

HACKINBO[®]

Winter **2025** Edition

25^a EDIZIONE



America's Cyber Defense Agency

NATIONAL COORDINATOR FOR CRITICAL INFRASTRUCTURE SECURITY AND RESILIENCE

Search

Topics

Spotlight

Resources & Tools

News & Events

Careers

About

[Home](#) / [News & Events](#) / [Cybersecurity Advisories](#) / [Alert](#)

Widespread Supply Chain Compromise Impacting npm Ecosystem

SHARE:

ALERT

Widespread Supply Chain Compromise Impacting npm Ecosystem

THAT'S YOUR OPEN SOURCE,
NOT SOMEONE ELSE'S

Release Date: September 2025

CISA is releasing this Alert to provide guidance in response to a widespread software supply chain compromise involving the world's largest JavaScript registry, npmjs.com. A self-replicating worm—publicly known as "Shai-Hulud"—has compromised over 500 packages.^[i]

After gaining initial access, the malicious cyber actor deployed malware that scanned the environment for sensitive credentials. The cyber actor then targeted GitHub Personal Access Tokens (PATs) and application programming

Hack and Defend (your) Open Source

Roman Zhukov

Principal Architect - Security Communities Lead
ex. - Head of Product Security at Intel's Data Center SW BU
Security Lead and Contributor to Open Source

DISCLAIMER

The opinions expressed are solely my own and do not necessarily reflect the official views or opinions of my current or previous employer(s).



Do you use open source in your work?

1,658 projects scanned by Black Duck audits



97%



70%



An average

911

OSS components were found per application

64%



of OSS components were **transitive dependencies**

86%

of the codebases contained at least one vulnerability

81%

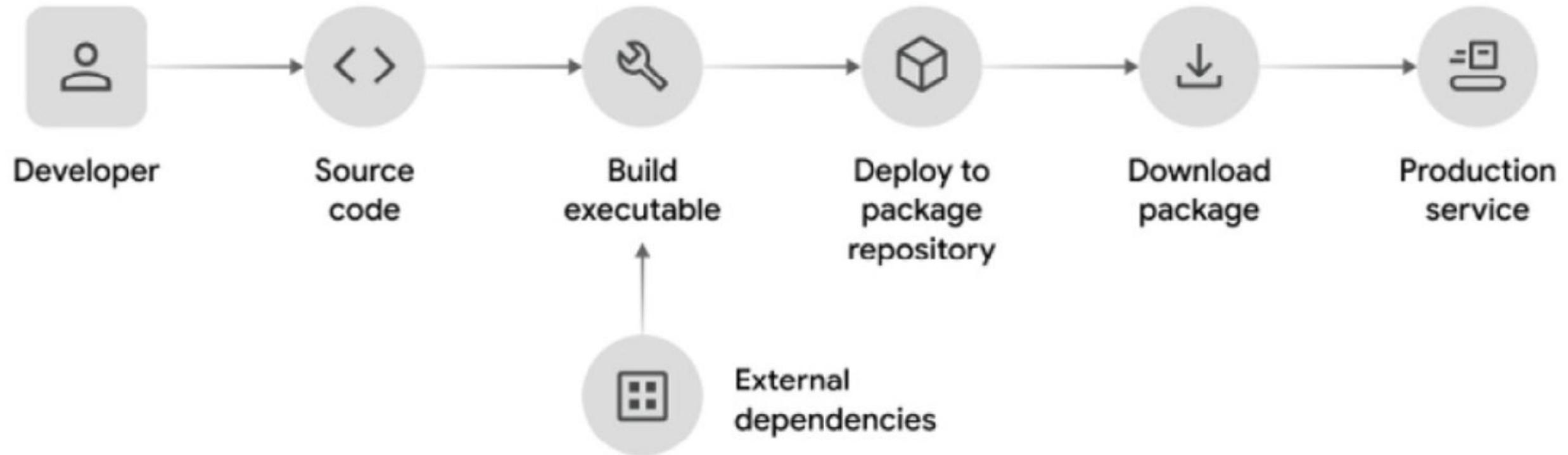
of the codebases contained high- or critical-risk vulnerabilities

Maintenance and Operational Risk

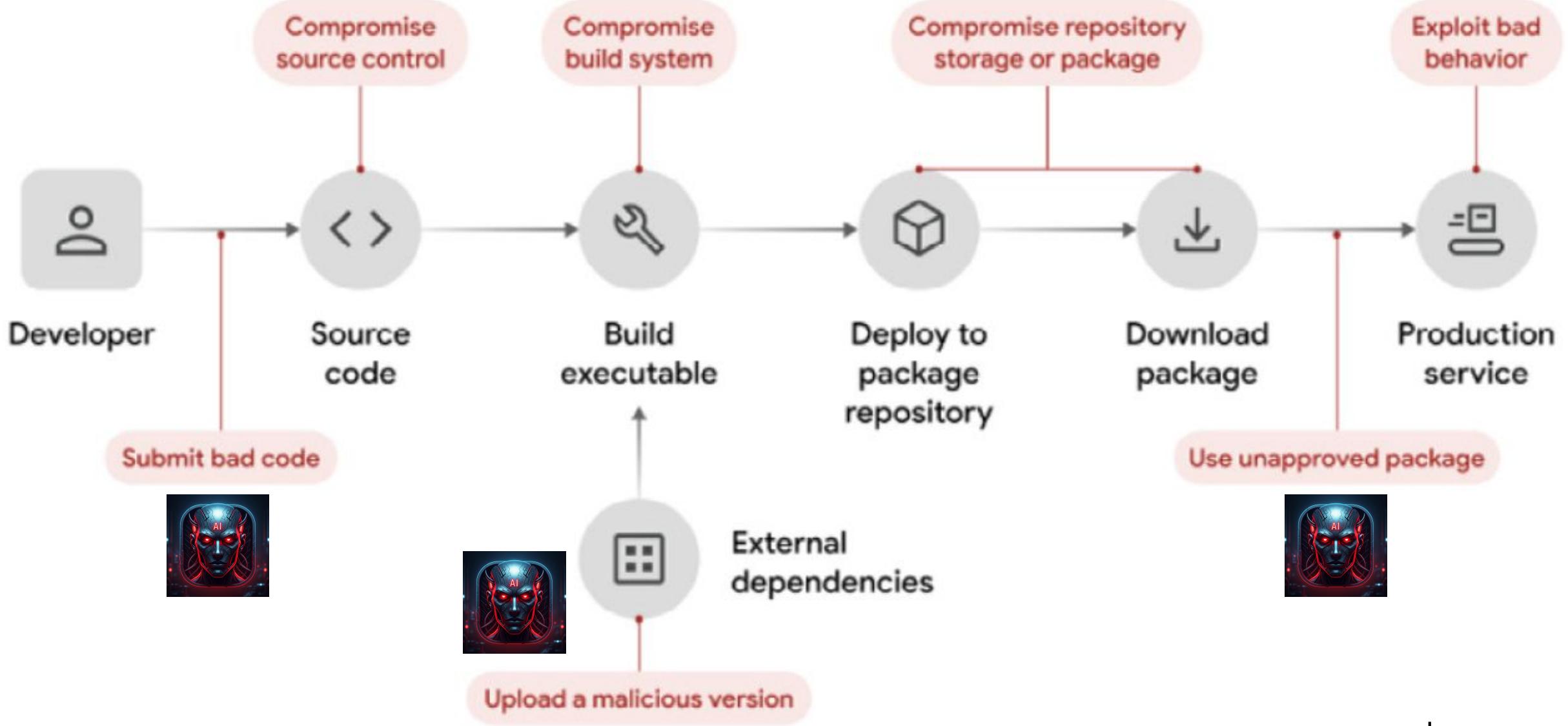
91% of all codebases contained outdated OSS components

