ESCAPE RESEARCH

THCON 2026

TOULOUSE HACKING CONVENTION

Vibe Coding at Scale

SYSTEMATIC DISCOVERY OF AUTHORIZATION FAILURES AND DATA EXPOSURE IN AI-GENERATED APPS

35,000	2,000	100+
EXPOSED VULNERABILITIES DISCOVERED	HIGH	CRITICAL

Nohé Hinniger-Foray

linkedin.com/in/nohehf/

- R&D Software Engineer @ **Escape.tech**
- Attack Surface Management (**ASM**) **Lead**
- Automating discovery & pentesting
- ENSEEIHT / Tls-Sec

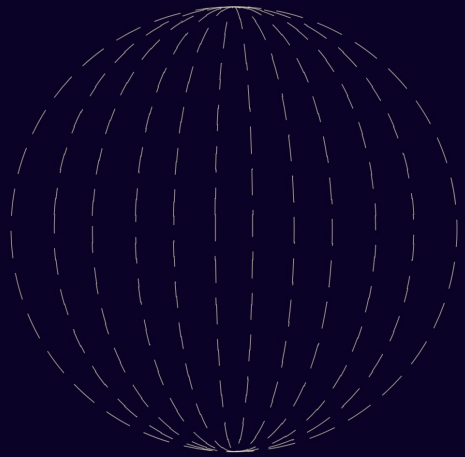
Gabin Fouquet

linkedin.com/in/gabin-fouquet/

- Security Research Product Owner
- In charge of vulnerability quality on Escape products
- ENSEEIHT / Tls-Sec

ESCAPE

ESCAPE

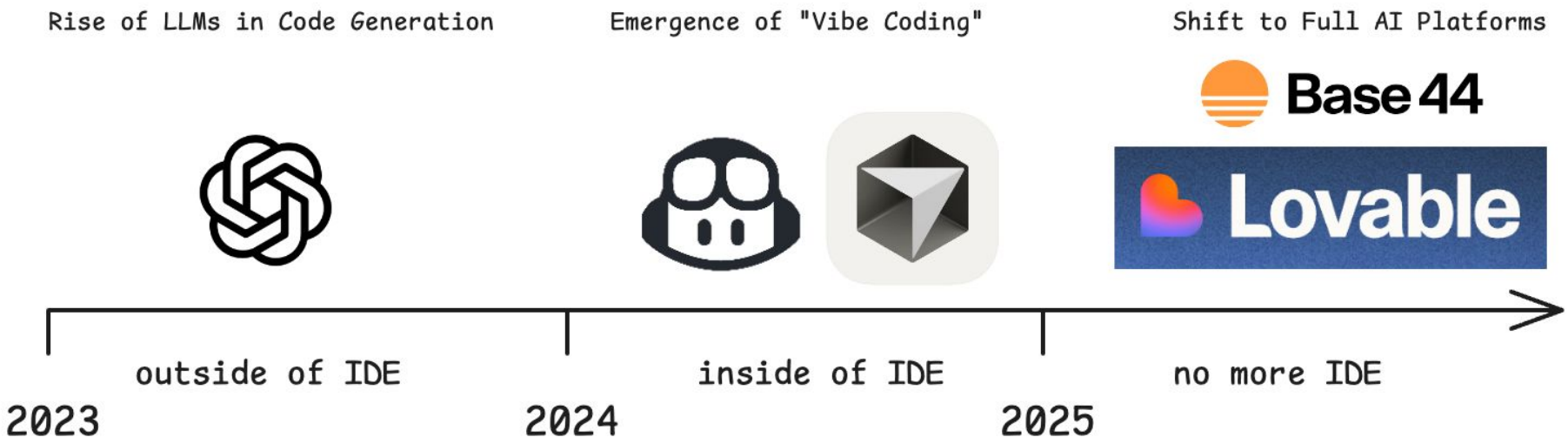


The State of Security of Vibe Coded Apps

How we discovered 2,000 high impact vulnerabilities and sensitive data leaks in apps built with vibe coding platforms

https://escape.tech/state-of-security-of-vibe-coded-apps

From LLMs to “Vibe-Coding”



ESCAPE

Question to the Audience

How many of you use LLMs / AI in your job (or studies)?

ESCAPE

Question to the Audience

How many of you use LLMs / AI in your job (or studies)?

Is it to code ?

The “gut feeling”

- **LLMs ⇒ bugs, hallucinations, …**
- Human supervised LLM produce bugs
  - Fully AI generated code might be worse ?
- **Unreviewed** code, created by **less technical people**
- **Generates** both **backend** and **frontend**, **infrastructure**, handles the **sensitive data**, ...

⇒ There might be security issues there ? 

ESCAPE

Catalyst 1: Increase in Attack surface

15K

exposed **Endpoints** per **organization** on average

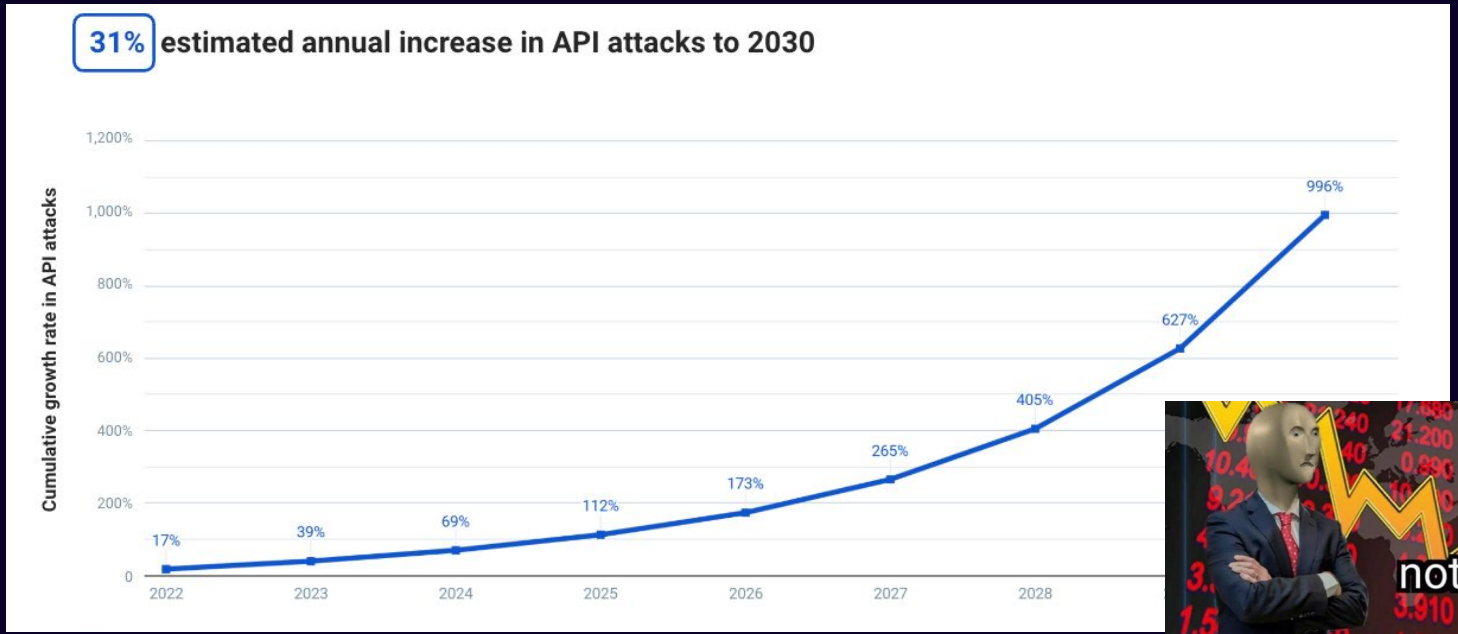
3K

exposed **API Services** on **1 single organization**

**Escape State of the API Security 2025*

ESCAPE

Catalyst 2: Increase in Attacks



+1000% in API Attacks

by the end of the decade

**Leading Digital Transformation: Best Practices for Becoming a Secure API-First Company (Kong)*

ESCAPE

The “Explosive Cocktail”

1. Companies have more Applications that they can track
2. Increase in reported attacks
3. **No more human in the loop**

What could go wrong?

