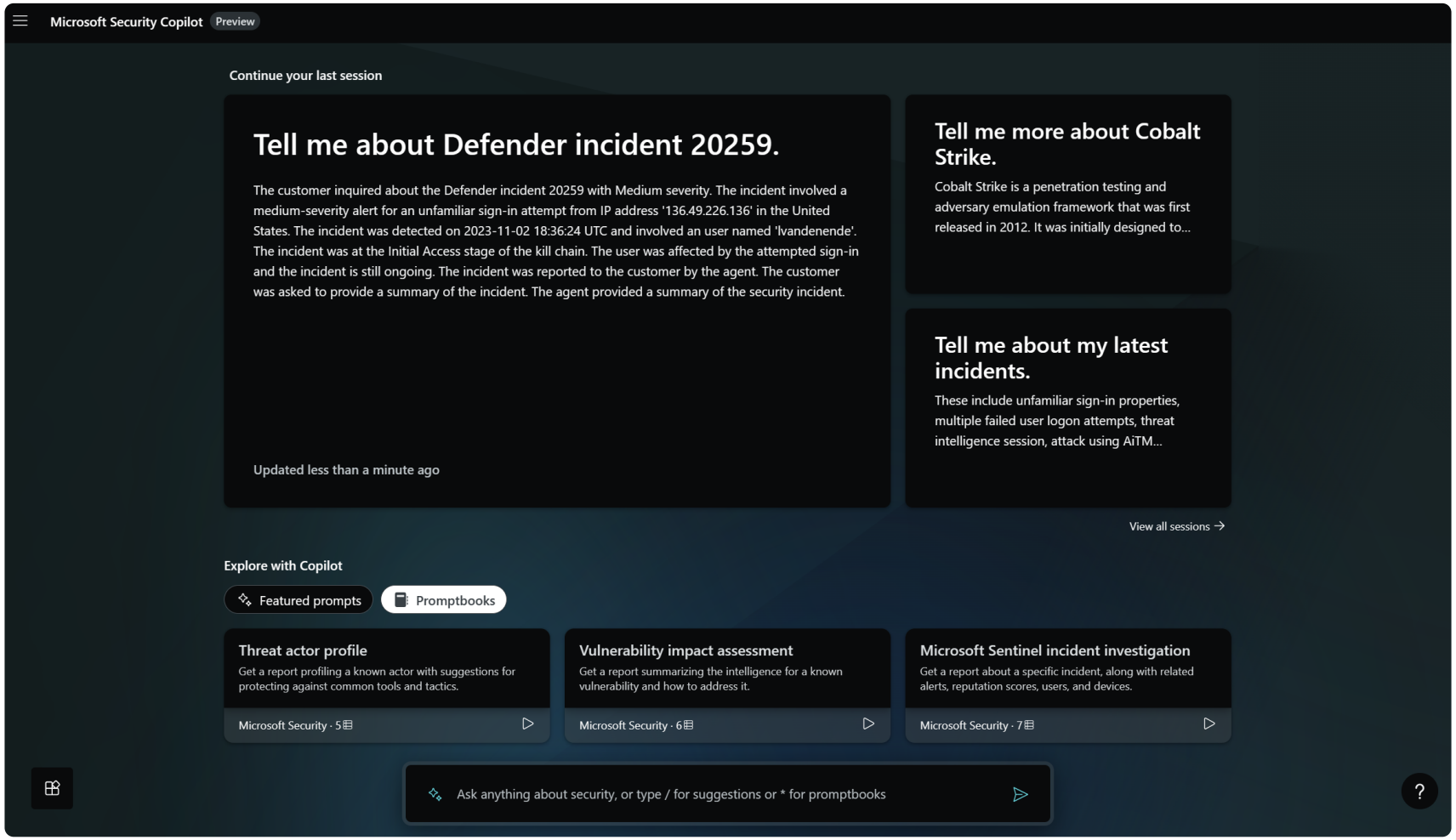


Operated with simple natural language queries

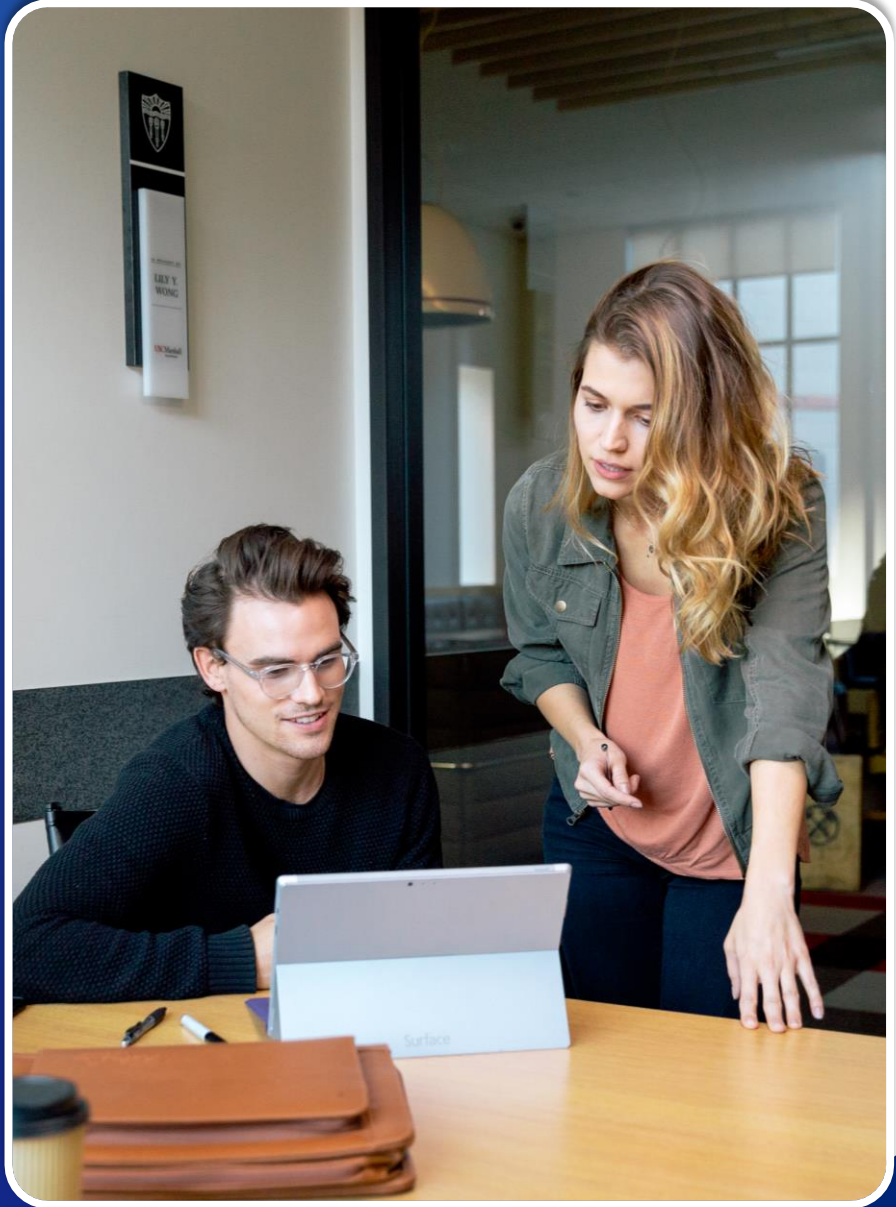
	Prompt	Planner	Build Context	Responding	Response
Human	Submits a prompt				Receives response
Security Copilot		Determines initial context and builds a plan using all the available skills	Executes the plan to get the required data context to answer the prompt	Combines all data and context and the model will work out a response	Formats the data

Microsoft Security Copilot – Video

Security Copilot standalone experience



Demo



If there are TI articles related to this CVE, provide a list and summary of them & include links.

3 steps completed 31s

Jan 4, 10:38 AM | 32.10s

Here are some Threat Intelligence articles related to CVE-2021-44228:

1. Onyx Sleet exploits vulnerable VMWare Horizon servers: In October 2022, Onyx Sleet took advantage of vulnerable VMware Horizon servers to conduct discovery and reconnaissance activity, and implemented defense evasion measures on the compromised hosts.
2. 8220 Gang Uses Log4Shell Vulnerability to Install CoinMiner: Ahnlab Security Emergency response Center (ASEC) has recently confirmed that the 8220 Gang attack group is using the Log4Shell vulnerability to install CoinMiner in VMware Horizon servers.