90%

of all codebases contained components more than 10 versions behind the most current version



src: google.com



src: google.com

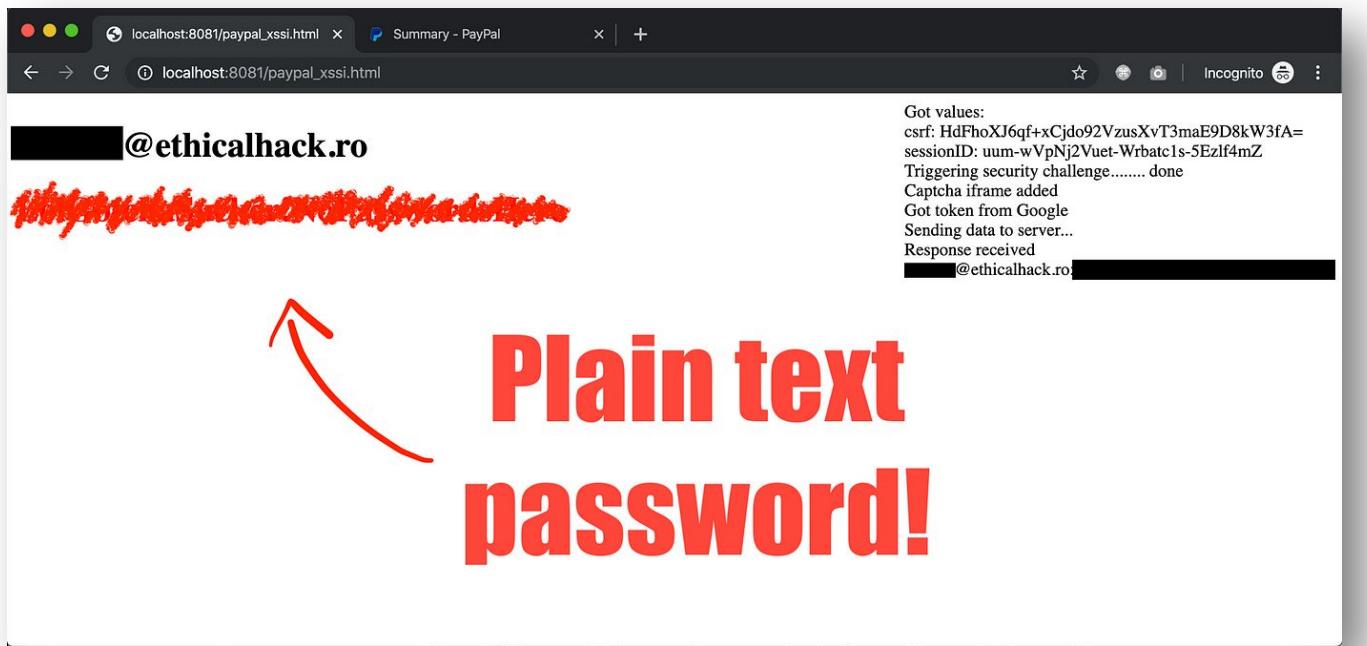
The Top 10 OSS Risks



owasp.org/www-project-open-source-software-to-p-10/

- | | |
|-------------|---|
| OSS-RISK-1 | Known Vulnerabilities |
| OSS-RISK-2 | Compromise of Legitimate Package |
| OSS-RISK-3 | Name Confusion Attacks |
| OSS-RISK-4 | Unmaintained Software |
| OSS-RISK-5 | Outdated Software |
| OSS-RISK-6 | Untracked Dependencies |
| OSS-RISK-7 | License Risk |
| OSS-RISK-8 | Immature Software |
| OSS-RISK-9 | Unapproved Change (Mutable) |
| OSS-RISK-10 | Under/Over-Sized Dependency |

1 Dependency confusion



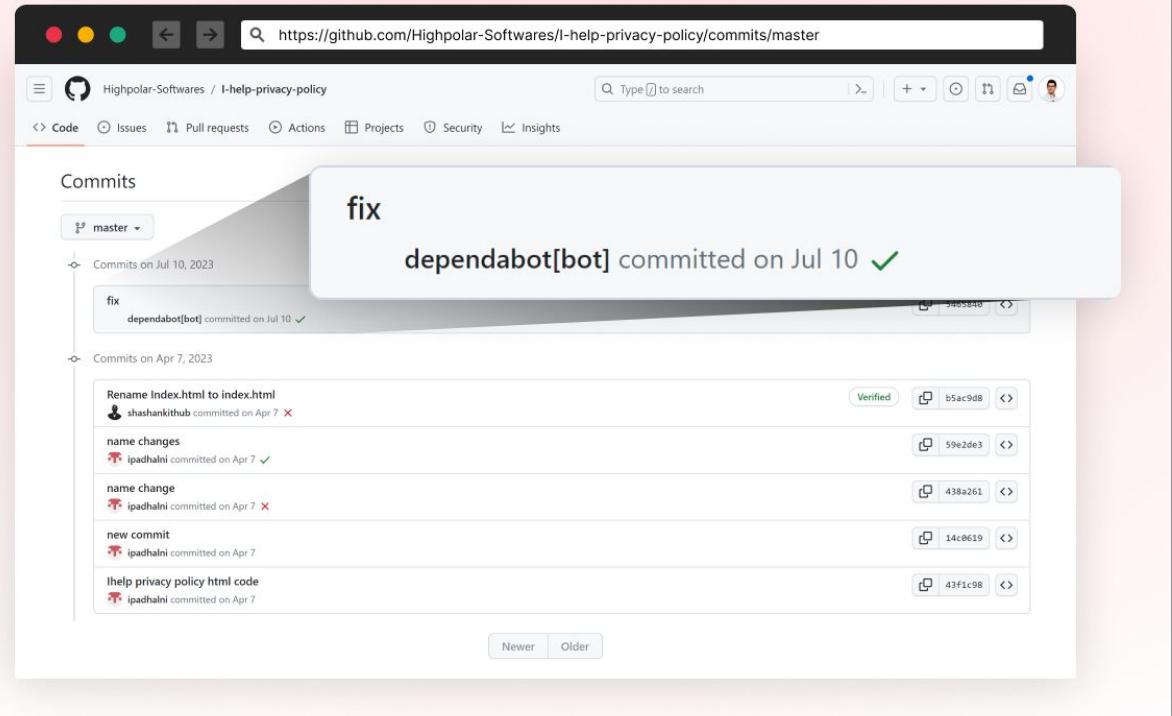
```
> npm install analytics-paypal
// Used to launch malicious script to
make DNS requests to
dns.alexbirsan-hacks-paypal.com
to run data exfiltration. Now
deprecated.
```