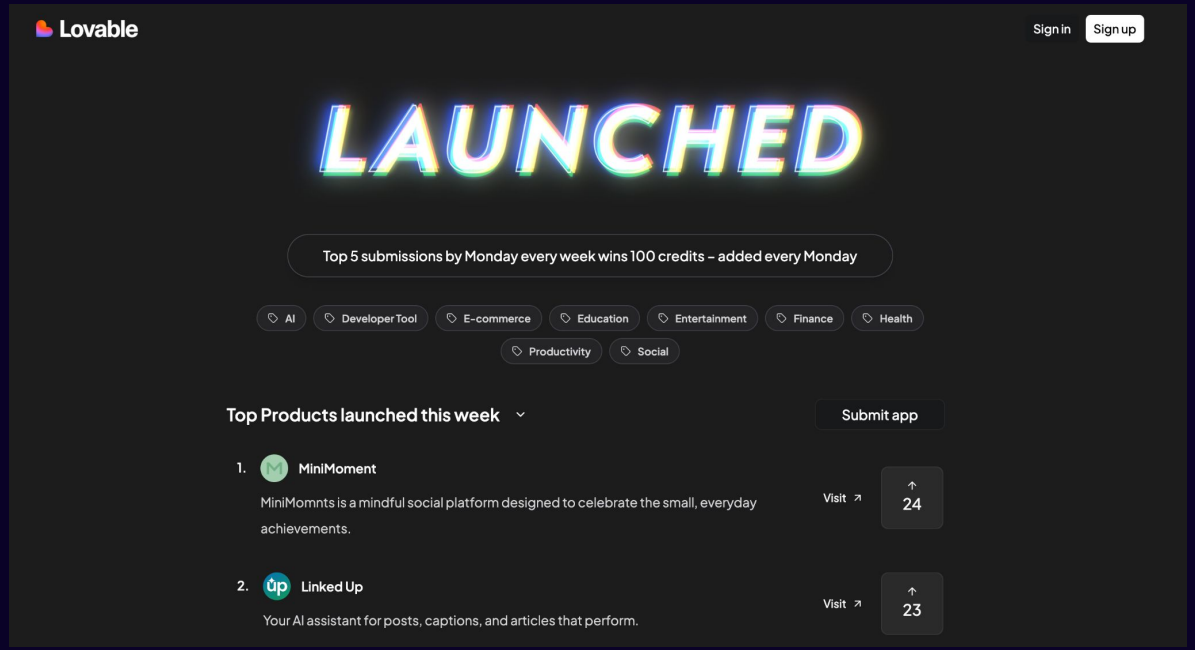
ESCAPE

Methodology: Automated pentesting of “vibe coded” APPs

1.

