

MALWARE IN THE GIST:
How malicious packages bypass existing security tools

Saturday August 9, 11–11.30am

(Creator stage 4 – room 228)



Paul McCarty

SourceCodeRED | SecureStack | GitHax

RESEARCH | SNOWBOARDING | DEVSECOPS | PUNK ROCK

SECURESTACK

NASA JPL

BOEING

U.S.ARMY

Regence

Head of Research @ S>fety



Open-source projects I work on

DevSecOps Playbook:
https://github.com/6mile/DevSecOps-Playbook

Visualizing the software supply chain:
https://github.com/SecureStackCo/visualizing-software-supply-chain

MALOSS:
https://github.com/6mile/MALOSS

commit-audit:
https://github.com/6mile/commit-audit

Software Supply Chain Security Training
https://sourcecodered.com

gimmePATz:
https://github.com/6mile/gimmepatz

Threat modelling software supply chains:
https://github.com/6mile/TVPO



OSV/OpenSSF Malicious Packages:
https://github.com/ossf/malicious-packages

OSC&R
https://github.com/pbom-dev/OSCAR



Ghast:
https://github.com/bin3xish477/ghast



Project Discovery Nuclei Templates:
https://github.com/projectdiscovery/nuclei-templates



(Minimal Viable Secure Product)
https://github.com/vendorsec/mvsp/

Source CodeRED

How does open source malware evade detection?

1. How the malware is delivered
2. Target devs & CI/CD pipelines
3. No good security tools or tracking of threat actors
4. Security teams don’t understand the threat

Source CodeRED

Why do bad guys use NPM?



NPM has systemic flaws

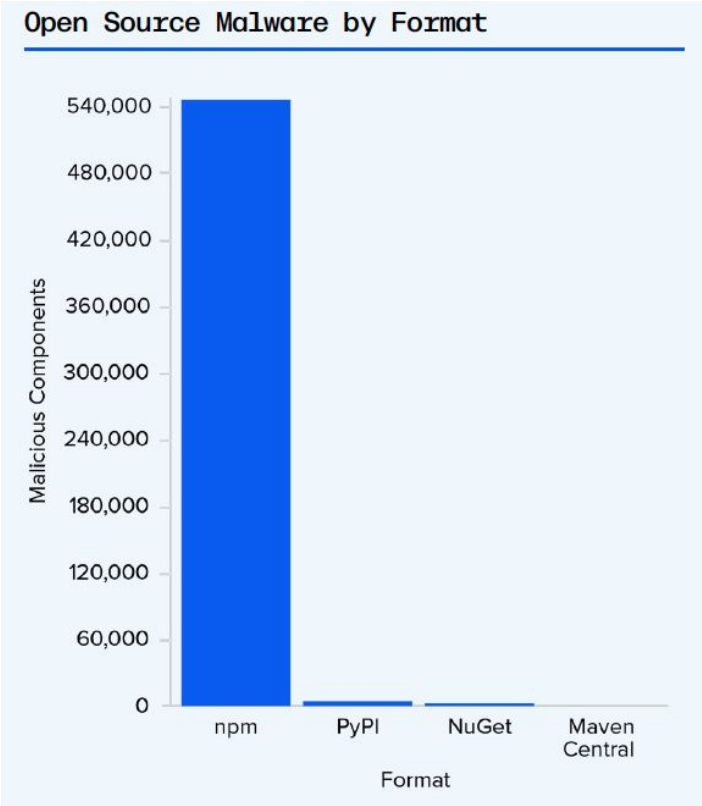
- Install scripts - RCE by design
- Barrier to entry is low
- Anyone can claim a namespace
- Pkg metadata can be faked including downloads, git repo, author & more
- ATO is easier: Doesn't require MFA
- LOOONG time to remove malicious packages from registry
- GHSA/NPM doesn't track malware data: authors, contributors, etc



Source CodeRED

Why npm?