1 Dependency confusion

2 Typosquatting



EleuterAI instead of EleutherAI
*//fake LLM model was added to
Hugging Face*

1
2
3

Trojan injection

> **dependabot[bot]** committed
//Malicious pull request that's accepted contains info stealer

npm Search packages

coa TS
3.1.3 • Public • Published an hour ago

Readme Explore BETA 3 Dependencies 159 Dependents 34 Versions

Tip: Click on a version number to view a previous version's package page

Install > npm i coa

Current Tags

Version	Downloads (Last 7 Days)	Tag
3.1.3	0	latest

Version History

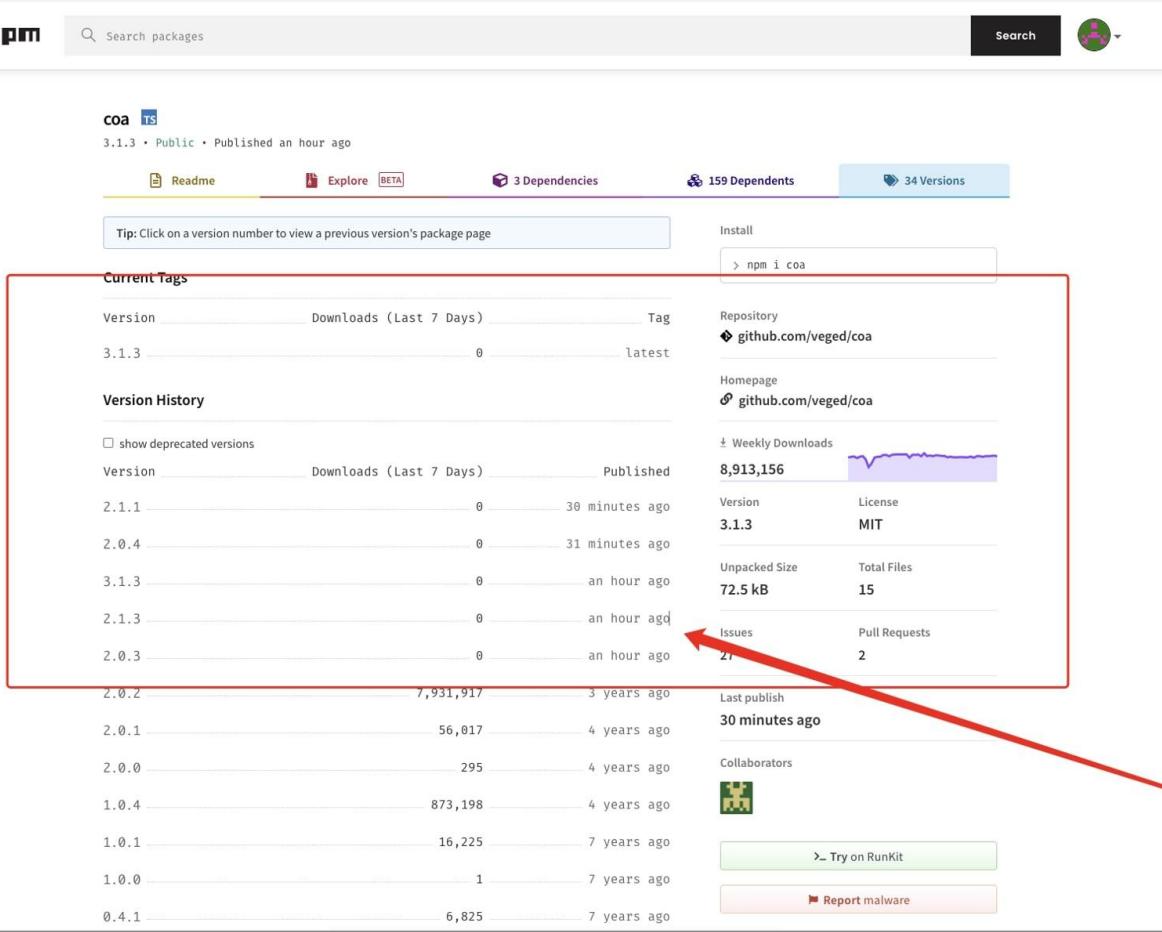
show deprecated versions

Version	Downloads (Last 7 Days)	Published
2.1.1	0	30 minutes ago
2.0.4	0	31 minutes ago
3.1.3	0	an hour ago
2.1.3	0	an hour ago
2.0.3	0	an hour ago
2.0.2	7,931,917	3 years ago
2.0.1	56,017	4 years ago
2.0.0	295	4 years ago
1.0.4	873,198	4 years ago
1.0.1	16,225	7 years ago
1.0.0	1	7 years ago
0.4.1	6,825	7 years ago

Repository github.com/veged/coa
Homepage github.com/veged/coa
Weekly Downloads 8,913,156

Version 3.1.3 License MIT
Unpacked Size 72.5 kB Total Files 15
Issues 27 Pull Requests 2

Last publish 30 minutes ago
Collaborators 
[Try on RunKit](#) [Report malware](#)



1

2

3

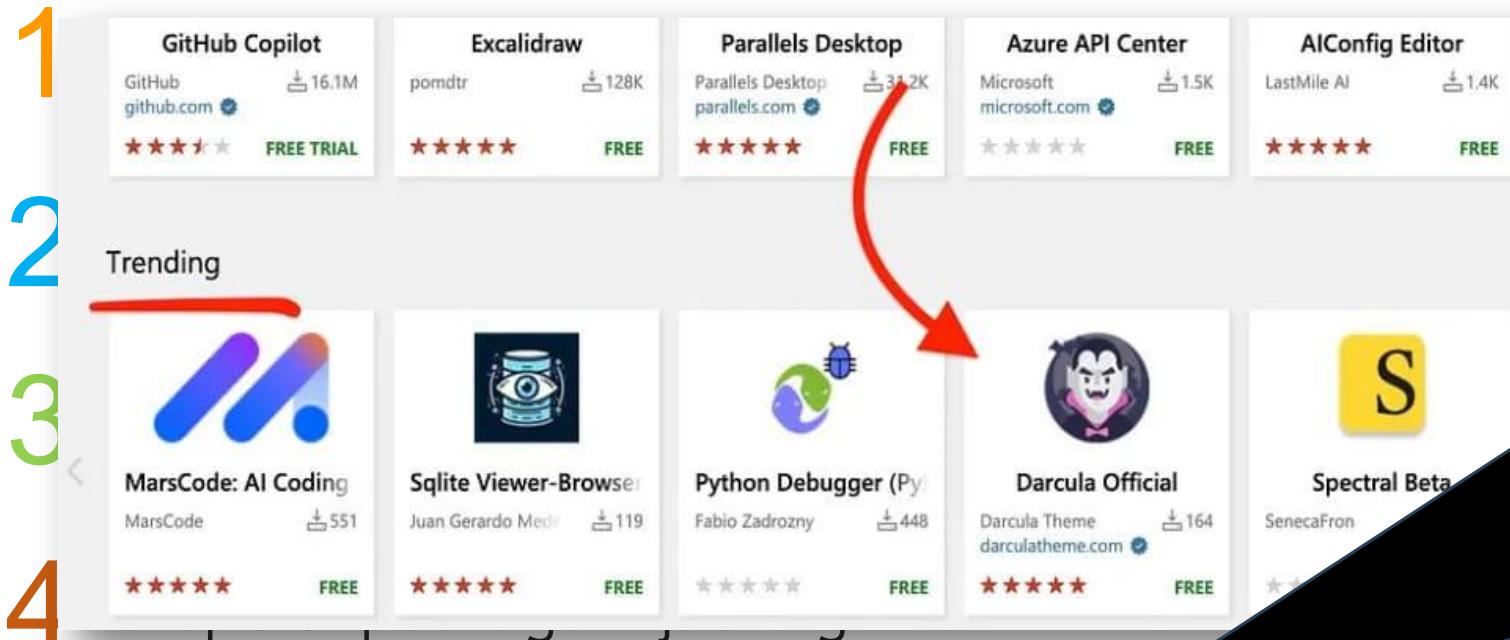
4

Repo or package Hijacking

> coa
was hijacked over npm account takeover incorporating password stealer
//command line option parser with over 9 Million weekly downloads

INSPIRED BY A TRUE STORY

Homemadesoul.