ESCAPE

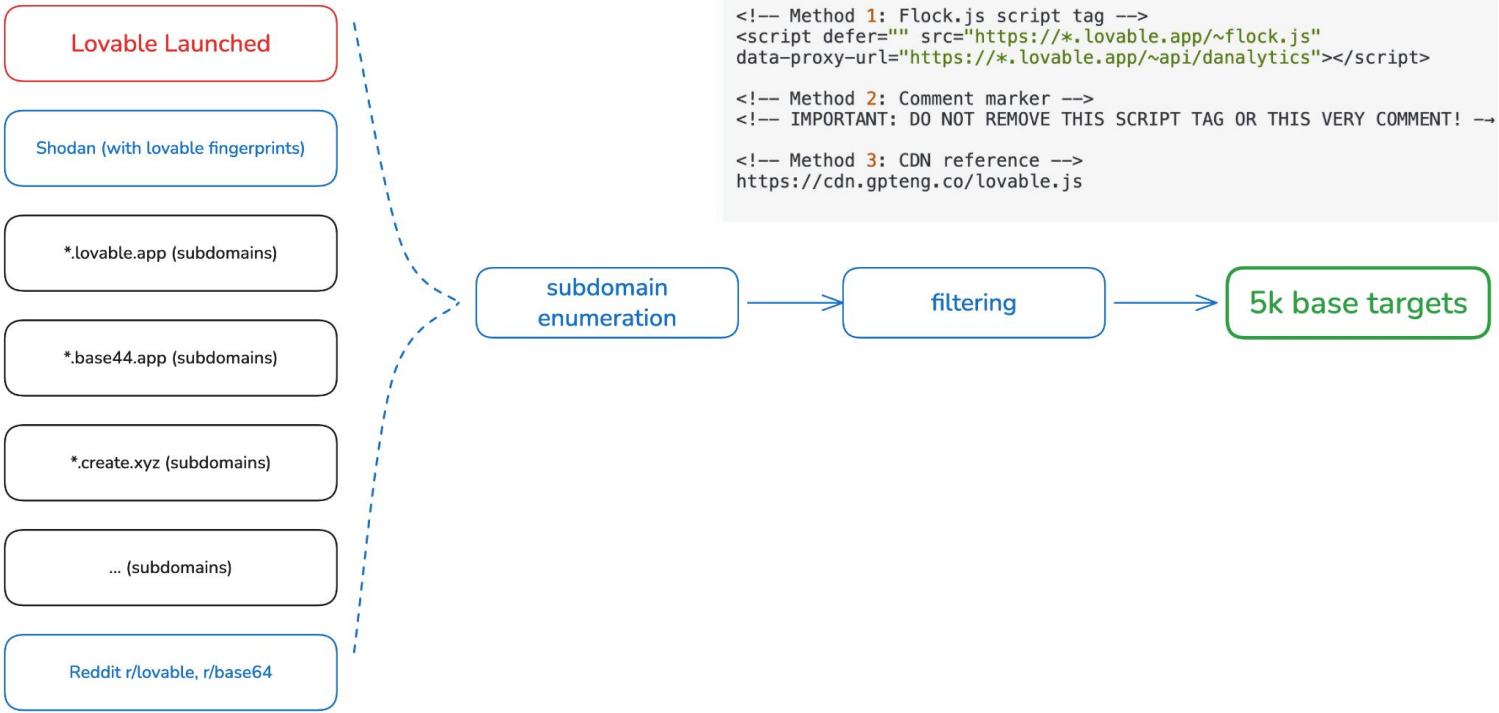
1. Input: Vibe Coded Apps Sources



https://launched.lovable.dev/

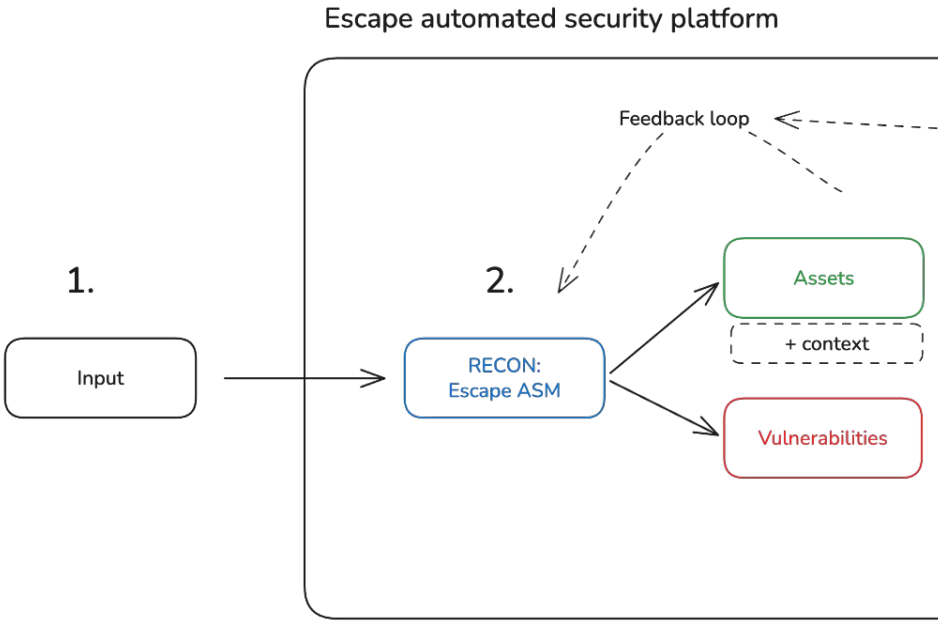
ESCAPE

1. Input: Vibe Coded Apps Sources



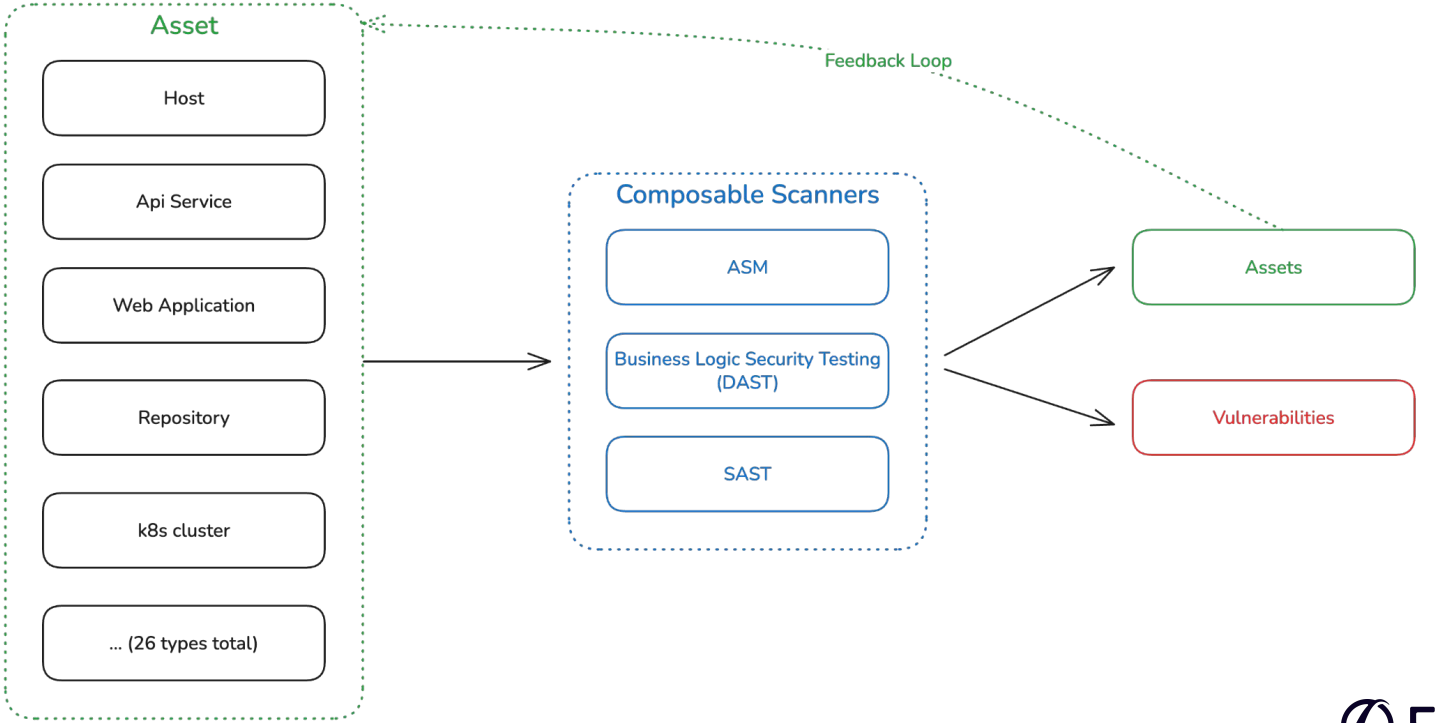
ESCAPE

Methodology: Automated pentesting of “vibe coded” APPs



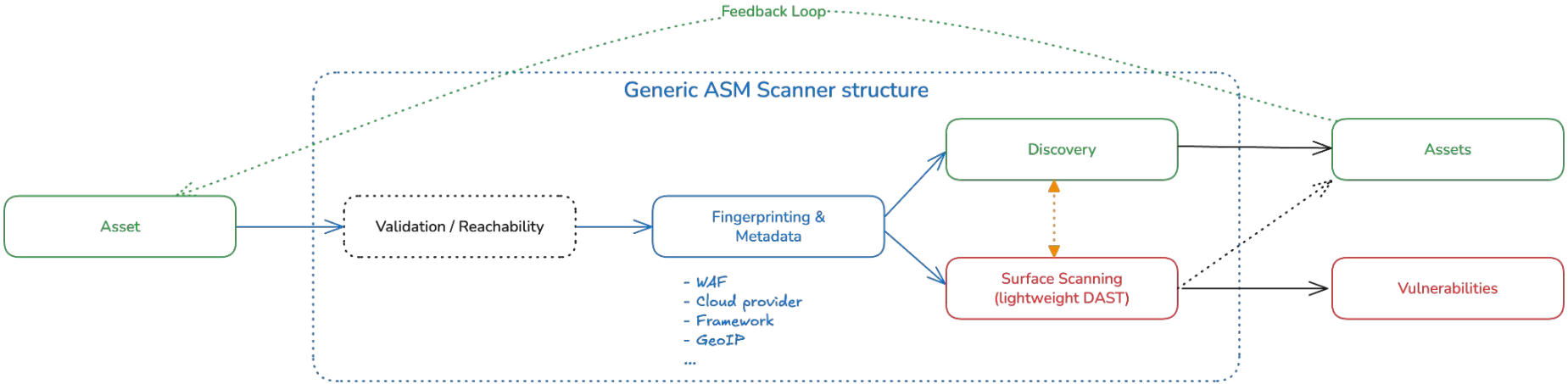
