



# Frog Proof Security

*How to Confidently Ignore 88% of Your Critical Vulnerabilities*



**Mike Holland**

*Senior Solutions Engineer, JFrog*

**Binaries are the**  
**SINGLE SOURCE**  
**for All Your Software**



# New Developer Tools

are Constantly and Rapidly Introduced



*GitHub Copilot*



*Sourcegraph Cody*



*Windsurf*



*Amazon Q*



*Claude Code*



*JFrog fly*



*Cline*



*Gemini*



*Goose*



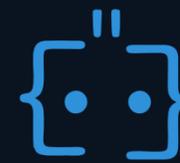
*Qodo*



*Codex*



*Devin*



*Augment*



*Cursor*

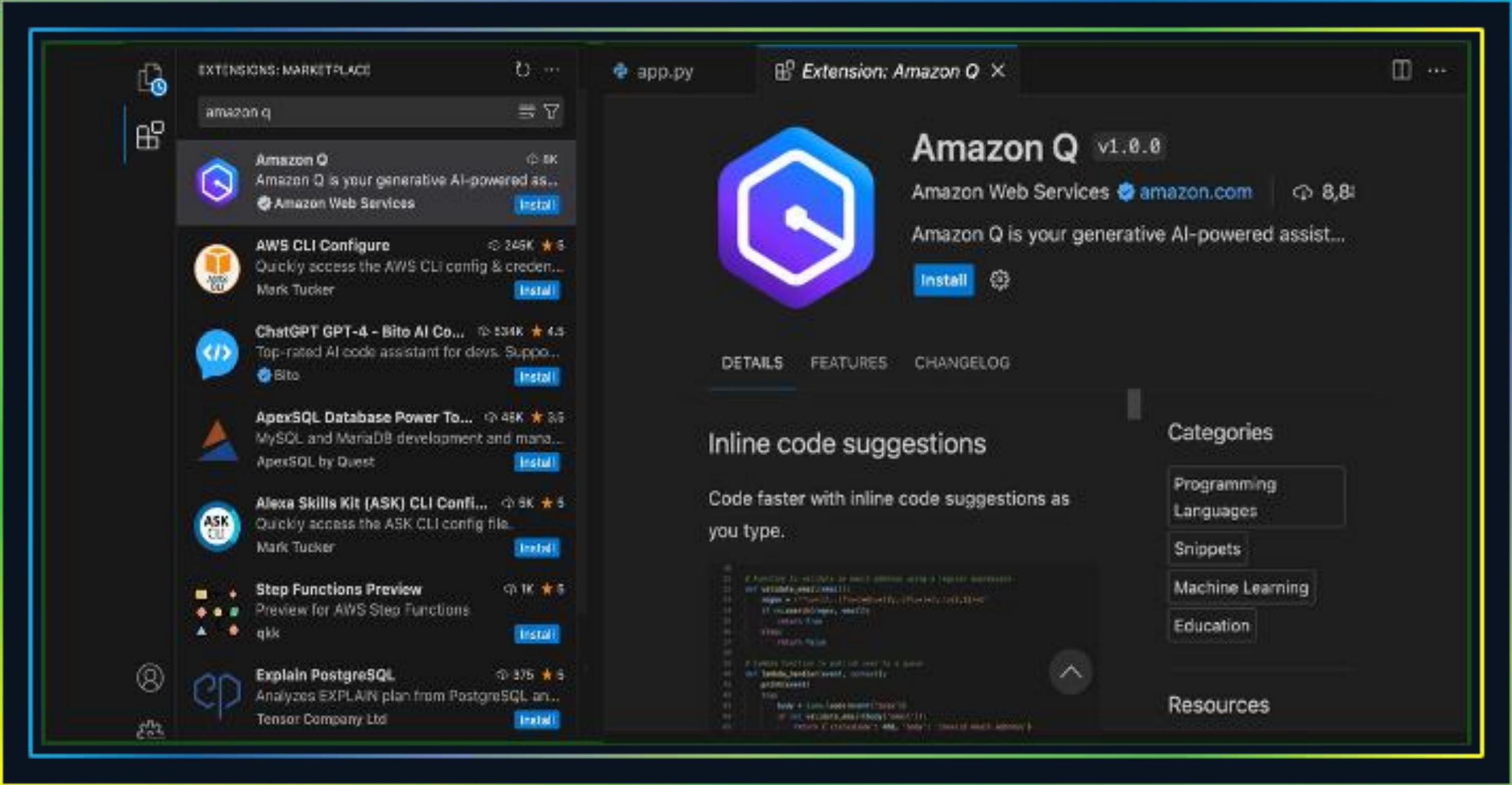


*Tabnine*

New is where  
**Threat Actors Thrive**



# Let's Look at One Attack...



# Let's Look at One Attack...

The  Register®

## Compromised Amazon Q extension told AI to delete everything – and it shipped

Malicious actor reportedly sought to expose AWS 'security theater'

 [Tim Anderson](#)

Thu 24 Jul 2025 // 14:26 UTC

The official Amazon Q extension for Visual Studio Code (VS Code) was compromised to include a prompt to wipe the user's home directory and delete all their AWS resources.

The bad extension was live on the VS Code marketplace for two days, though it appears that the intent was more to embarrass AWS and expose bad security rather than to cause immediate harm.



# How it Happened:

Attacker creates PR with malicious prompt to Amazon Q VS Code extension repository



PR accepted. New Version released

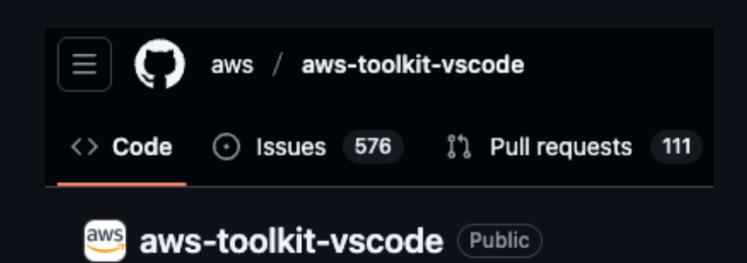


Developers obtain malicious Amazon Q VS Code extension

VS Code extension communicates with Amazon Q utilizing malicious prompt



Amazon Q returns malicious instructions to execute



# SSC Threat Entry Points



# NPM Package Hijack Overview ("Shai-Hulud" and co.)

26

Packages compromised



2M

Downloads of compromised package versions

- JFrog was first to report 5 of the compromised packages
- Other packages identified by JFrog security scanners and marked as 'malicious' within just hours
- Several JFrog customers remained seamlessly protected with Curation blocking the risk