3. China-based Cinnamon Tempest, lone wolf turned LockBit 2.0 affiliate: The threat actor that Microsoft tracks as Cinnamon Tempest (DEV-0401) is a China-based cybercriminal group that's been active since at least July 2021.
4. Mint Sandstorm compromises targets using Log4Shell vulnerability in VMware Horizon: Mint Sandstorm (PHOSPHORUS), an activity group assessed to be working for the government of Iran, exploited Log4Shell in unpatched, internet-facing instances of VMware Horizon.
5. RiskIQ: Continued CharmPower Exploitation of Log4j Vulnerability by Iranian State-Sponsored Actors: Iranian state-sponsored threat actors PHOSPHORUS (APT35, Charming Kitten) and Nemesis Kitten were observed attempting to exploit the Log4j remote code execution vulnerability (CVE-2021-44228), using open-source JNDI exploit kits.

Please note that the summaries provided are brief and do not cover all the details in the articles. I recommend visiting the provided links for more in-depth information on each article.

Looks right Needs improvement Inappropriate

Ask anything about security, or type / for suggestions or * for promptbooks

Wrap up

GHAS

- Shift left with AI-powered AppSec blog post: https://github.blog/2023-11-08-ai-powered-appsec/
- Waitlist for AI-Powered AppSec: https://github.com/features/preview/security\
- GHAS-Lab: https://github.com/skills/secure-code-game
- GHAS certifications: Examregistration.github.com

Security Copilot

- Microsoft Security Copilot documentation | Microsoft Learn