ESCAPE

2. RECON: Internal setup



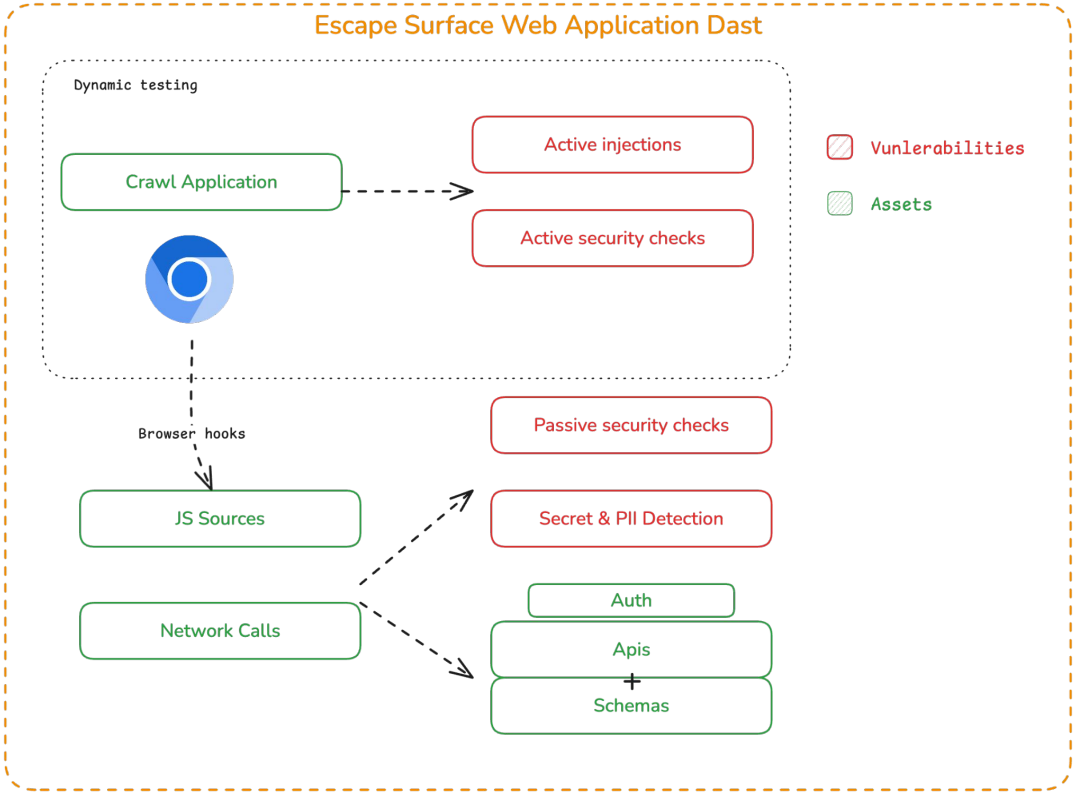
ESCAPE

2. RECON: Anatomy of an ASM scanner



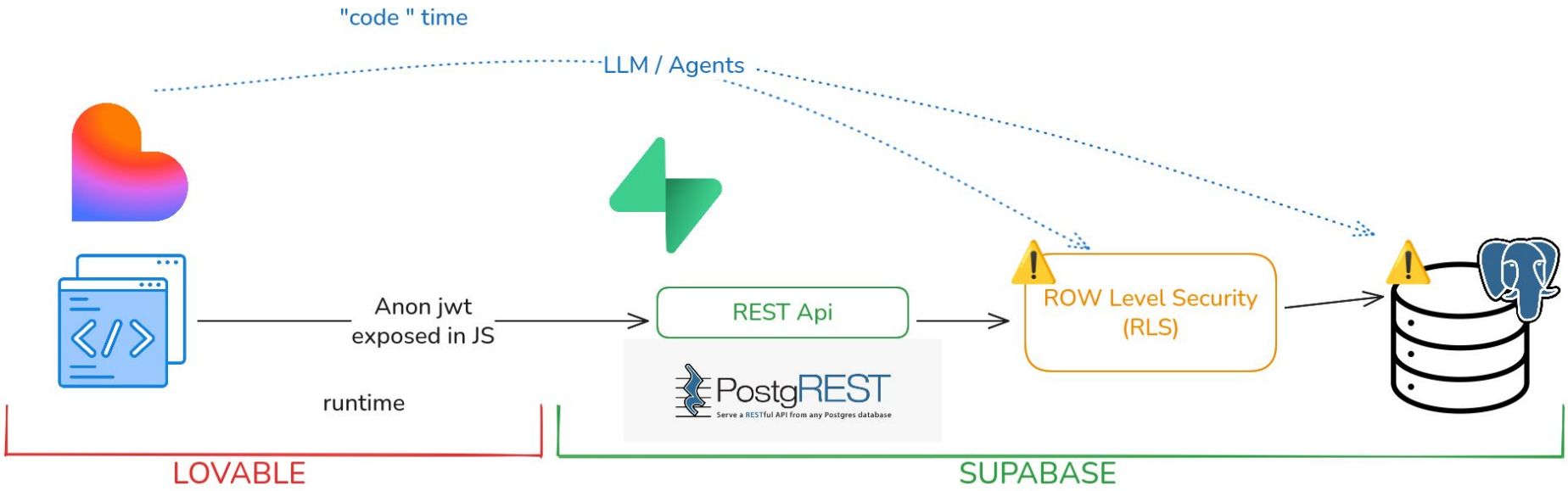
ESCAPE

2. RECON: Anatomy of WEB App Scanning



ESCAPE

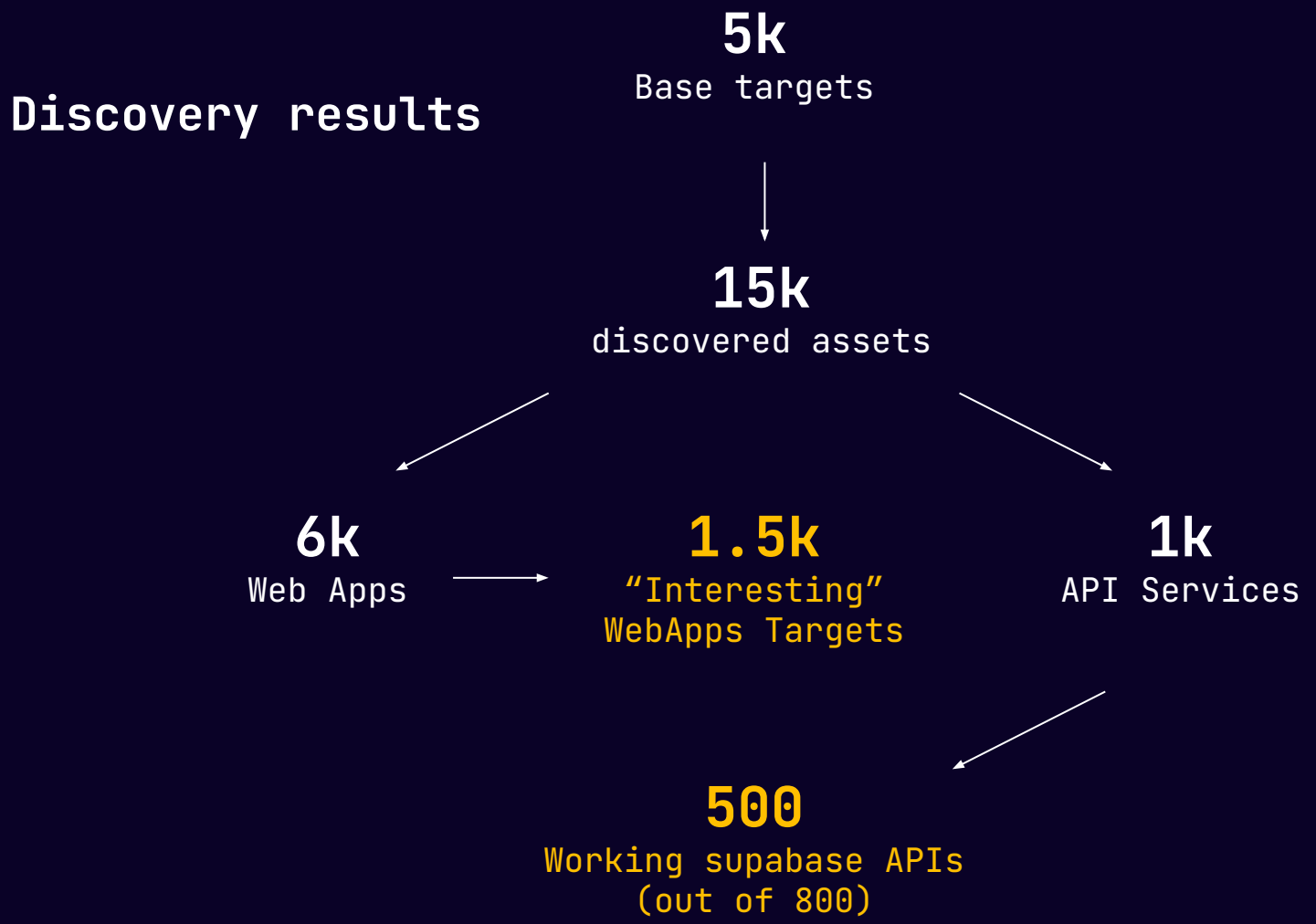
Lovable + Supabase integration for fullstack apps