> “Dracula Official” is a popular colour **VSCode Marketplace plugin** with **over 7 million installs**.
//Malicious “**Darcula Official**” is an **infostealer** sends user’s data to a remote server via an **HTTPS POST**

5 Dev environment compromise

Top 10 CI/CD Security Risks

owasp.org/www-project-top-10-ci-cd-security-risks/



CICD-SEC-1	Insufficient Flow Control Mechanisms
CICD-SEC-2	Inadequate Identity and Access Management
CICD-SEC-3	Dependency Chain Abuse
CICD-SEC-4	Poisoned Pipeline Execution (PPE)
CICD-SEC-5	Insufficient PBAC (Pipeline-Based Access Controls)
CICD-SEC-6	Insufficient Credential Hygiene
CICD-SEC-7	Insecure System Configuration
CICD-SEC-8	Ungoverned Usage of 3rd Party Services
CICD-SEC-9	Improper Artifact Integrity Validation
CICD-SEC-10	Insufficient Logging and Visibility

Usual Tuesday Morning

Intel-Innersource

Overview Repositories Packages People 18

source 1 inner-source

Intel-Innersource

Intel-Innersource

This org.

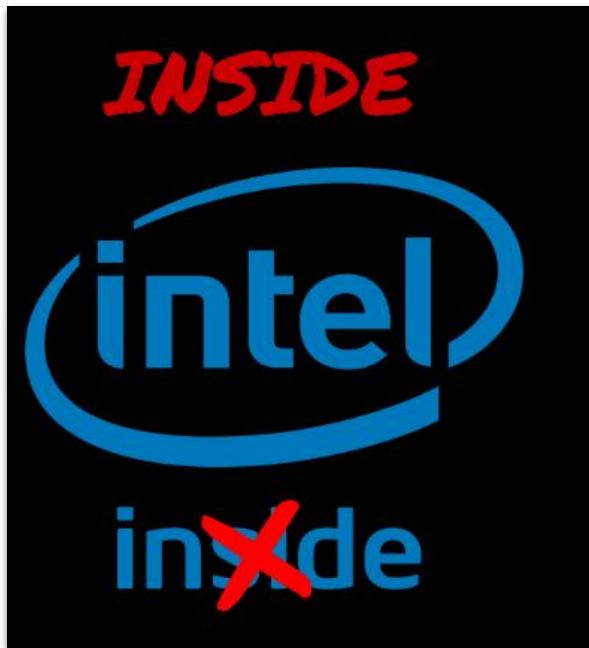
© 2025 GitHub, Inc.

share my personal information

Follow

People

Report abuse

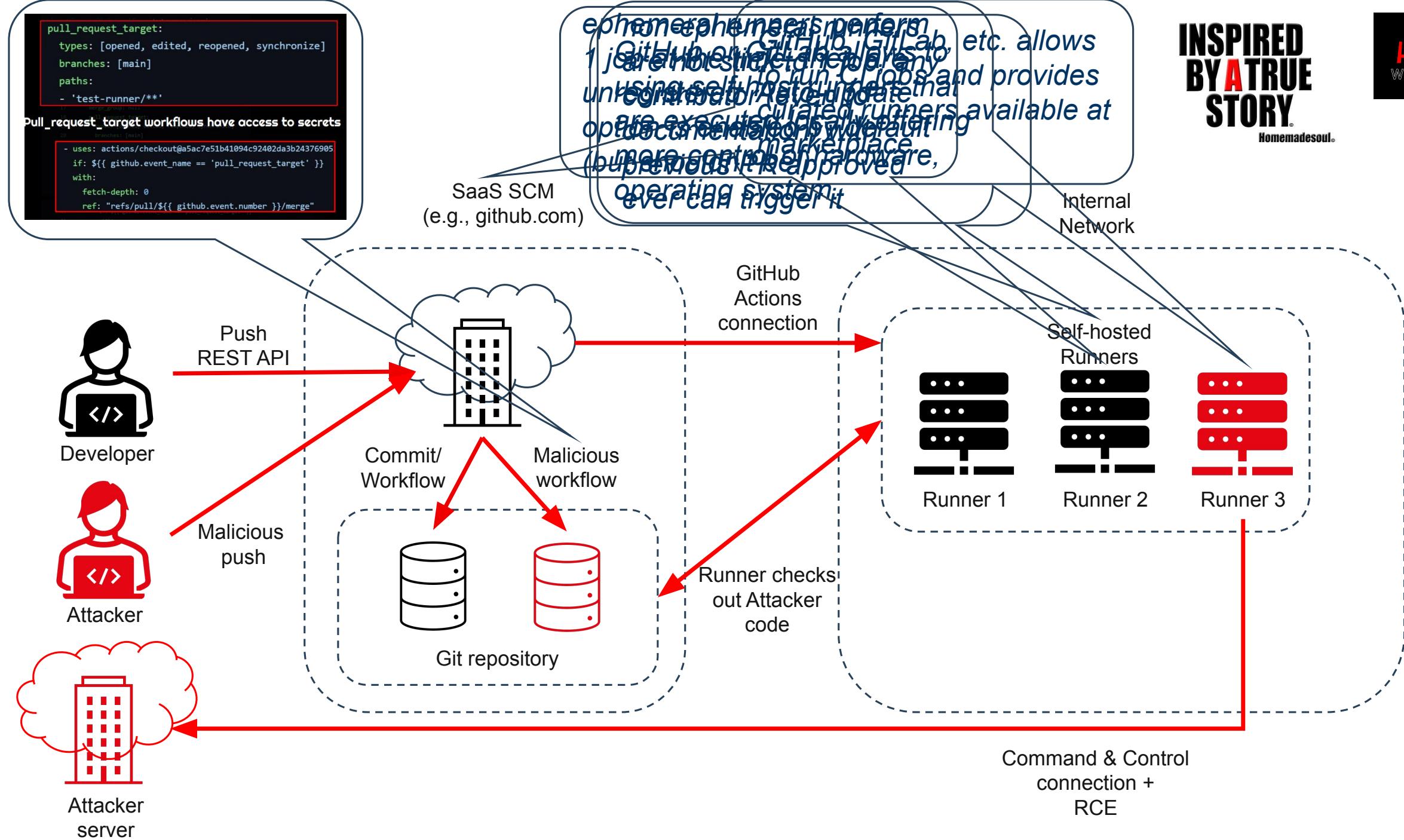


UNPRECEDENTED ACCESS

```

447 },
448 {
449   "id": 472953435,
450   "node_id": "R_kgDOHDCyWw",
451   "name": "core-royal",
452   "full_name": "intel-restricted/.core-royal",
453   "private": true,
454   "owner": {
455     "login": "intel-restricted",
456     "id": 71398875,
457     "node_id": "MDIyOk9yZ2FuaXphdGlvbjcxMzk40Dc1",
458     "avatar_url": "https://avatars.githubusercontent.com/u/",
459     "gravatar_id": "",
460     "url": "https://api.github.com",
461     "html_url": "https://github.com",
462     "followers_url": "https://api.github.com/users/intel-restricted/followers",
463     "following_url": "https://api.github.com/users/intel-restricted/following/{user}",
464     "gists_url": "https://api.github.com/users/intel-restricted/gists{/gist_id}",
465     "starred_url": "https://api.github.com/users/intel-restricted/starred{/owner}",
466     "subscriptions_url": "https://api.github.com/users/intel-restricted/subscriptions",
467     "organizations_url": "https://api.github.com/users/intel-restricted/orgs",
468     "repos_url": "https://api.github.com/users/intel-restricted/repos",
469     "events_url": "https://api.github.com/users/intel-restricted/events{/privacy}",
470     "received_events_url": "https://api.github.com/users/intel-restricted/received_events",
471     "type": "Organization",
472     "site_admin": false
473   },
474   "html_url": "https://github.com/intel-restricted/.core-royal",
475   "description": "Royal Core Intellectual Property"
476 },
477 "fork": false,

```



How do you choose an open source?



tj-actions / changed-files

Code

Issues 6

Pull requests 8

Discussions

Actions

Projects

Security



changed-files

Public

README

Code of conduct

Contributing

MIT license

Security

Ubuntu

macOS

Windows

Used by

31529

code quality

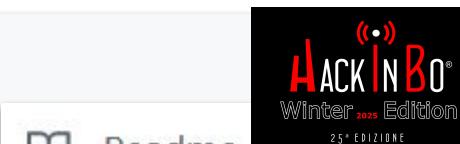
A

CI passing

Update release version passing

all contributors 30

changed-files



Readme

MIT license

Code of conduct

Contributing

Security policy

Activity

Custom properties

2.6k stars

10 watching

316 forks

Report repository

Releases 15

v47 Latest

on Sep 13

The Malicious Imposter Commit

tj-actions / changed-files Q Type / to search

<> Code Issues 7 Pull requests 2 Discussions Actions Projects Security 1 Insights

⚠️ This commit does not belong to any branch on this repository, and may belong to a fork outside of the repository.