Attackers injected malicious code to intercept and divert crypto currency transactions



# Yet another Malicious NPM Package....



SC Media  
A CISA Resource

CISO STORIES TOPICS TOPIC HUBS EVENTS PODCASTS RESEARCH SC AWARDS

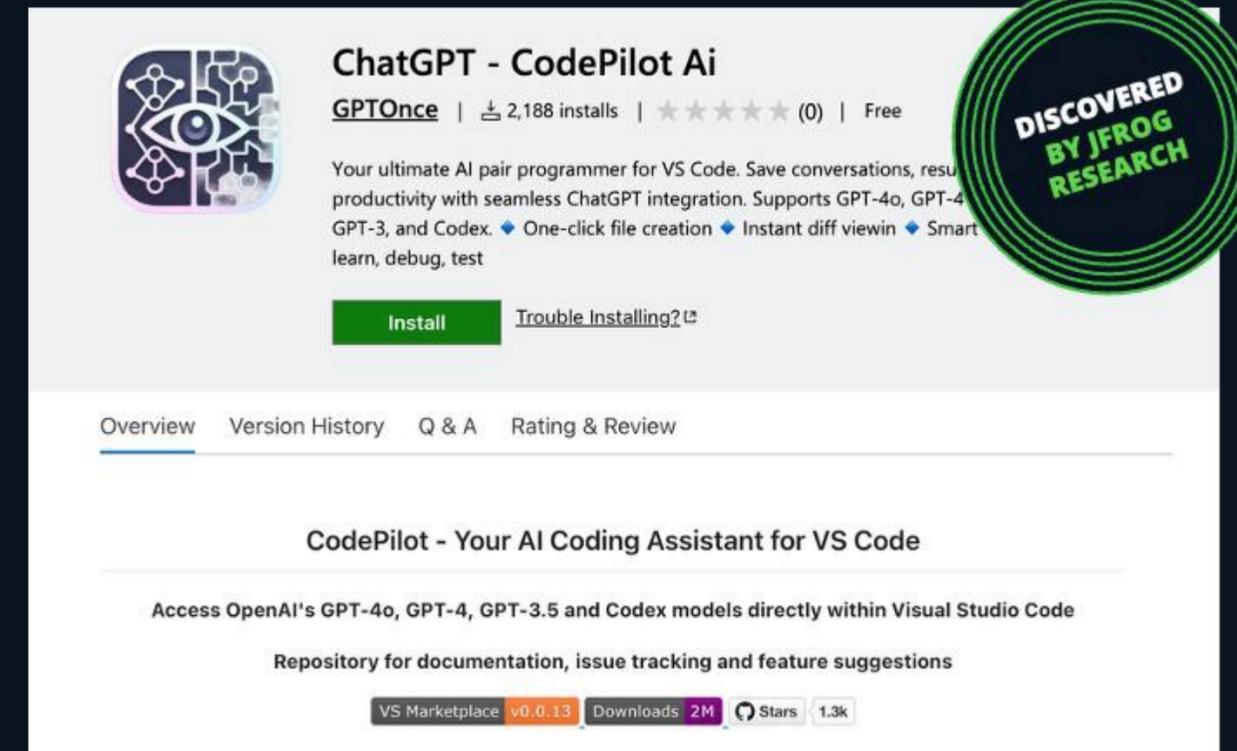
Application security, Threat Intelligence, Malware, Supply chain

## Google Calendar used as middleman for stealthy NPM malware

May 15, 2025 Share

By [Laura French](#)

The legitimate Google Calendar platform was abused to facilitate malicious command-and-control (C2) server connection in a stealthy NPM malware campaign discovered by Veracode researchers.



**ChatGPT - CodePilot Ai**  
GPTOnce | 2,188 installs | (0) | Free

Your ultimate AI pair programmer for VS Code. Save conversations, reuse snippets, and boost productivity with seamless ChatGPT integration. Supports GPT-4o, GPT-4o mini, GPT-3.5, and Codex. [One-click file creation](#) [Instant diff view](#) [Smart learn, debug, test](#)

[Install](#) [Trouble Installing?](#)

Overview Version History Q & A Rating & Review

### CodePilot - Your AI Coding Assistant for VS Code

Access OpenAI's GPT-4o, GPT-4, GPT-3.5 and Codex models directly within Visual Studio Code

Repository for documentation, issue tracking and feature suggestions

VS Marketplace v0.0.13 Downloads 2M Stars 1.3k

**DISCOVERED BY JFROG RESEARCH**

Same malware, same attacker - **Now on VSCode**



But this is not just a VSCode issue

# But this isn't just for VSCode...

The screenshot displays the 'Extensions for VS Code Compatible Editors' marketplace. At the top, there is a search bar with the text 'Search by Name, Tag, or Description' and a magnifying glass icon. To the right of the search bar is a dropdown menu labeled 'All Categories'. Below the search bar, it shows '5214 Results' and 'Sort by Relevance' with a downward arrow.

The main content area features a grid of extension cards. Each card includes an icon, the extension name, the publisher, version number, star rating, and download count. The extensions shown are:

- Language Suppo...** by redhat, version 1.45.2025081405, 5-star rating, 2.9M downloads.
- GitLens – Git su...** by somodio, version 2025.8.2109, 4.5-star rating, 3M downloads.
- rust-analyzer** by rust-lang, version 0.4.2988, 5-star rating, 1.8M downloads.
- Material Icon Th...** by FKief, version 5.26.0, 4.5-star rating, 1.2M downloads.
- YAML** by redhat, version 1.16.0, 5-star rating, 1.5M downloads.
- Continue - open-...** by Continue, version 1.1.79, 4.5-star rating, 500K downloads.
- Flutter** by Dart-Code, version 3.116.0, 5-star rating, 959K downloads.
- Dart** by Dart-Code, version 3.115.0, 5-star rating, 1M downloads.
- GitLab Workflow** by GitLab, version 0.38.1, 5-star rating, 531K downloads.
- clangd** by llvm-vs-code-extend..., version 0.2..., 5-star rating, 866K downloads.
- Cline** by claude-drewan, version 3.20.1, 4.5-star rating, 982K downloads.
- Code Spell Chec...** by streetsidesoftware, version 4.2.3, 4.5-star rating, 690K downloads.