⚠ CVE-2025-48757

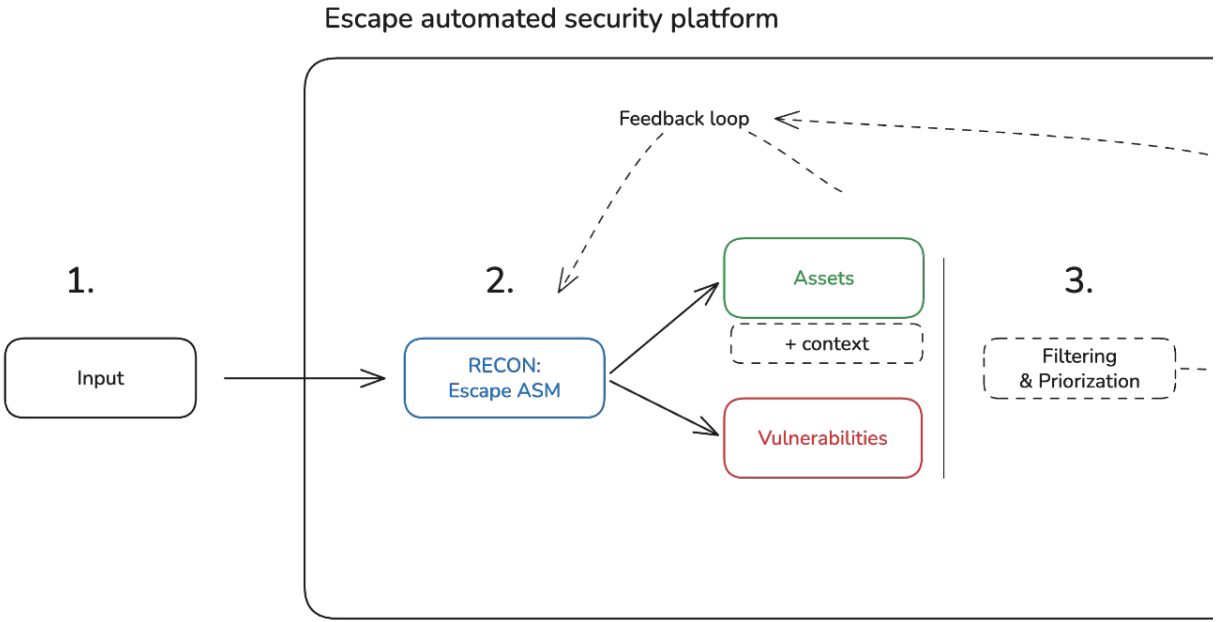
ESCAPE

Discovery results



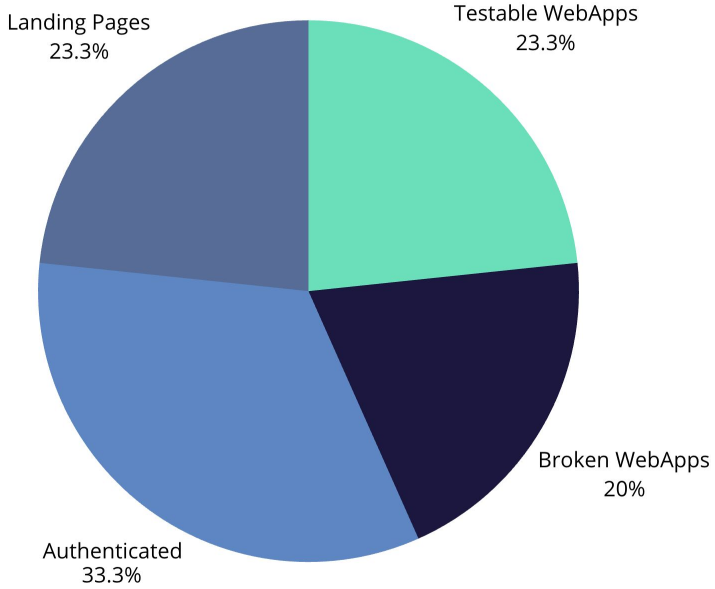
ESCAPE

Methodology: Automated pentesting of “vibe coded” APPs

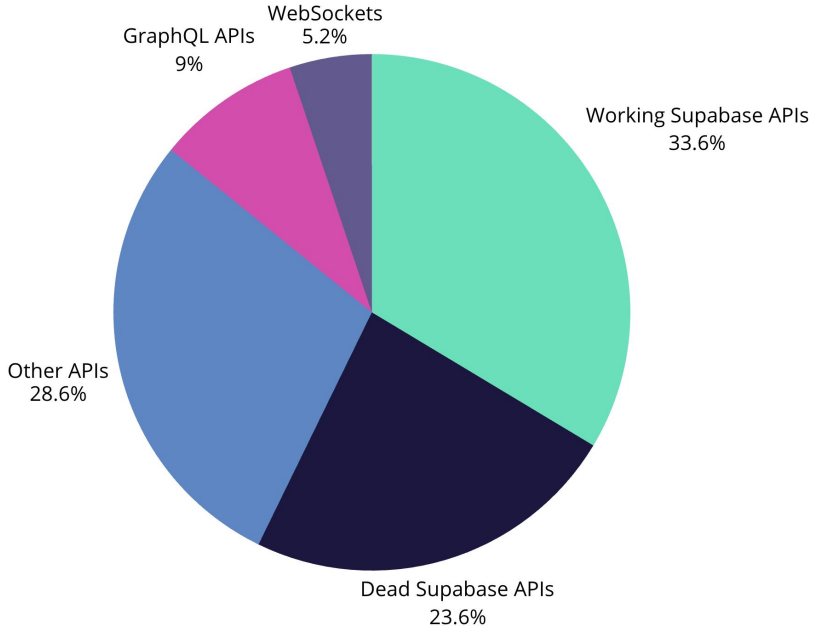


ESCAPE

3. Filtering:
“Vibe-coded” applications are really broken, who could have guessed ?



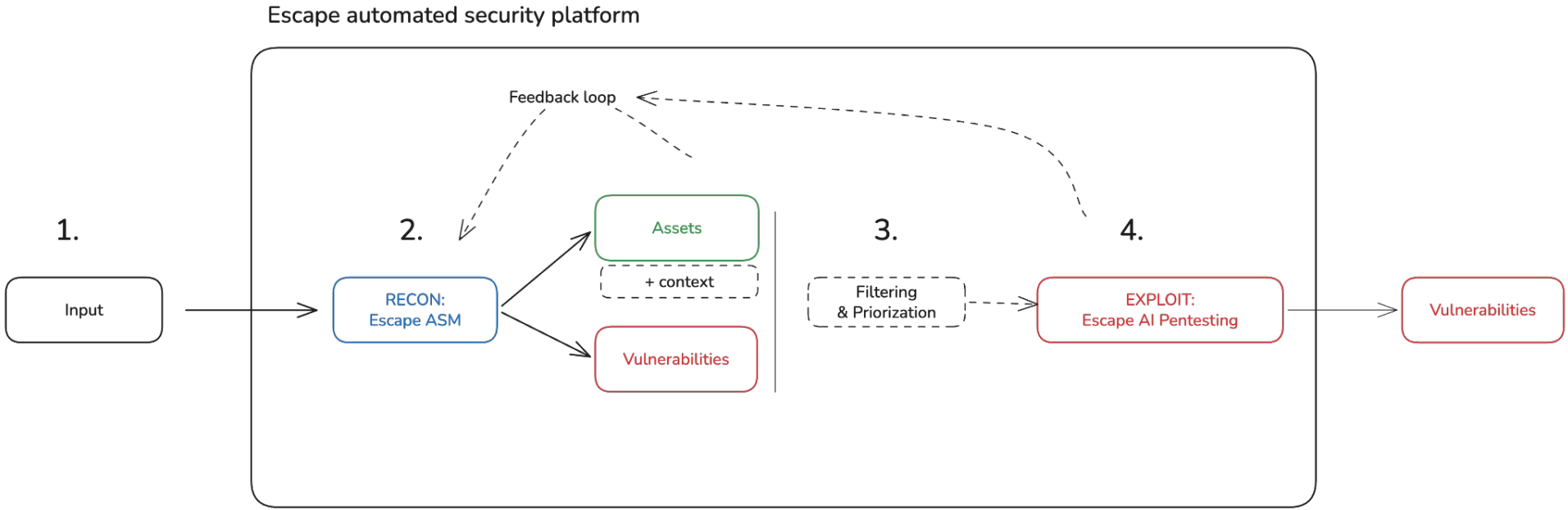
WebApps types (estimate)



API types