Commit 0e58ed8

star renovate[bot] committed 12 hours ago

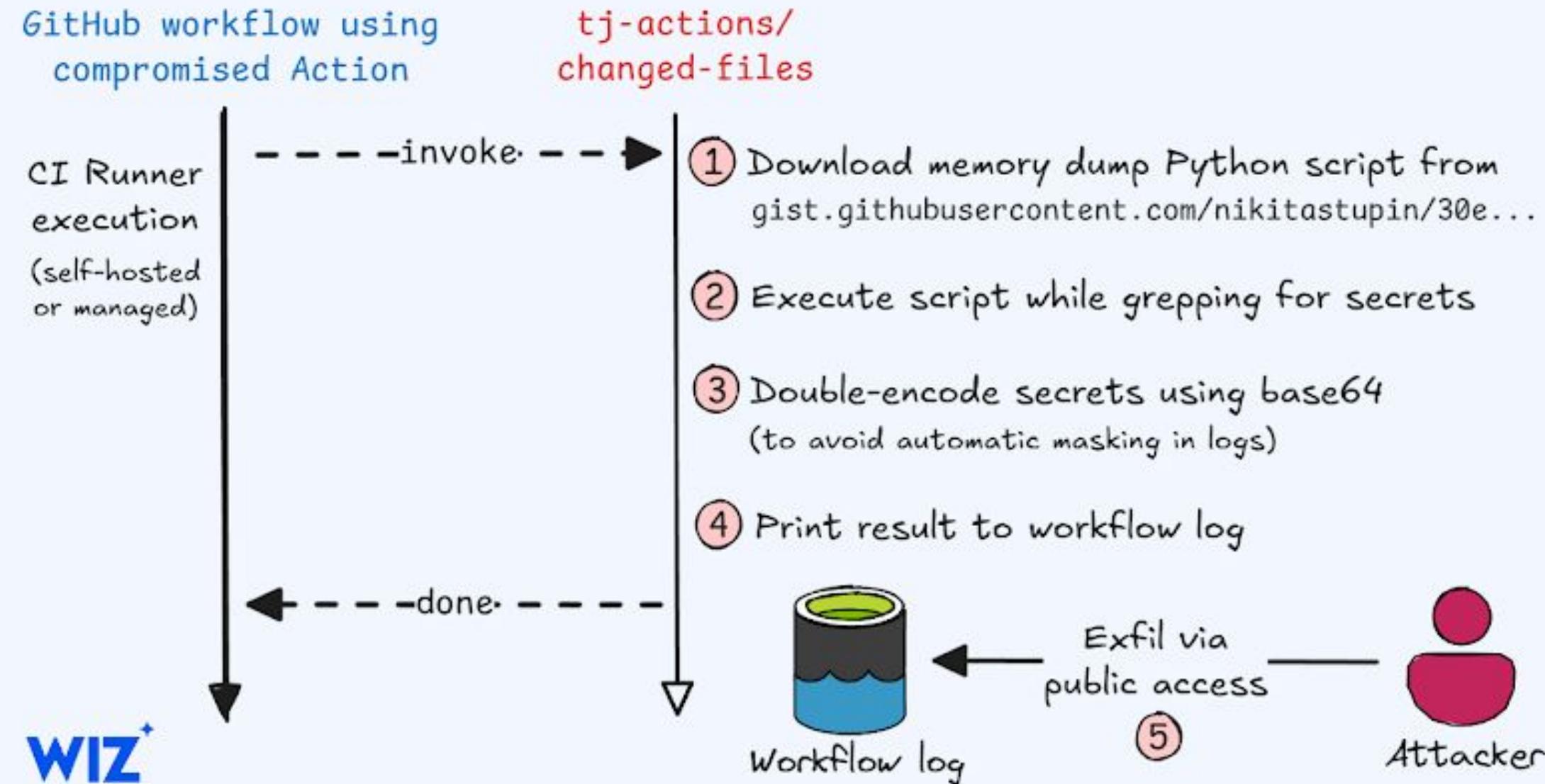
chore(deps): lock file maintenance (#2460)

· key v45.0.7 ... v1

Filter files... 1 file changed +12 -1 lines changed

tj-actions/changed-files Supply Chain Attack

Post-compromise secret exfiltration flow from dependent repositories



How to “hack” open source?

Offensive AI Compilation

A curated list of useful resources that cover Offensive AI.

Contents

- **Abuse**
 - **Adversarial Machine Learning**
 - **Attacks**
 - **Extraction**
 - **Limitations**
 - **Defensive actions**
 - **Useful links**
 - **Inversion (or inference)**

OSINT

- [SNAP_R](#): Generate automatically spear-phishing posts on social media. stars 150
- [SpyScrap](#): SpyScrap combines facial recognition methods to filter the results and uses natural language processing to obtain important entities from the website the user appears. stars 201

Phishing

- [DeepDGA](#): Implementation of DeepDGA: Adversarially-Tuned Domain Generation and Detection. stars 26
- [ScamAgents](#): How AI Agents Can Simulate Human-Level Scam Calls

Threat Intelligence

- [From Sands to Mansions: Enabling Automatic Full-Life-Cycle Cyberattack Construction with LLM](#)

Side channels

- [SCAAML](#): Side Channel Attacks Assisted with Machine Learning. stars 171



Abuse

Exploiting the vulnerabilities of AI models.

Adversarial Machine Learning

Adversarial Machine Learning is responsible for assessing their weaknesses and providing countermeasures.

Attacks

It is organized into four types of attacks: extraction, inversion, poisoning and evasion.

Extraction attacks
(or theft model)

Inversion attacks
(or inference)

Poisoning attacks

Evasion attacks

github.com/jiep/offensive
-ai-compilation

popup.html	chore: add project docs and initial repository files	3 weeks ago
popup.js	chore: add project docs and initial repository files	3 weeks ago
README	Code of conduct	MIT license
	 	

Select2AI_Extension - Effortless Text Selection and AI Interaction

[Download Now!](#)



Overview

Select2AI_Extension is a powerful browser extension that enables you to easily select text on any webpage. With this extension, you can instantly interact with GitHub Models AI to get summaries, explanations, and answers. You can also ask custom questions in a user-friendly floating window, complete with light and dark themes and smooth animations.

Contributors 2



samrat225
NK2552003 Nitish

Languages



JavaScript 59.0% • CSS 29.1%
HTML 11.9%

Behavior activities

MALICIOUS

Generic archive extractor

- WinRAR.exe (PID: 7420)

SMARTLOADER has been detected

- cmd.exe (PID: 7308)

SUSPICIOUS

Reads security settings of Internet Explorer

- lua.exe (PID: 3200)

Checks for external IP

- svchost.exe (PID: 2276)

https://github.com/samrat225>Select2AI_Extension ID: 5212e12e-5c6c-49bb-b4ad-7cf04d797ca1

HIGH **VERIFIED**

Threat Description
This repository has been identified as a smartloader.