According to a 2024 study by Sonatype, 98.5% of all malicious packages are served by npm

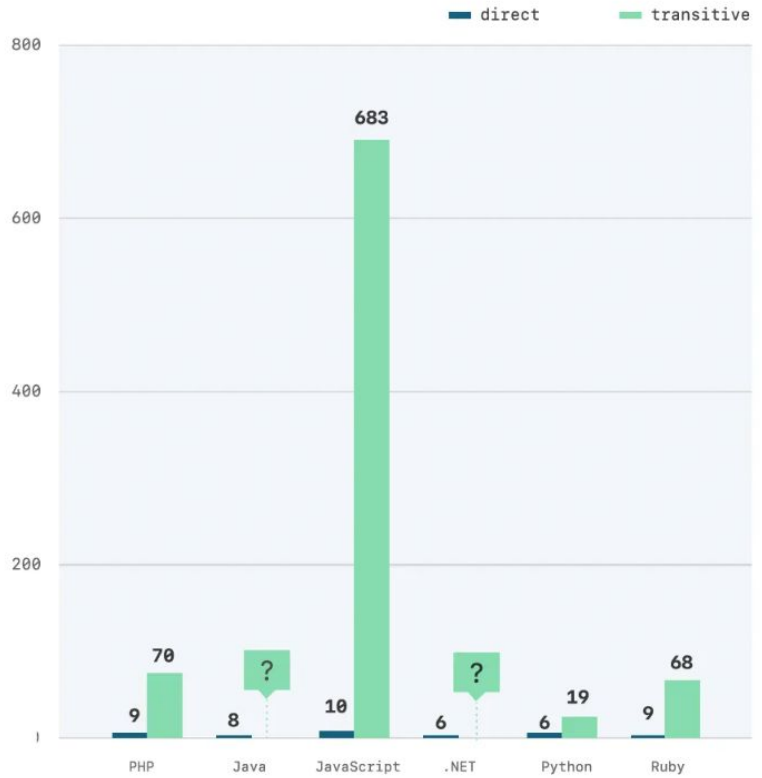


https://www.sonatype.com/resources/whitepapers/2024-open-source-malware-threat-report

Source CodeRED

Why npm?

Javascript, because of its design has 10X the number of dependencies as the next highest language



https://octoverse.github.com/2020/

Source CodeRED

How does open-source malware evade security tools?

Source CodeRED

Open-Source Malware: Interpreted languages

- Typically no binaries in pkgs
- Common techniques from malware analysis like hashing & disassembly are useless
- Highly iterative files so payloads, exfil, etc change constantly
- Sandboxes like any.run, Joe's, Triage don't help



Source CodeRED

Open-Source Malware: Obfuscation

- Obfuscation != def malware
- Often totally legitimate in JS
- Very successful at confusing detection techniques
- Many obfuscation techniques
- Forces human to get involved

```
var _0x2d2e=['log','1034790swXldb','172
64HCqoqK','19784XsRDrH','1297221GzOUfq'
var _0x284b=function(_0x5af078,_0x5d001
return _0x2d2e3a;};var _0x2dda33=_0x284
{try{var _0x427ee4=-parseInt(_0x1bb145(
```

Source CodeRED

Open-Source Malware: Runs in browser

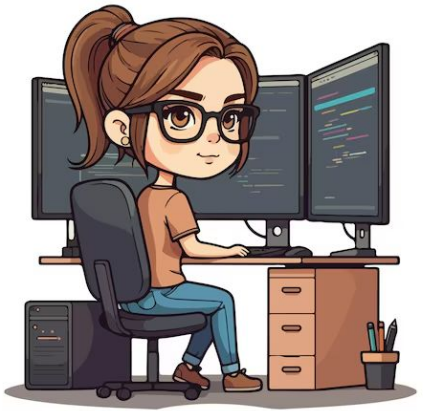
- Javascript runs in browser, does its damage in browser
- Flexible: can target devs, web apps/backend
- Can access tokens, cookies, stored passwords, etc
- Browser extensions