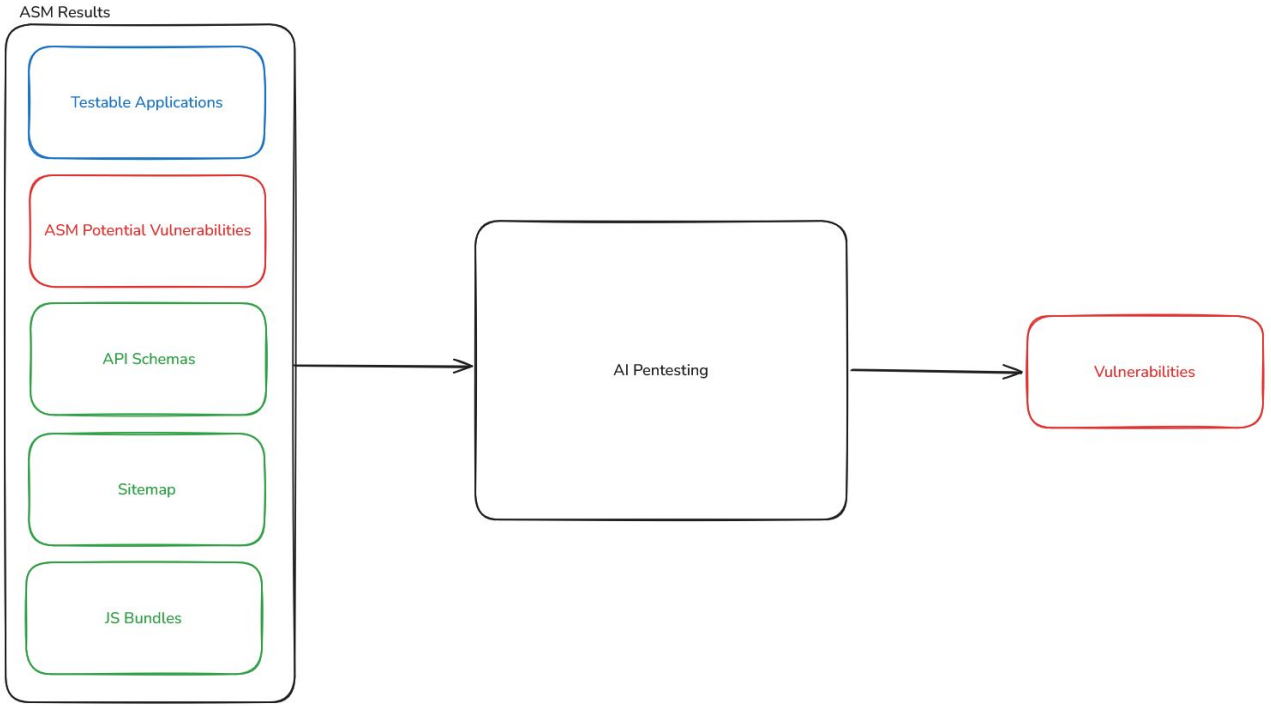
ESCAPE

Methodology: Automated pentesting of “vibe coded” APPs



ESCAPE

4. PENTEST: How are we testing these applications ?

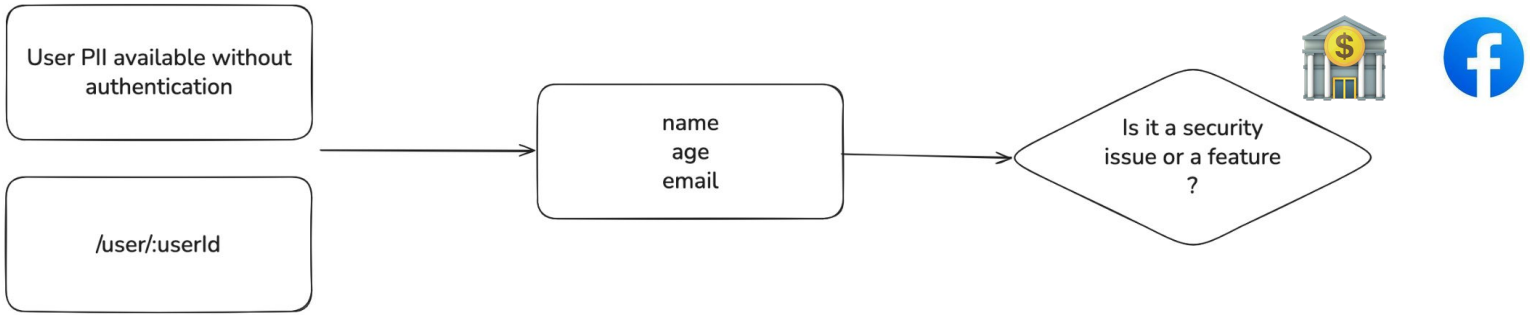


ESCAPE

4. PENTEST: Why did we decide to use AI Pentesting

- Input flexibility to leverage **all available data**
- Detecting **complex business flaws** impossible for DAST without manual config