Timeline
Reported: 11/2/2025, 6:41:59 PM
Verified: 11/2/2025, 7:23:40 PM

Submitted By
@mazznrz

Tags
github smartLoader Loader

External References

[Evidence](#)



OPEN SOURCE MALWARE

A community database, API and collaboration platform to help identify and protect against open-source malware

[Browse Threats](#) [Sign In to Report](#)

70,271 Malicious Packages 12 Malicious Repositories 1 Malicious CDNs

opensourcemalware.com

Defense Frameworks and Tools

Secure Supply Chain Consumption Framework (S2C2F)

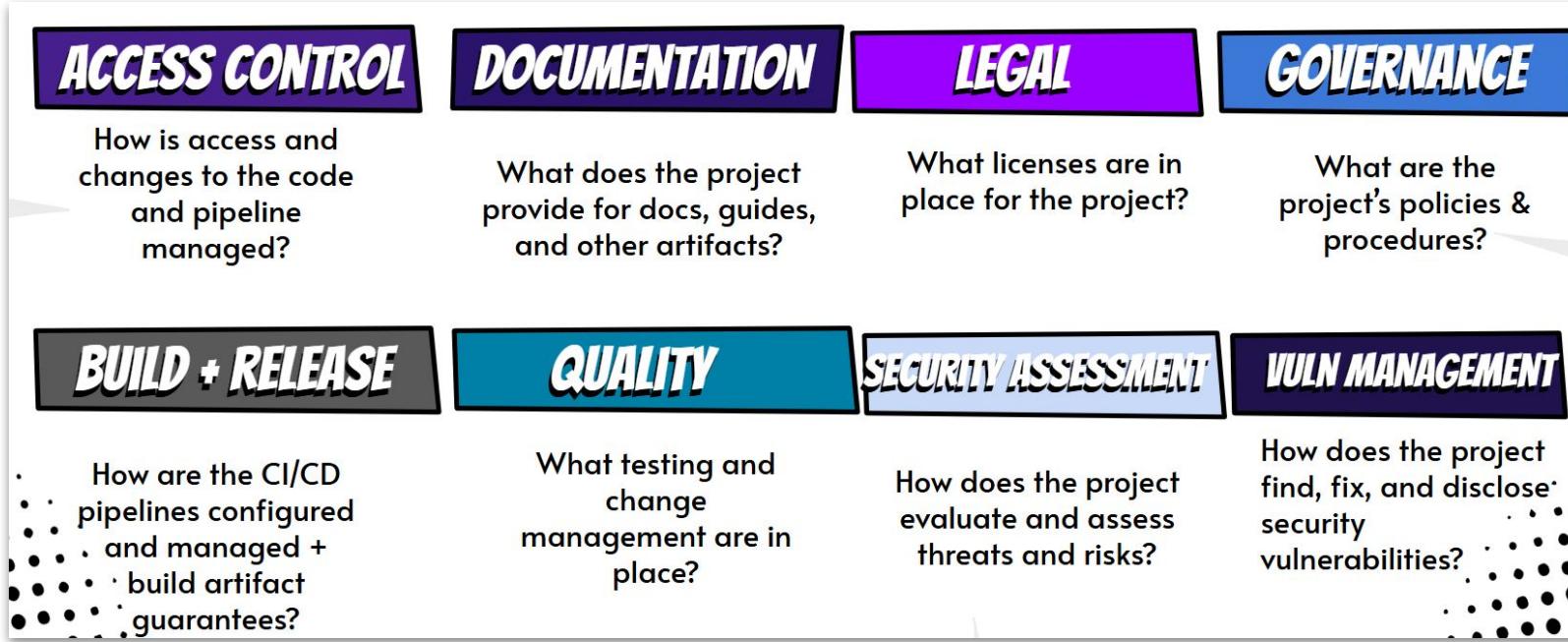


Level 1	Level 2	Level 3	Level 4
Minimum OSS Governance Program <ul style="list-style-type: none">• Use package managers [ING-1]• Local copy of artifact [ING-2]• Scan with known vulns [SCA-1]• Scan for software licenses [SCA-2]• Inventory OSS [INV-1]• Manual OSS updates [UPD-1]	Secure Consumption and Improved MTTR <ul style="list-style-type: none">• Scan for end of life [SCA-3]• Have an incident response plan [INV-2]• Auto OSS updates [UPD-2]• Alerts on vulns at PR time [UPD-3]• Audit that consumption is through approved ingestion method [AUD-2]• Validate integrity of OSS [AUD-3]• Secure package source file configuration [ENF-1]	Malware Defense and Zero-Day Detection <ul style="list-style-type: none">• Deny list capability [ING-3]• Clone OSS source [ING-4]• Scan for malware [SCA-4]• Proactive security reviews [SCA-5]• Enforce OSS provenance [AUD-1]• Enforce consumption from curated feed [ENF-2]	Advanced Threat Defense <ul style="list-style-type: none">• Validate the SBOMs of OSS consumed [AUD-4]• Rebuild OSS on trusted infrastructure [REB-1]• Digitally sign rebuilt OSS [REB-2]• Generate SBOM for rebuilt OSS [REB-3]• Digitally sign protected SBOMs [REB-4]• Implement fixes [FIX-1]

github.com/ossf/s2c2f

Security Baseline - OSPS

62 requirements across 3 levels of maturity covering 8 areas of security practices



OSPS-AC-02.01: When a new collaborator is added, the version control system **MUST** require manual permission assignment, or restrict the collaborator permissions to the lowest available privileges by default.

Focused - no **SHOULD**, only **MUST**
Realistic - practical to implement

[github.com/ossf
/security-baseline](https://github.com/ossf/security-baseline)

OWASP Secure Pipeline



1 Plan

2 Develop

3 Integrate

4 Release

5 Operate

Just released

Level	Purpose	Typical Users	Example Practices
Level 1 – Foundational	Establishes baseline pipeline hygiene. Focuses on simple, enforceable controls.	Small teams or organizations building their first secure pipeline.	MFA, endpoint protection, source control hardening, approved tooling.
Level 2 – Standard	Expands to automation, monitoring, and evidence of enforcement.	Mature DevSecOps teams managing regulated workloads.	Automated secret scanning, policy-as-code enforcement, release reviews.
Level 3 – Advanced	Demonstrates continuous verification, independence, and assurance at scale.	Enterprises or high-risk environments.	Automated gating, audit logging, verified builds, third-party attestations.

github.com/OWASP/www-project-spv

Automate It

The screenshot shows the GitHub repository page for Keycloak. At the top, there are links for README, Code of conduct, Apache-2.0 license, and Security. Below the repository name, there's a summary bar with metrics: latest release v26.2.5, openssf best practices passing, CLOMonitor Report A, openssf scorecard 9.3, stars 28k, commit activity 231/month, and translated 52%. A red arrow points from the 'openssf best practices' badge to the 'bestpractices.dev' section at the bottom.

KEYCLOAK

Open Source Identity and Access Management

Add authentication to applications and secure services with minimum effort. No need to deal with storing users or authenticating users.

Keycloak provides user federation, strong authentication, user management, fine-grained authorization, and more.

github.com/keycloak/

The screenshot shows the bestpractices.dev page for Keycloak. It features a large OpenSSF Best Practices badge. Below it, text says: "Projects that follow the best practices below can voluntarily self-certify and show that they've achieved an Open Source Security Foundation (OpenSSF) best practices badge." It also includes instructions for embedding the badge and links for expanding panels, showing all details, and showing only incomplete criteria.

Keycloak

Projects that follow the best practices below can voluntarily self-certify and show that they've achieved an Open Source Security Foundation (OpenSSF) best practices badge. [Show details](#)

If this is your project, please show your badge status on your project page! The badge status looks like this: [openssf best practices](#) [passing](#). Here is how to embed it: [Show details](#)

These are the [passing](#) level criteria. You can also view the [silver](#) or [gold](#) level criteria.