Below the first two rows, several other extension cards are visible but partially obscured or faded, including icons for a blue cube, a red cube, a blue and yellow cube, a blue horse, a yellow warning triangle, and a blue cube with a white 'C'.

# How installing a fake extension from Open VSX led to cryptocurrency theft

This is a story of how a blockchain developer lost US\$500 000 to a fake Solidity extension from the Open VSX marketplace.



**zak.eth**   
@0xzak



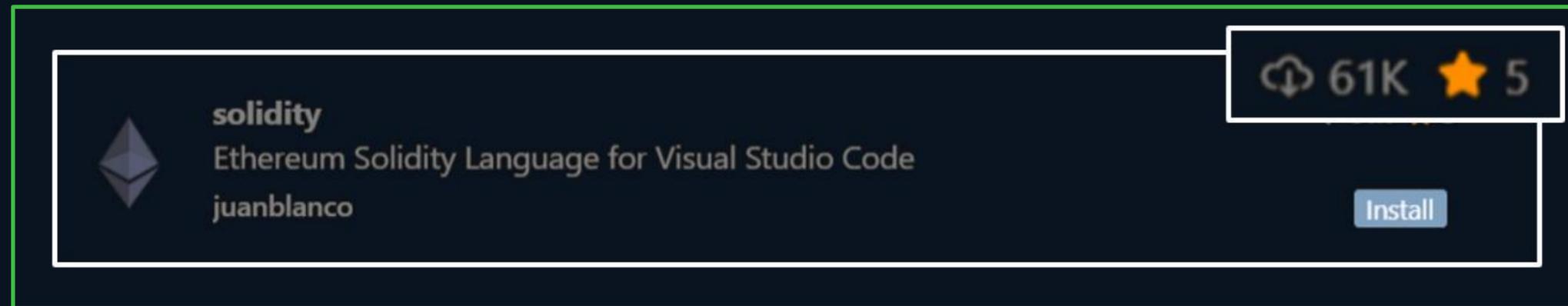
I've been in crypto for over 10 years and I've Never been hacked. Perfect OpSec record.

Yesterday, my wallet was drained by a malicious [@cursor\\_ai](#) extension for the first time.

If it can happen to me, it can happen to you. Here's a full breakdown. 



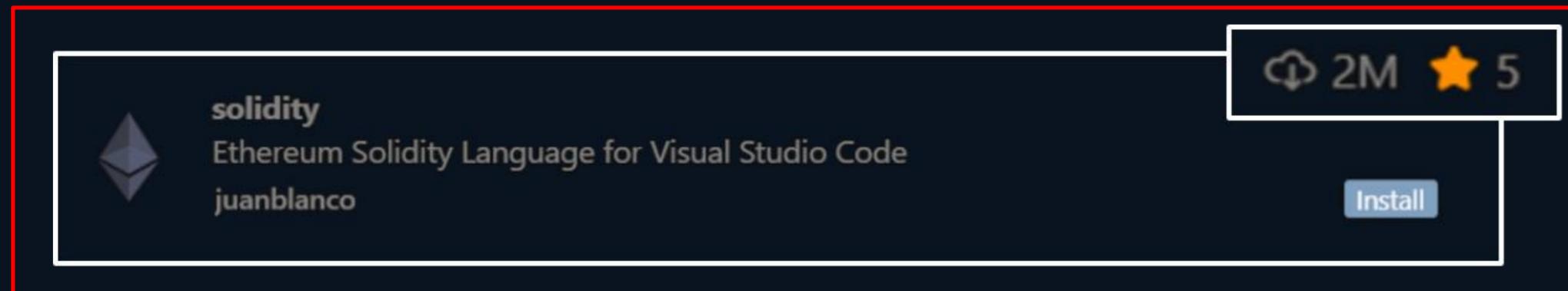
# Spot the Malicious Extension



 **solidity**  
Ethereum Solidity Language for Visual Studio Code  
juanblanco

61K ★ 5

Install



 **solidity**  
Ethereum Solidity Language for Visual Studio Code  
juanblanco

2M ★ 5

Install



# Why do SCA tools **SUCK?!**

(at detecting malicious packages)

[Why SCA Sucks \(at detecting malicious packages\)](#)



**JFrog Security** ✓  
@JFrogSecurity



**Strong piece from SourceCodeRed** [sourcecodered.com/sca-sucks/](https://sourcecodered.com/sca-sucks/)

If traditional SCA struggles to spot malicious packages, what sets JFrog's malicious-package detection apart?

Our approach combines in-house scanners, curated public databases focused specifically on malicious packages (for example, OpenSSF), and continuous monitoring of attack activity.

That proactive, layered strategy has let us identify true "zero-day" malicious packages in the wild, including:

PyPI: mcp-runcmd-server

1. [research.jfrog.com/post/3-malicio ...](https://research.jfrog.com/post/3-malicio...)

PyPI: soopsocks

1. [research.jfrog.com/post/check-you ...](https://research.jfrog.com/post/check-you...)

npm: toolkdv

1. [jfrog.com/blog/malicious ...](https://jfrog.com/blog/malicious...)

# The Solution

We Need to  
Curate the Packages used  
in + the Tools used to  
**Develop Software**

# Prevent The Next Attack With JFrog Curation

## JFrog's Security Research Team Investigated the Lifespan of Hijack Attack

- Point Developers to Artifactory
- Set policies to eliminate hijack attack risk  
Block new/immature packages (based on age)
- Make "Compliant Version Selection" effortless  
Divert package managers to mature versions for a seamless dev experience

The screenshot shows the 'New Curation Policy' configuration page in the JFrog UI. The breadcrumb trail is 'All Projects > Curation > Policies > New Curation Policy'. The main configuration area consists of five steps, each with a green checkmark icon:

- Step 1:** Policy Name: Immature Package
- Step 2:** Scope: Organization-wide
- Step 3:** Policy Condition: Package version is immature (moderate)
- Step 4:** Waivers (Optional)
- Step 5:** Actions & Notifications. Below this step, it says 'Select the required action if a violation occurs'. Two options are shown: 'Block' (selected with a green checkmark) and 'Dry run'.