Example

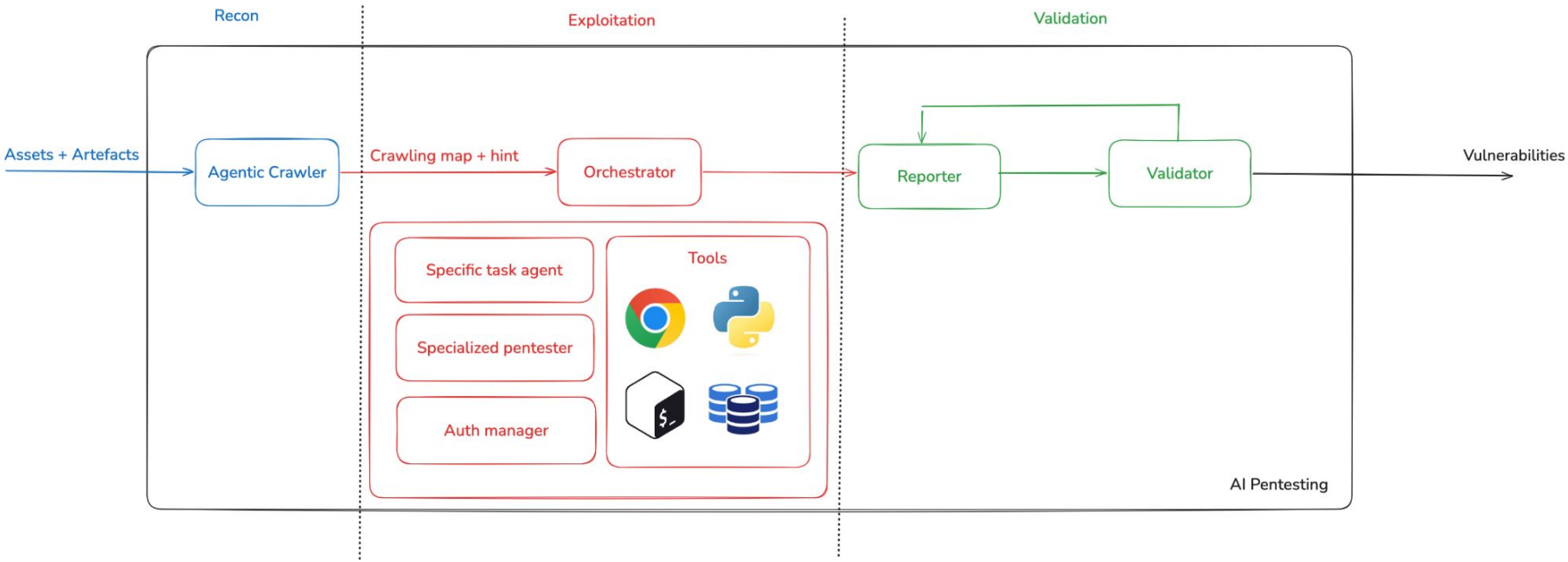


ESCAPE

4. PENTEST: Testing LLMs with LLMs ?

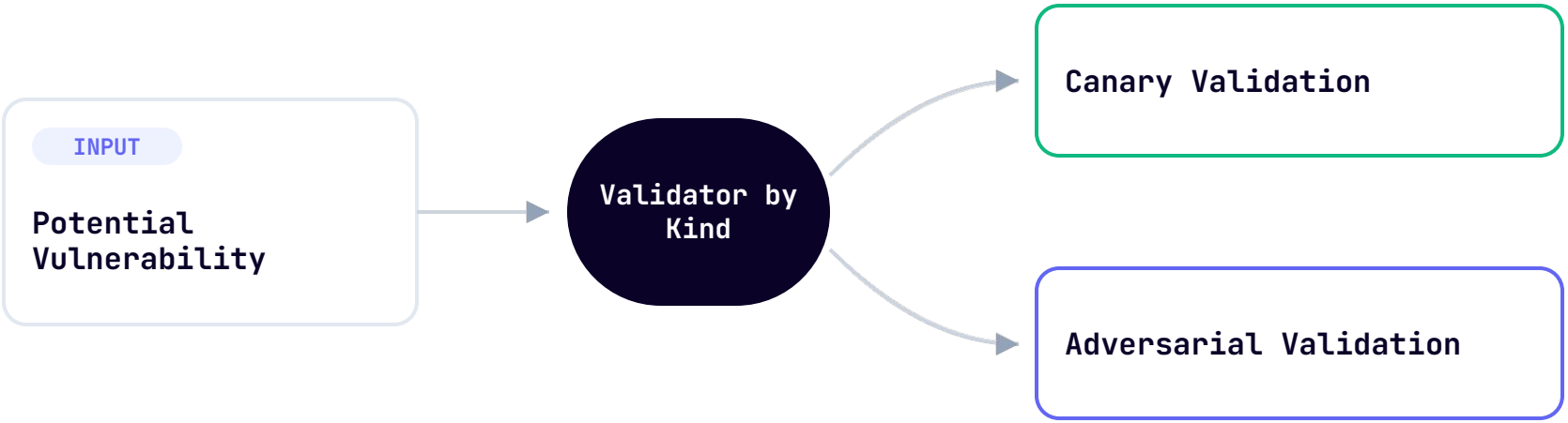
ESCAPE

4. PENTEST: AI Pentesting orchestration



ESCAPE

4. PENTEST: AI Pentesting validation process



When to use Canaries:

Used for deterministic triggers like **XSS**. If the agent triggers the XSS in our provided browser, it is a confirmed high-fidelity finding.

When to use Adversarial:

Crucial for complex logic like **IDOR or BOLA**. Since these lack simple "canary" markers, an adversarial LLM agent validates the unauthorized data access.

ESCAPE

Disclaimer: Challenges & Limitations

- **Broken / Untestable targets**

- **No Authentication**
  - WIP At the time: Automatic Authentication

- **Surface / Production mode scanning**
  - Harmless
  - Disabled checks (hard injections, ...)
  - Disabled state altering operations

"vibe-coded" applications are really broken, trust me

This introduced a strong bias
decreasing the results quantity and severity

ESCAPE

API and SPA Vulnerabilities on Vibe-Coded Apps

34,232
vulnerabilities

~100
critical impact

2k+
high impact

400+
secret leaks

175
instances of PII Leaks
(for a total of 500+ PIIs)

ESCAPE

API and SPA Vulnerabilities on Vibe-Coded Apps

4
SSRFs

2
re-confirmed
CVE-2025-48757

4
confirmed **File Disclosure**

2
0 click account takeover

2 git leaks

12
confirmed **BOLA (IDOR)**

250+
Vulnerable dependencies (with CVEs)

400+
Misconfigured permissions / broken access control

ESCAPE

Security Landscape Shift: From Technical Exploits to Business Logic Failures

Declining: "Old" Vulnerabilities

Traditional injection attacks are becoming rarer due to modern framework protections and automated scanning.

- Cross-Site Scripting (XSS)
- SQL Injection (SQLi)
- Command Injections

Rising: Logic & Authorization

Modern apps suffer from complex authorization flaws that bypass traditional firewalls and security layers.

- BOLA (Broken Object Level Auth)
- IDOR
- Complex Business Logic Flaws

ESCAPE

Deep dive:
Medical platform leaking PII and Passwords of ALL doctors

Cause: RLS bypass

ESCAPE

Deep dive:
0 click Account takeover (of 950 active sessions)

Cause: Broken Custom Auth + BOLA

ESCAPE

Conclusion: Can AI cure AI?

image: xkcd.com

ESCAPE

Conclusion: Can AI cure AI?

Take 1: **AI is here**, whether you like it or not

image: xkcd.com

ESCAPE

Conclusion: Can AI cure AI?

Take 1: **AI is here**, whether you like it or not

Take 2: It’s **increasing attack surface** at an exponential rate

image: xkcd.com

ESCAPE

Conclusion: Can AI cure AI?