[Expand panels](#) [Show all details](#) [Show only incomplete criteria](#)

bestpractices.dev

github.com/ossf/
scorecard

The screenshot shows the OpenSSF Scorecard Report for the Keycloak GitHub repository. The report has a score of 9.3. It lists several categories with their respective scores and risk levels:

- Dangerous-Workflow: CRITICAL (Score: 10)
- Signed-Releases: HIGH (Score: 8)
- Vulnerabilities: HIGH (Score: 8)
- Code-Review: HIGH (Score: 10)
- Maintained: HIGH (Score: 10)
- Token-Permissions: HIGH (Score: 10)
- Binary-Artifacts: HIGH (Score: 10)
- Pinned-Dependencies: MEDIUM (Score: 10)

OpenSSF Scorecard Report

github.com/keycloak/keycloak

API URL: https://api.scorecard.dev/projects/github.com/keycloak/keycloak
COMMIT: 0e28bd398132fa23fa55f03ae65385db7e9efb7
GENERATED AT: 2025-06-09
SCORECARD VERSION: v5.2.1-5-g1f95bf1a

SORT: Risk level (desc)



OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Automate It



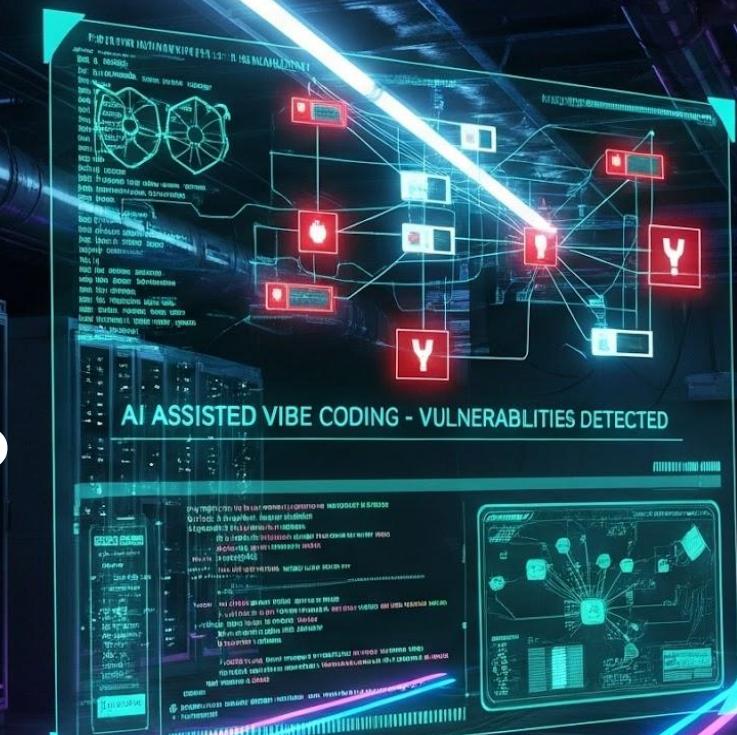
The screenshot shows the GitHub repository page for "mindersec/minder". At the top left is the Minder logo, a stylized purple bear head. To the right is the repository name "minder" in large blue letters. Below the name are four status badges: "Main passing" (green), "coverage 57%" (red), "License Apache2.0" (green), and "SLSA level 3" (green). Underneath the badges are links for "Installation", "Documentation", and "Releases". A section titled "What is Minder?" contains a brief description: "Minder is an open source platform that helps development teams and open source communities build more secure software, and prove to others that what they've built is secure. Minder helps project owners proactively manage their security posture by providing a set of checks and policies to minimize risk along the software supply chain, and attest their security practices to downstream consumers." At the bottom of the page is a diagram illustrating the Minder workflow: a blueprint leads to a terminal icon, which then points to a cloud icon containing the Minder logo, which in turn points to a GitHub icon with a checkmark, and finally to a Jenkins icon with a checkmark.

github.com/mindersec/minder

[minder-rules-and-profiles / rule-types / github /](#)

- [trufflehog_github_actiontestdata](#)
- [README.md](#)
- [actions_check_default_permissions.yaml](#)
- [actions_check_pinned_tags.yaml](#)
- [allowed_selected_actions.yaml](#)
- [artifact_attestation_slsa.yaml](#)
- [artifact_signature.yaml](#)

Wait. How about AI?





Nate



@natemcgrady

Subscribe



...

you vibe code malware on purpose

I vibe code malware on accident

we are not the same

@anthropic-ai/clause-code  2.0.27 • Public • Published 2 days ago

 Readme

 Code 

 0 Dependencies

 156 Dependents

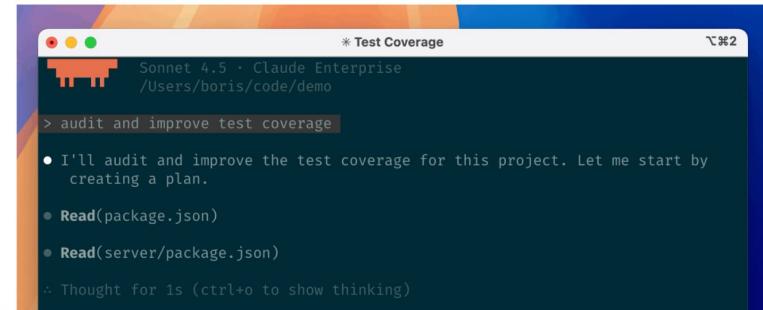
 237 Versions

Claude Code

Node.js 18+ | npm v2.0.27

Claude Code is an agentic coding tool that lives in your terminal, understands your codebase, and helps you code faster by executing routine tasks, explaining complex code, and handling git workflows -- all through natural language commands. Use it in your terminal, IDE, or tag @claude on Github.

Learn more in the [official documentation](#).



Install

 npm i @anthropic-ai/clause-code

Repository

 github.com/anthropics/clause-code

Homepage

 github.com/anthropics/clause-code

Weekly Downloads

5,126,268

Version

2.0.27

License

[SEE LICENSE IN R...](#)

Unpacked Size

78.1 MB

Total Files

49

@chatgptclaude_club/clause-code  0.0.19 • Public • Published 3 hours ago

 Readme

 Code 

 3 Dependencies

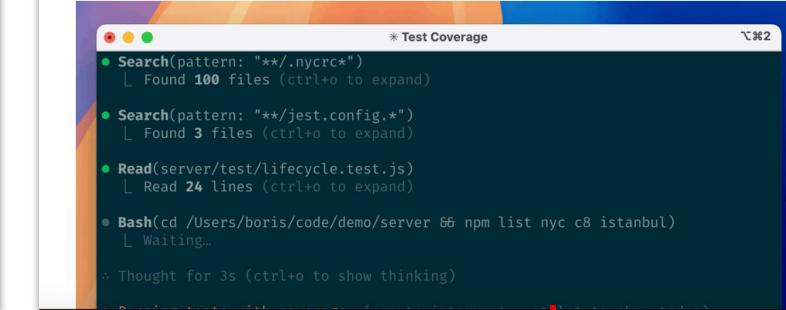
 0 Dependents

Claude Code

Node.js 18+ | npm v2.0.27

Claude Code is an agentic coding tool that lives in your terminal, understands your codebase, and helps you code faster by executing routine tasks, explaining complex code, and handling git workflows -- all through natural language commands. Use it in your terminal, IDE, or tag @claude on Github.

Learn more in the [official documentation](#).