On the right side, there is a 'Curation Policy Details' panel with the following information:

- Policy Name: Immature Package
- Scope: Organization-wide
- Policy Condition: Package version is immature (moderate) with an 'Operational' risk level indicator.
- Supported: npm, Maven, Gradle, Go, Docker, Helm, and others (+4).
- Description: Detects 3rd party packages whose version release date is less than 14 days old. Immature packages might impose an operational risk due to the fact that they have not yet been tested sufficiently for factors such as stability, scale and more.
- Policy Effectiveness: Covered Repositories List

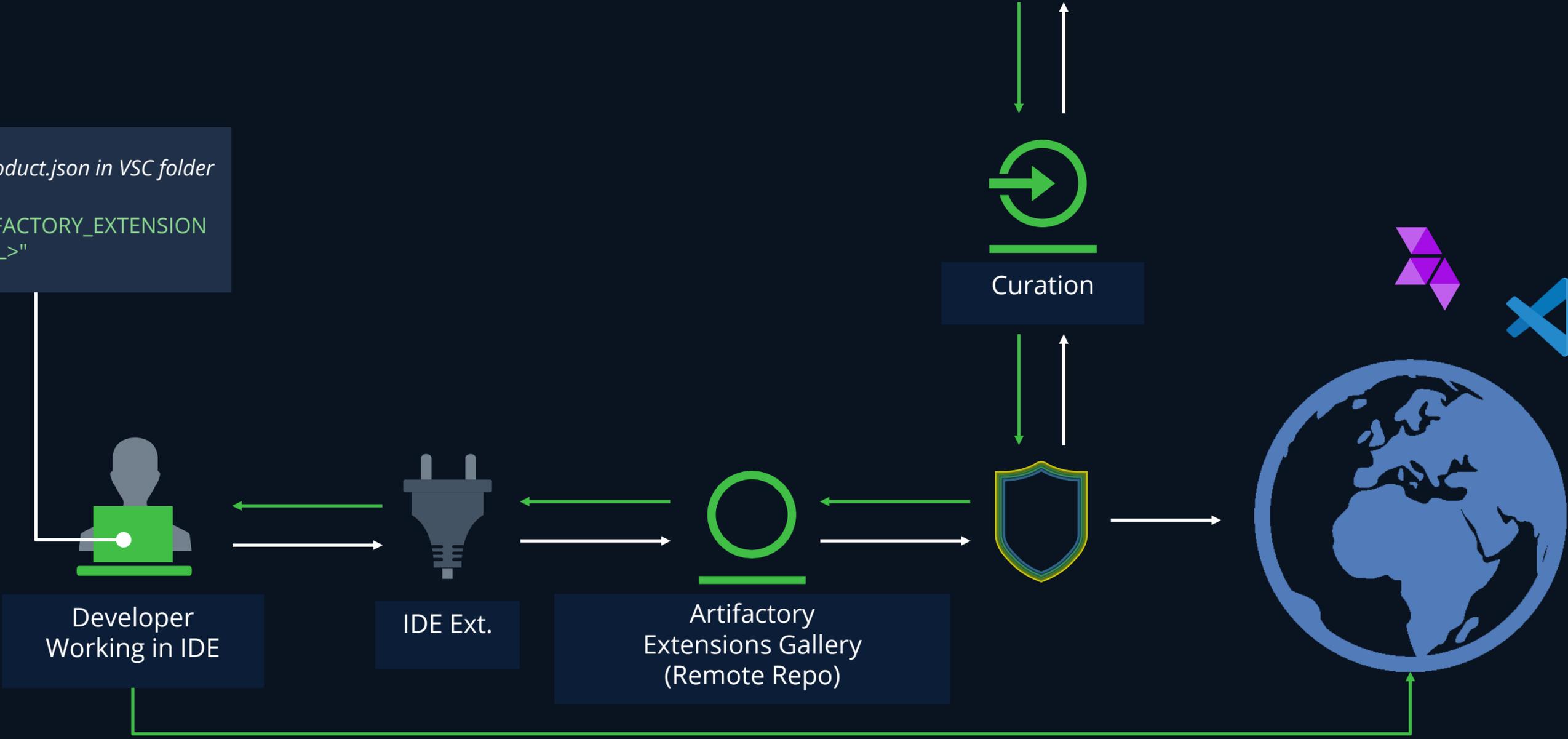
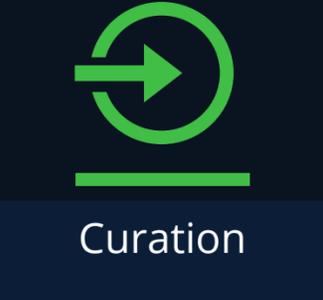
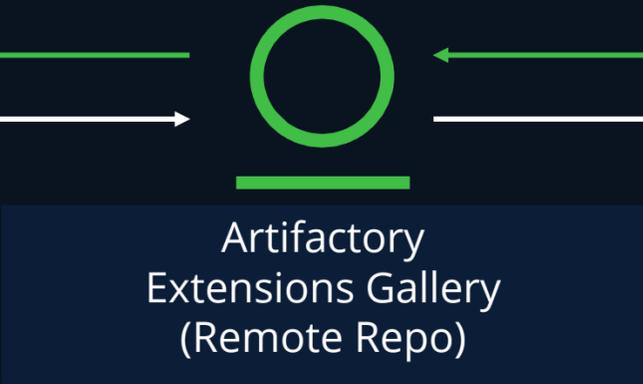
At the bottom right, there are 'Cancel' and 'Save Policy' buttons.

This is a blurred duplicate of the screenshot above, showing the same configuration steps and details for the 'Immature Package' policy.