Take 1: **AI is here**, whether you like it or not

Take 2: It's **increasing attack surface** at an exponential rate

Take 3: But it makes **security** more **scalable** than ever

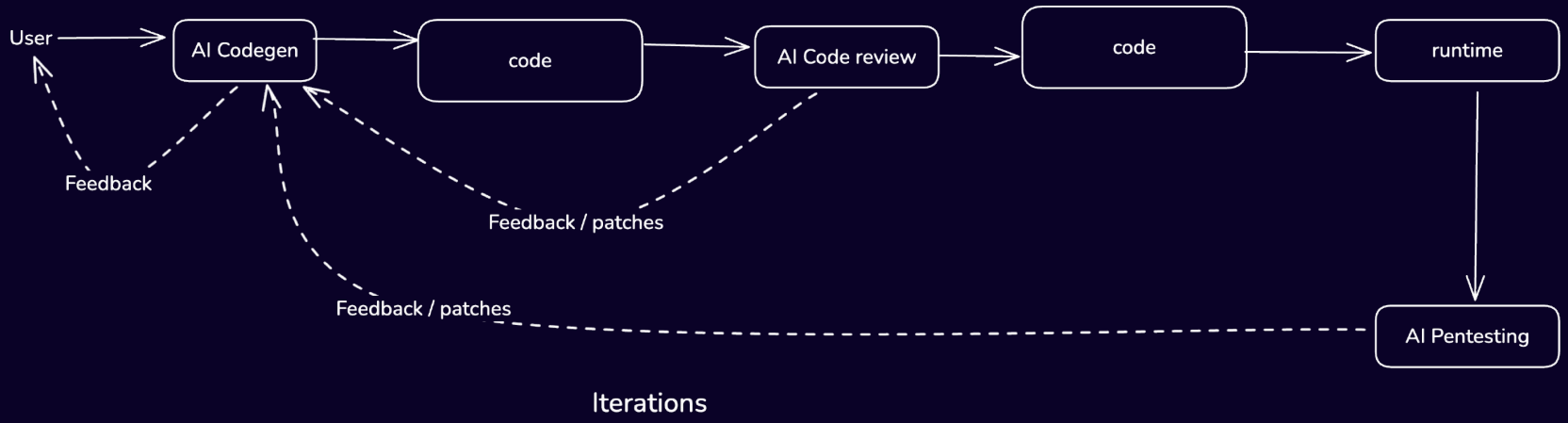


image: xkcd.com

ESCAPE

Opening: The race of AI cyber

Github Enterprise
CVE-2026-3854
by Wiz

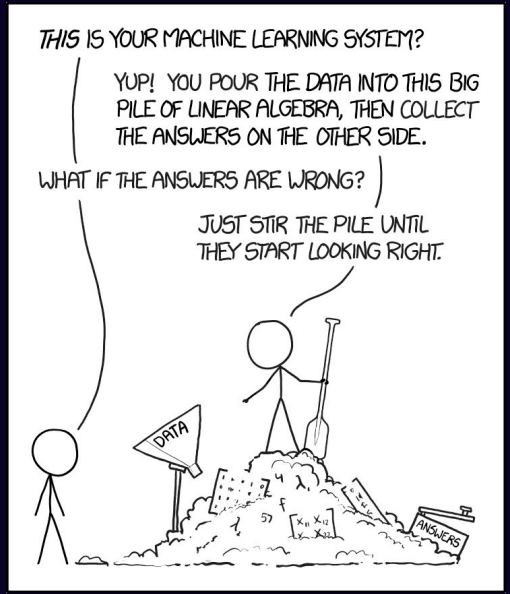
Copy Fail

Copy Fail
CVE-2026-31431
by Xint Code

If we can **automate pentesting** with AI ⇒ **so hackers can.**
AI Assisted CVEs ⇒ new 0 days ?
We have to defend with LLMs.
Who will win this race ?
Will we see major AI caused breaches in the near future ?

ESCAPE

2026 in a nutshell



My job: Writing AIs with AIs to find vulnerabilities in AI generated apps so other AIs won’t pirate it.

ESCAPE

ESCAPE

https://escape.tech/

For teams that are 100x outnumbered

Nohé Hinniger-Foray

R&D Engineer @ escape.tech (YC W23) 💚
ENSEEIHT, INSA, ENAC tls-sec alumni.

Gabin Fouquet

Security Research Product Owner @ Escape

Thank you ! Questions ?

Q: IS THIS LEGAL ?

A: tldr; yes.

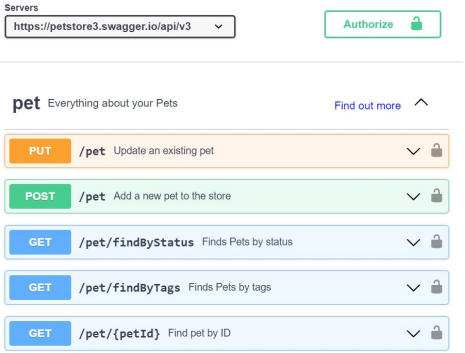
175

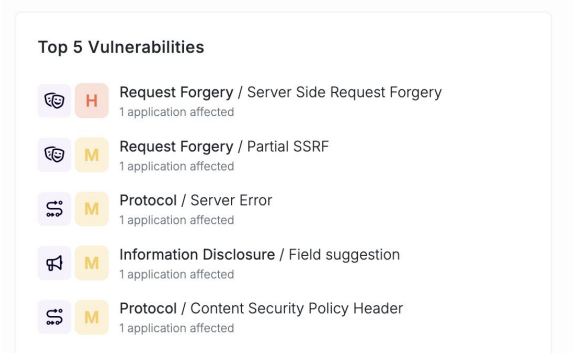
instances of PII Leaks (for a total of 500+ PIIs)

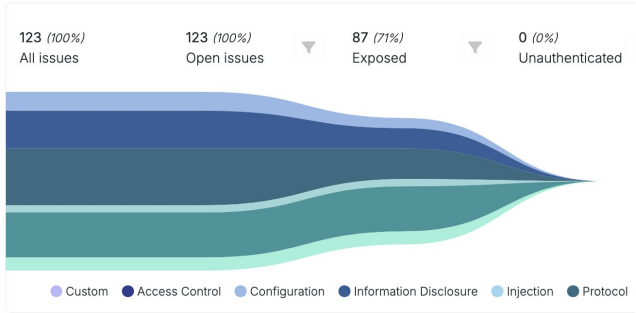
- Medical Records
- IBANs
- Phone numbers
- Emails

ESCAPE

Conclusion: Mitigation strategies for Enterprises







1. Continuous Exposure Monitoring
Know your attack surface in real-time

2. Continuous Security Testing
Find vulnerabilities before others find it

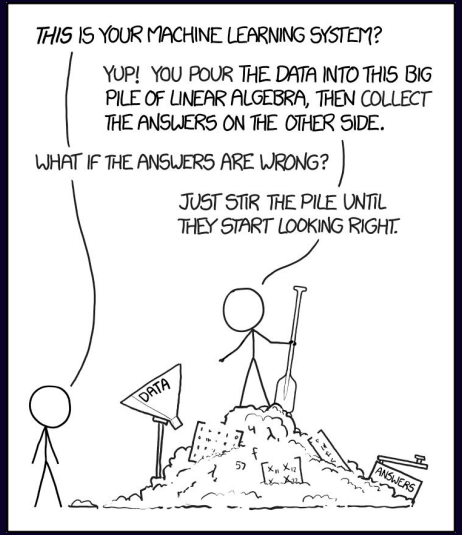
(CI/CD Integration)

3. Security by Design
Collaborate with devs to mitigate risk from conception to deployment

(Shift-left Security)

ESCAPE

Conclusion: Mitigation



Review of AI generated code in critical paths

code with AI, but review your access control layer
or database schema,
I beg you

ESCAPE

image: xkcd.com