Install

 npm i @chatgptclaude_club/clause-code

Repository

 github.com/anthropics/clause-code

Homepage

 github.com/anthropics/clause-code

Weekly Downloads

207

Version

0.0.19

License

[SEE LICENSE IN R...](#)

Unpacked Size

78.4 MB

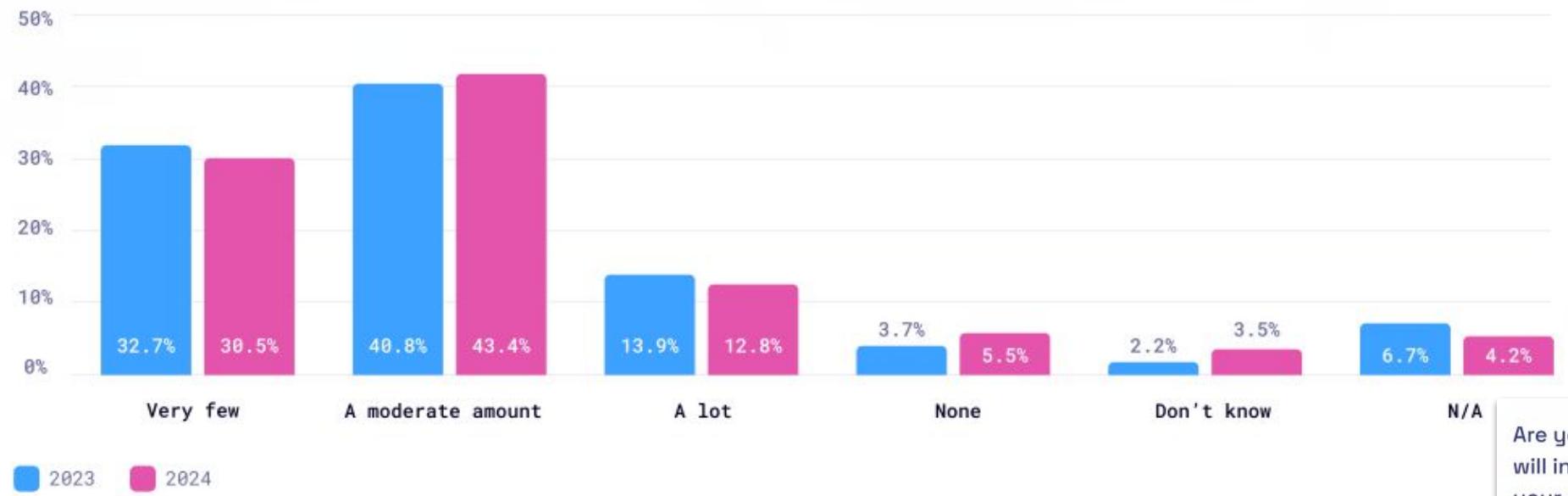
Total Files

52

getsafety.com

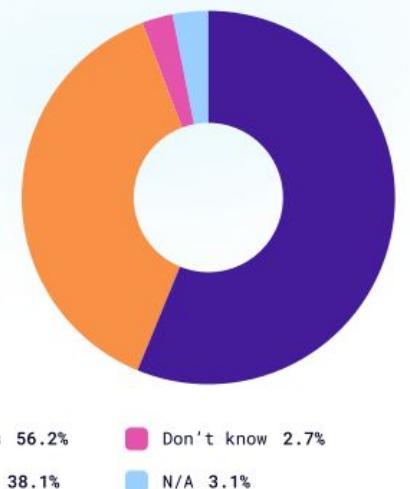
- ▶ This npm package deploys a payload that seeks Claude Code installations
- ▶ Steal Anthropic credentials
- ▶ Setup a proxy between legitimate api.anthropic.com
- ▶ Includes a bidirectional C2 server claude-code.chatgptclaude.club

How many vulnerabilities has AI introduced into your code?



2024 State of open source security report, snyk.io

Are you concerned that using AI coding tools will introduce security vulnerabilities into your applications?



1. TL;DR

Short on time? Here's what really matters:

- **You Are the Developer – AI is the Assistant:** The developer (you) remains in full control of the code, and you are responsible for the security of the system that may be caused by the code. Critically evaluate and edit AI-generated code just as you would code written by others. Never blindly accept suggestions in situations where that could eventually cause harm. [\[ifip2021\]](#) [\[anssibsi2024\]](#)
- **Apply Engineering Best Practices Always:** AI-generated code isn't a shortcut around engineering processes such as static analysis, documentation, and version control discipline. [\[markvero2025a\]](#)
- **Be Security-Conscious:** Assume AI-written code can have bugs or vulnerabilities, because it often does. AI can introduce security issues like using outdated cryptography or outdated dependencies, ignoring error handling, or leaking sensitive data in the suggested code. Make sure dependency suggestions are safe and not pulling in known bad packages. [\[shihchiehdai2025a\]](#), [\[anssibsi2024b\]](#)
- **Guide the AI:** AI is a powerful assistant, but it works best with your guidance. Write clear precise prompts that specify security requirements. Don't hesitate to modify or reject AI outputs. Direct your AI tool to build its own instructions file based on this guide. [\[swaroopdora2025a\]](#) [\[haoyan2025a\]](#)
- **Ask the AI to review and improve its own work.** Once you have some AI-written code, where possible, ask it to review and improve its own work (repeating these steps as necessary). This technique is sometimes called Recursive Criticism and Improvement (RCI) and can be remarkably effective. For instance, "Review your previous answer and find problems with your answer" followed by "Based on the problems you found, improve your answer" for one or more iterations. Encourage the use of tools such as linters, SAST, dependency checkers, etc. through the improvement cycles. [\[catherinetony2024a\]](#)
- **Express your concerns to the AI.** If you have concerns about something AI has generated, express your concerns in detail, and ask it to analyze that code to determine whether or not it's okay. Include relevant information to increase the likelihood of a useful response. Ensure that if something is stated as a fact, it's actually a fact. Review that answer.



New OpenSSF Guidance on AI Code Assistant Instructions

best.openssf.org/Security-Focused-Guide-for-AI-Code-Assistant-Instructions

Key Takeaways

- ▶ **Understand** where the risks come from for you. Repeat.
- ▶ **AI is attack amplifier.** Keep in mind basic security principles. Educate others.
- ▶ **Never stop hacking** - test often. It's open [source] for everybody.
- ▶ **Automate it** with a bunch of decent open source tools available for you.



<https://www.redhat.com/en/resources/product-security-risk-report-2024>

Quiz time – top CWE in the Age of AI

CWE	Percentage of codebases with vulns linked to CWE	Description
CWE-400	70%	 <p>Uncontrolled Resource Consumption: The software does not properly control the allocation and release of system resources, such as memory, CPU time, or disk space. This can lead to DoS attacks.</p>
CWE-200	60%	 <p>Exposure of Sensitive Information to an Unauthorized Actor: The software exposes sensitive information, such as passwords, credit card numbers, or personal data, to unauthorized actors. This can happen through various means, such as insecure storage, unencrypted transmission, or improper access controls.</p>
CWE-79	56%	 <p>Improper Neutralization of Input During Web Page Generation (Cross-Site Scripting): The software does not neutralize or incorrectly neutralizes user-supplied input before including that input in an HTML page. This allows attackers to inject malicious scripts that can steal user data or take control of their browser.</p>
CWE-185	48%	 <p>Incorrect Regular Expression: The software specifies a regular expression in a way that causes data to be improperly matched or compared. Regular expressions should be subjected to thorough testing techniques such as equivalence partitioning, boundary value analysis, and robustness testing.</p>

2025 Open Source Security and Risk Analysis Report,
blackduck.com

SECURE OPEN SOURCE.
IT'S YOURS, NOT
SOMEONE ELSE'S.



[LINKEDIN.COM/IN/ROZHUKOV](https://www.linkedin.com/in/ROZHUKOV)