# Curating Developer Extensions

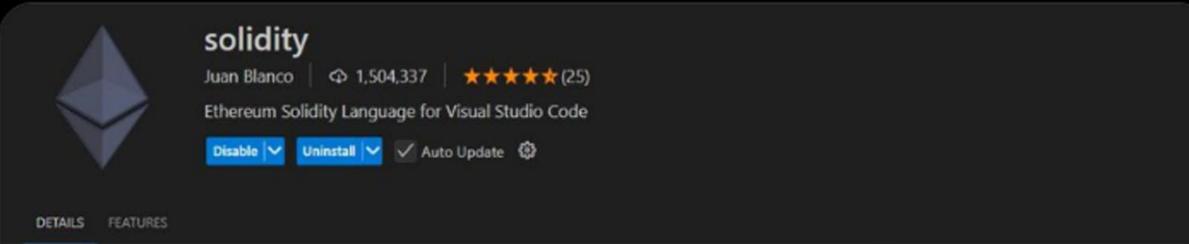
```
// in the file product.json in VSC folder  
"serviceUrl":  
"<YOUR_ARTIFACTORY_EXTENSION_GALLERY_URL>"
```



# The Challenge

**Juan Blanco** @juanfranblanco

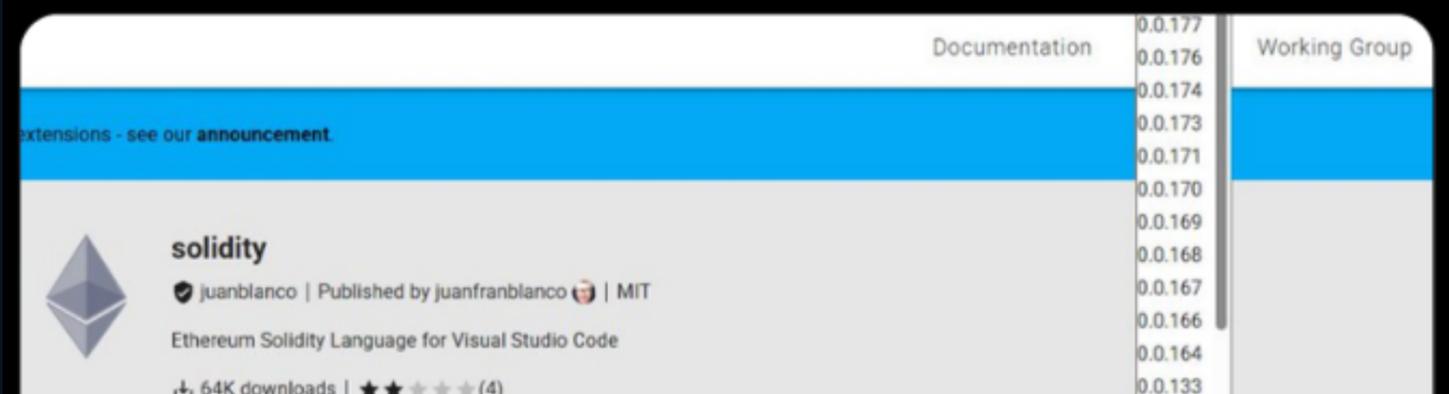
The extension that was impersonating vscode solidity (and many others following the same pattern) have been removed. We have seen that a fake extension or virus can spam many downloads (if that was their technique). So how to identify is the right extension? The best way is to look at the published date. The vscode solidity extension was published on the 2015-11-19 at 7:35 am, the published date cannot be faked. The extension was one of the first ones in the marketplace, after the official announcement of the extension sdk the day before. So in case of doubt, when choosing any extension.. check the date. @ethereum @code



**solidity**  
Juan Blanco | 1,504,337 | ★★★★★ (25)  
Ethereum Solidity Language for Visual Studio Code  
Disable | Uninstall | Auto Update

**Juan Blanco** @juanfranblanco

In OpenVSX (used by cursor) this is the right extension [open-vsx.org/extension/juan...](https://open-vsx.org/extension/juan...) OpenVSX allows you to download and view older versions of the extension, this is the best way to validate fake extensions there as you can see the first time that the extension was published there was 5 years ago and it was version 0.095.



Documentation	Working Group
0.0.177	
0.0.176	
0.0.174	
0.0.173	
0.0.171	
0.0.170	
0.0.169	
0.0.168	
0.0.167	
0.0.166	
0.0.164	
0.0.163	
0.0.133	

**solidity**  
juanblanco | Published by juanfranblanco | MIT  
Ethereum Solidity Language for Visual Studio Code  
64K downloads | ★★★★★ (4)



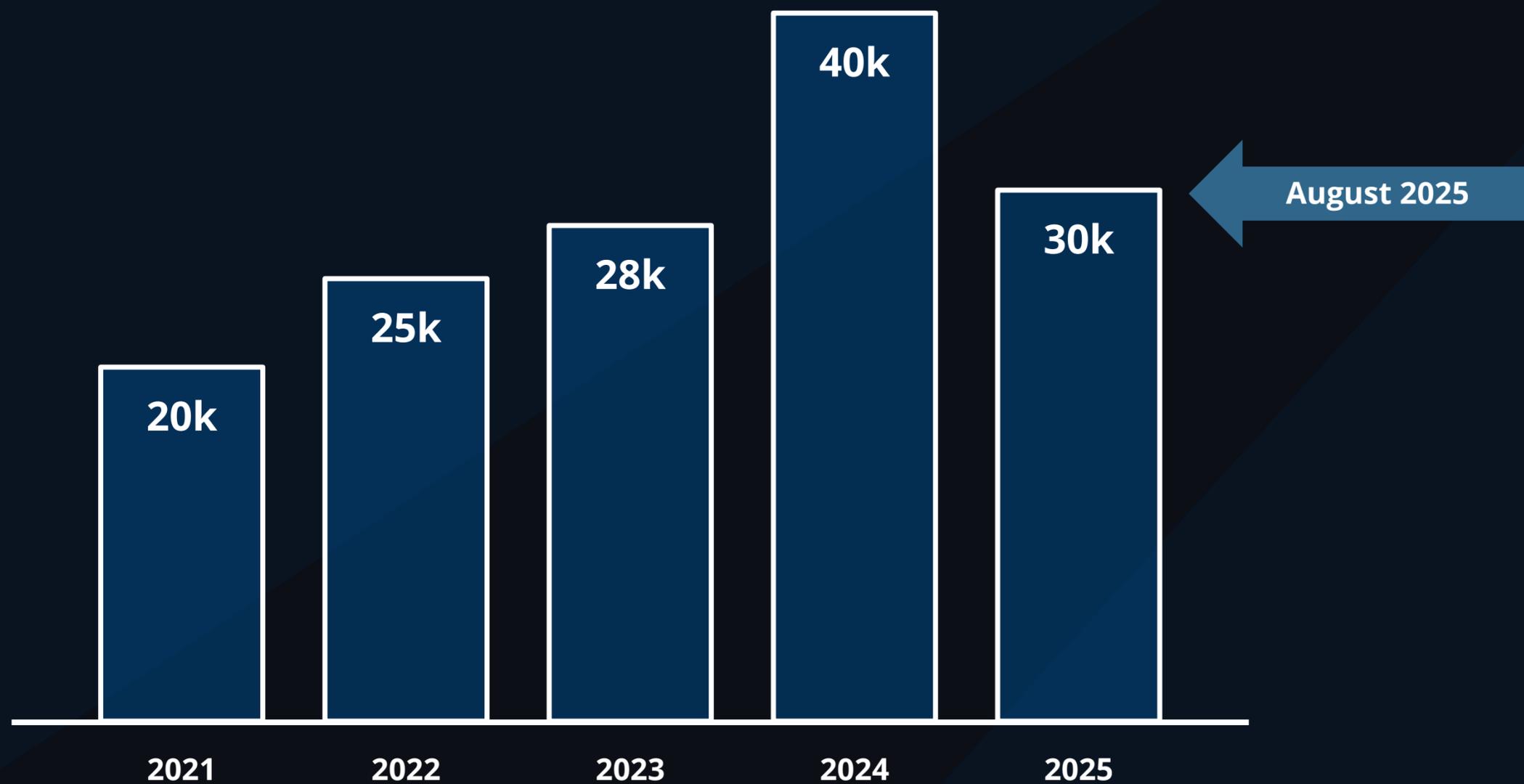
While Attackers are  
**Targeting the New**