Source CodeRED

Open-Source Malware: Targeting Devs & Dev environments

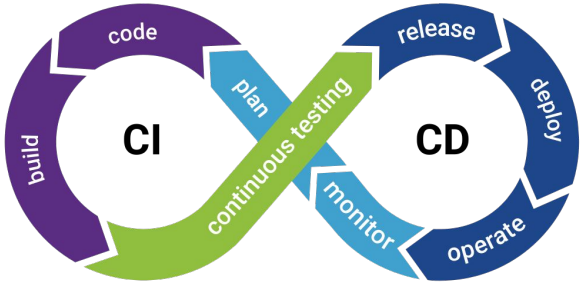
- Devs are targeted for two reasons: To deploy malware, and for account take over
- Security tools devs use like SAST & SCA don’t find malware
- Often no EDR, or exclusions exist
- Lots of browser extensions
- AI tooling - MCP, IDE, Claude, etc



Source CodeRED

Open-Source Malware: Targeting CI/CD pipelines

- No EDR/IDS in pipelines
- Call's libraries and containers with no oversight
- No logging, or logging ignored
- CI is where intellectual property meets lots of access (API keys, tokens, etc)



Source CodeRED

Open-Source Malware: Don’t get detected

- Most popular malicious package databases are GHSA & OSV
- Many security vendors use those two to determine maliciousness
- If you keep your open-source malware out of those two, you are good to go
- If your malware DOES get flagged, yank it quickly





Source CodeRED

How does OSS malware enter your org?

Bad guys target what's hot, so new technologies are always at the top of the list







The open source malware GOAT playbook



- Copy a NPM package that was popular, but out of date and had some vulnerabilities
- Use a non-scoped package, and create scope
- Leave author, GitHub repo, contributors metadata alone
- Fake the download stats
- Don’t add malicious payload until after version 10 of package
- Use separate package for payload
- Don’t use install scripts
- Don’t use binary payload
- If your malware gets flagged by OSV, quickly remove it

Source CodeRED

Threat actor profiles

Source CodeRED

Lazarus Group



- 100% focused on crypto & NK survival
- Targets crypto devs with fake jobs
- Flooding NPM with dependency confusion & typo-squatting, everything crypto/payments
- Payloads seek specific items: wallet files, cookies, exchange tokens
- Hides malicious code via obfuscation or within many JS files
- Detection typically only via NIDS/NDR observing new domain egress

Source CodeRED

Lyrra Boys - Indonesian youth gang



veynlinh

Nava Linh

@veryLinh

- Indonesia: majority package source
- Mimics Chinese/Vietnamese actors
- Chaotic: multiple payload types
- Poor obfuscation techniques
- Poor opsec - Feel safe so cocky
- Will repurpose legitimate web consoles
- Targets payment details across platforms
- Discord-based exfiltration
- Prefers lookalike packages
- Secondary monetization via bot traffic
- Low technical skill: avoids shells/RCE

Source CodeRED

Russian Criminals



- Packages deliver ransomware/infostealers with modified phishing kill chains
- Actors' cybercrime backgrounds manifest in familiar patterns
- Pretty good opsec
- Lots more .ru and .su since sanctions
- Infostealers evolved from binaries to native JavaScript - often obfuscated
- Focus on persistence and staging with eventual human involvement
- Diverse attack patterns converge into common approaches

Source CodeRED

Bug bounty researchers



- Thousands and thousands of these. Sometimes 40-50% of total malicious volume
- Payload is easy to find
- Typically exfils hostname, IP address, username data to OAST or known URLs
- Really, really bad opsec and no obfuscation
- Sometimes crosses a line and drops reverse shell or steals passwords. Oops

Source CodeRED

Learn more about open-source malware

- https://github.com/advisories?query=type%3Amalware
- https://osv.dev/list
- https://github.com/lxyeternal/pypi_malregistry/
- https://github.com/DataDog/malicious-software-packages-dataset/
- https://x.com/npm_malware
- https://github.com/tstromberg/supplychain-attack-data
- https://dasfreak.github.io/Backstabbers-Knife-Collection/
- https://intel.aikido.dev/
- https://vetpkg.dev/mal ?

Source CodeRED



S>fety

getsafety.com

Paul McCarty
6mile@linux.com