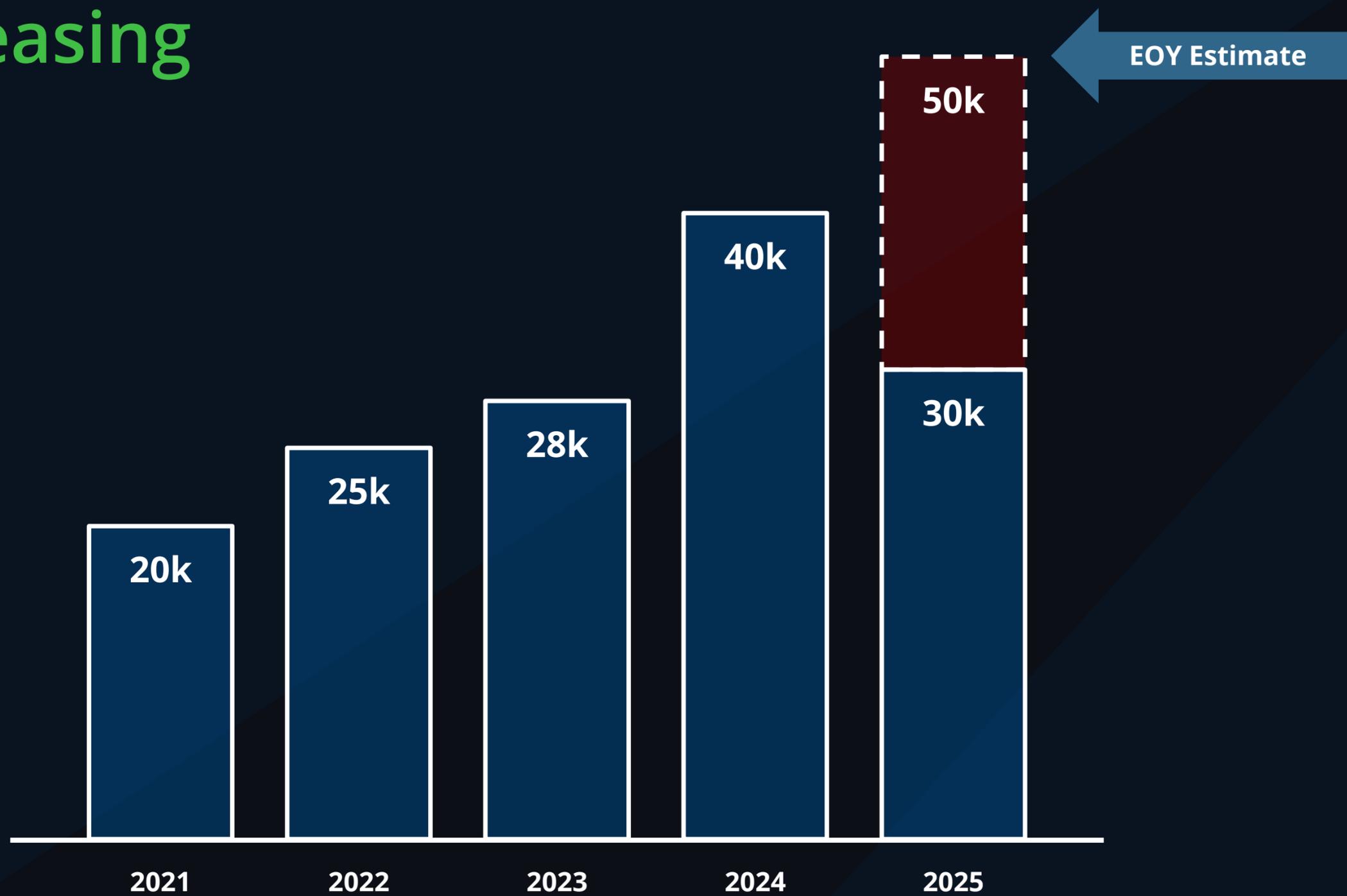


Developers are  
**Swamped by the Old**

# Number of CVEs are Ever Increasing



# Number of CVEs are Ever Increasing

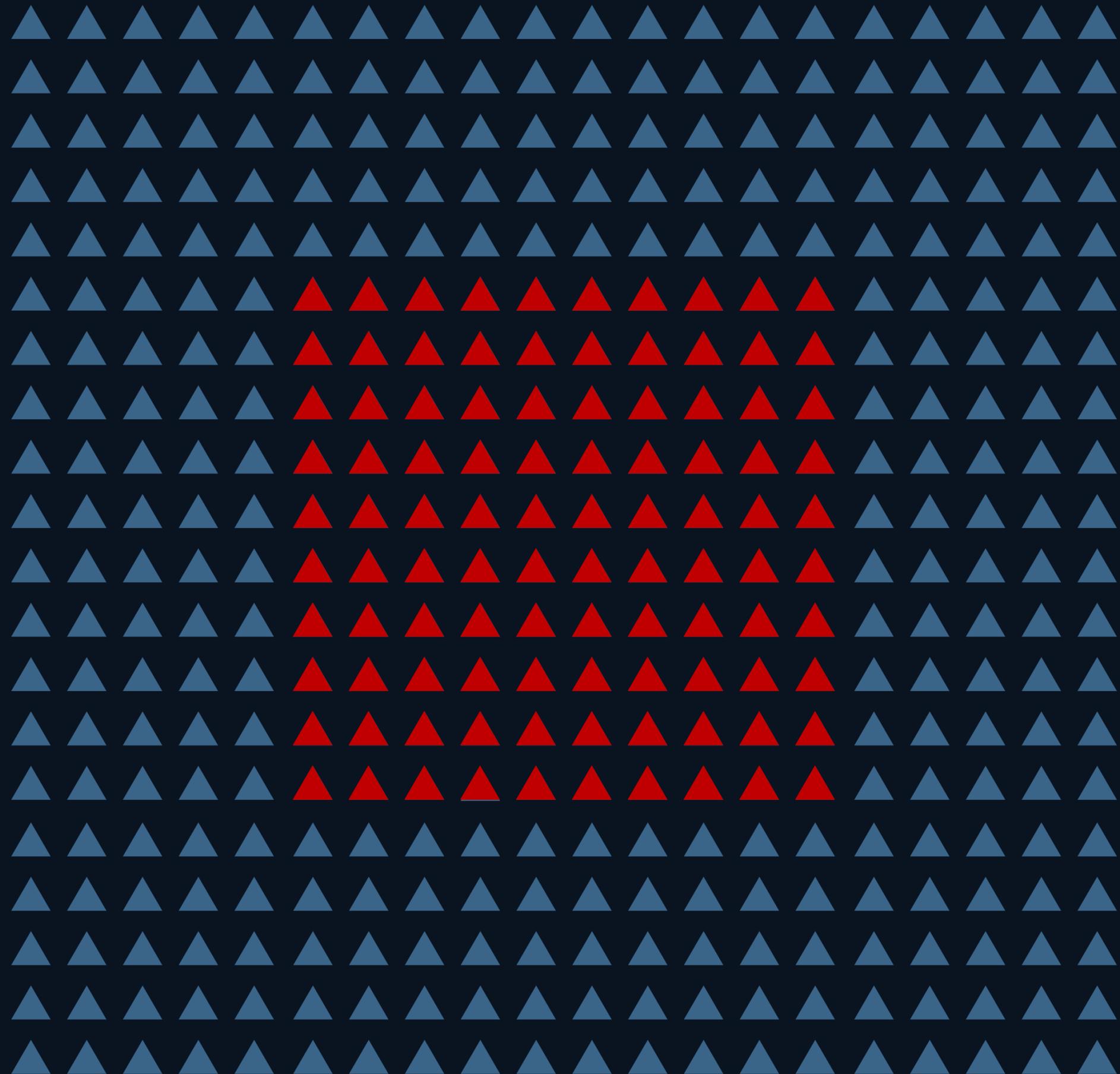


SLA times for Fixing CVEs  
are Constantly Decreasing



# We need to focus on CVEs

that are Critical  
that Can be Exploited  
and Run in Production





88%

of critical severity CVEs  
are grossly inflated

Exploitation Requires the Victim to  
**Download and Execute an  
Untrusted Malicious Package**



## 🦋 CVE-2023-29402 Detail

### Description

The go command may generate unexpected code at build time when using cgo. This may result in unexpected behavior when running a go program which uses cgo. This may occur when running an untrusted module which contains directories with newline characters in their



NIST: NVD

Base Score:

**9.8 CRITICAL**

ADP: CISA-ADP

Base Score:

**9.8 CRITICAL**



Enriched data by JFrog  
security research team



**CVE-2023-29402**  
**CVSS 9.8 Critical**

# We need to focus on CVEs

that are Critical  
that Can be Exploited  
and Run in Production





**73%** of  
**CVES**

don't even have an exploit



**85%** of  
**CVES**

aren't exploitable with common usage

# CVE Contextual Analysis

## What Does it Do?

- Flags vulnerabilities that are **applicable** and **exploitable** in your organization
- Provides clear and **research-backed remediation guidance**

## Why is it Important?

- Eliminates noise by focusing on **relevant vulnerabilities only**
- Improves **remediation efficiency**, eliminates wasted effort

*88% Critical and 57% High CVE scores are not as severe as indicated by the CVSS score*

CVE-2021-44228 XRAY-191654

Enriched data by JFrog Research team

Severity: Critical: CVSS V3 from NVD

CVSS Score: 10 (v3) | 9.3 (v2)

JFrog Research Severity: Critical

Contextual Analysis: **Applicable**

CWE: [CWE-20](#) Improper Input Validation  
[CWE-917](#) Improper Neutralization of Special Elements used in an Express...  
[CWE-400](#) Uncontrolled Resource Consumption  
[CWE-502](#) Deserialization of Untrusted Data

JFrog Research Contextual Analysis Public Sources Impact Paths References

Why is this **Applicable**

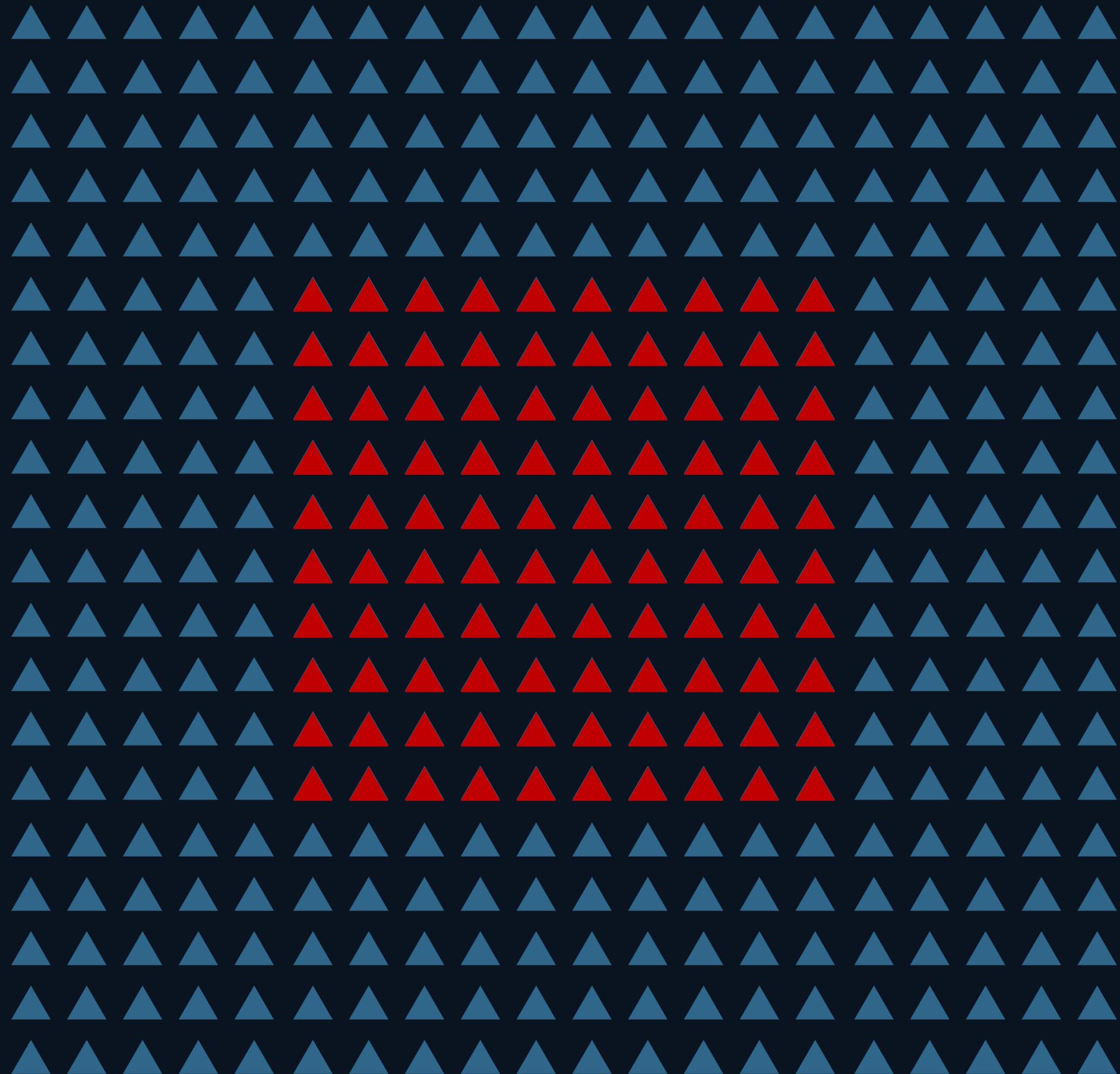
The following evidences were found

The vulnerable function info/fatal/log/warn/trace/error/debug is called with external input

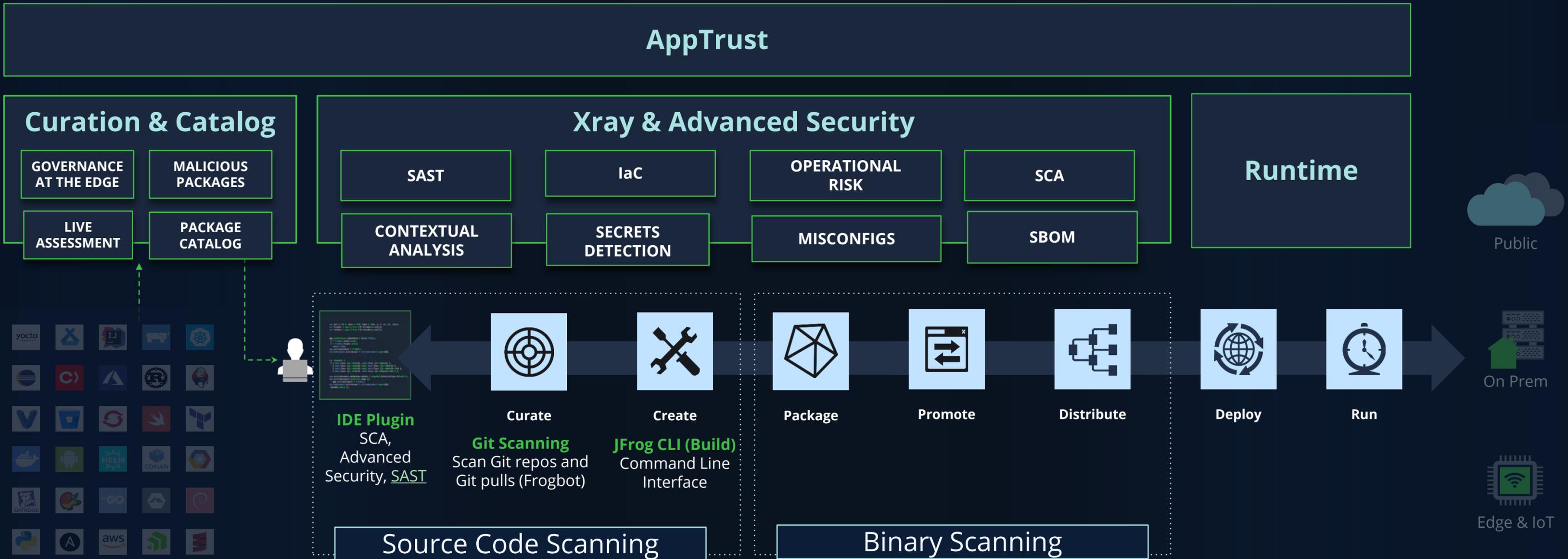
Path	/project-java/Log4jSocketServer.jar/log4j-api-2.8.1.jar/org/apache/logging/log4j
Location	LogManager::<clinit>
Path	/project-java/Log4jSocketServer.jar/log4j-api-2.8.1.jar/org/apache/logging/log4j
Location	LogManager::getContext
Path	/project-java/Log4jSocketServer.jar/log4j-api-2.8.1.jar/org/apache/logging/log4j

# We need to focus on CVEs

that are Critical  
that Can be Exploited  
and Run in Production



# End-to-end Security with JFrog



# Example: Establishing Trust with Evidence



# We Need to Focus On

CVEs that are Critical  
that Can be Exploited  
and Run in Production





Thank You