Recent posts

Malicious document identified in the conflict Israel & Gaza themed about terrorist organizations related to Iran

Dissecting GobRAT behaviors - Linux malware

Analyzing AsyncRAT distributed in Colombia by Blind Eagle

Using Jlaive to create batch files from .NET assemblies for defense evasion

Executing SCR files using desk.cpl and InstallScreenSaver API Call

DLL Hijacking with DeviceCensus.exe on Windows 11

Dissecting GobRAT behaviors - Linux malware

May 30, 2023 · 20 min read

Jose Luis Sánchez Martínez
Security Researcher

Summary

JPCERT/CC discovered recently attacks that infected routers in Japan with malware around February 2023.

This analysis is focused in the loaders discovered by them

- `60bcd645450e4c846238cf0e7226dc40c84c96eba99f6b2cffcd0ab4a391c8b3`
- `3e44c807a25a56f4068b5b8186eee5002eed6f26d665a8b791c472ad154585d1`

Some of the behaviors identified in the routers, are too generic, which means that can be used in Linux endpoints intrusions too. For that reason, I decided to analyze the samples and contribute to the Sigma community to idenfity new detection opportunities based on the samples and the analysis of JPCERT/CC.

> **INFO**
>
> The objective of the analysis is to provide information about the execution of these loaders and how we can detect them using Sigma Rules

Analysis

During this section, you will see telemetry based on `Sysmon` events and sigma rules that are triggering the behaviors.

Suspicious File Creation in Profile Folder

A file dropped by the loader is created in the `/etc/profile.d` path with the goal of persist. The file created is a `.sh` which will be executed during the start of login shells. That path is used to create persistence in the system by threat actors.

```
<Event>
    <System>
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>11</EventID>
        <Version>2</Version>
        <Level>4</Level>
        <Task>11</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
        <EventRecordID>16919</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
    </System>
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff05-6475-f547-38be1a560000}
        <Data Name="ProcessId">26546</Data>
        <Data Name="Image">/usr/bin/dash</Data>
        <Data Name="TargetFilename">/etc/profile.d/sshdaemon.sh</Data>
        <Data Name="User">-</Data>
    </EventData>
</Event>
```

I've created a sigma rule that can help us to detect this behavior, since it can be used by multiple threat actors.

```
title: Suspicious File Creation in Profile Folder
id: 13f08f54-e705-4498-91fd-cce9d9cee9f1
status: experimental
description: Detects the creation of files under profile.d path
references:
    - https://blogs.jpcert.or.jp/en/2023/05/gobrat.html
    - https://www.virustotal.com/gui/file/60bcd645450e4c846238cf0e7226d
author: Joseliyo Sanchez, @Joseliyo_Jstnk
date: 2023/30/05
logsource:
    product: linux
    category: file_event
detection:
    selection_path:
        TargetFilename|contains: '/etc/profile.d/'
    selection_extension:
        TargetFilename|endswith: '.sh'
    condition: all of selection_*
falsepositives:
    - Legitimate file creation
level: medium
```

Discovery Information About Operantig System in Silent Mode

One of the goals of these threat actors was do different tasks in silent mode. For that purpose, they have used `grep` with the parameter `-iq` to avoid print results in the terminal.

The use of `grep` make sense when you see the code and what they are trying to do. Next image is a piece of code of these loaders.

```
if [ -z $HOSTTYPE ]; then
    if uname -m 2>/dev/null | grep -iq aarch64; then
        HOSTTYPE=aarch64
    elif uname -m 2>/dev/null | grep -iq armv7; then
        HOSTTYPE=arm
    elif uname -m 2>/dev/null | grep -iq x86_64; then
        HOSTTYPE=x86_64
    elif uname -m 2>/dev/null | grep -iq "i386"; then
        HOSTTYPE=i386
    elif uname -m 2>/dev/null | grep -iq "i686"; then
        HOSTTYPE=i686
    elif cat /proc/cpuinfo 2>/dev/null | grep -iq AArch64; then
        HOSTTYPE=arm
    elif cat /proc/cpuinfo 2>/dev/null | grep -iEq "^CPU architecture: 8"; then
        HOSTTYPE=arm
    elif cat /proc/cpuinfo 2>/dev/null | grep -iq ARMv7; then
        HOSTTYPE=arm
    elif cat /proc/cpuinfo 2>/dev/null | grep -iEq "^CPU architecture: 7"; then
        HOSTTYPE=arm
    elif uname -m 2>/dev/null | grep -iq arm; then
        HOSTTYPE=arm
    elif cat /proc/cpuinfo 2>/dev/null | grep -Eq "Intel|AMD"; then
        HOSTTYPE=x86_64
    elif uname -m 2>/dev/null | grep -iq "mips"; then
    HOSTTYPE=mips
    elif uname -m 2>/dev/null | grep -iq "mipsel"; then
```

```
<Event>
    <System>
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>1</EventID>
        <Version>5</Version>
        <Level>4</Level>
        <Task>1</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
        <EventRecordID>16776</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
    </System>
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff05-6475-81cf-b73487550000}
        <Data Name="ProcessId">26562</Data>
        <Data Name="Image">/usr/bin/grep</Data>
        <Data Name="FileVersion">-</Data>
        <Data Name="Description">-</Data>
        <Data Name="Product">-</Data>
        <Data Name="Company">-</Data>
        <Data Name="OriginalFileName">-</Data>
        <Data Name="CommandLine">grep -iq aarch64</Data>
        <Data Name="CurrentDirectory">/opt</Data>
        <Data Name="User">-</Data>
        <Data Name="LogonGuid">{46700b68-0000-0000-ffff-ffffffffffff}</
        <Data Name="LogonId">65535</Data>
        <Data Name="TerminalSessionId">48</Data>
        <Data Name="IntegrityLevel">no level</Data>
        <Data Name="Hashes">-</Data>
        <Data Name="ParentProcessGuid">{46700b68-ff05-6475-f547-38be1a5
        <Data Name="ParentProcessId">26546</Data>
        <Data Name="ParentImage">/usr/bin/dash</Data>
        <Data Name="ParentCommandLine">/bin/sh</Data>
        <Data Name="ParentUser">-</Data>
    </EventData>
</Event>
```

A sigma rule that I thought might help us with this is following all the checks made by this loader is the next one.

```
title: Discovery Information About Operantig System in Silent Mode
id: d27ab432-2199-483f-a297-03633c05bae6
status: experimental
description: Detects the use of grep to identify information about the
references:
    - https://blogs.jpcert.or.jp/en/2023/05/gobrat.html
    - https://www.virustotal.com/gui/file/60bcd645450e4c846238cf0e7226d
author: Joseliyo Sanchez, @Joseliyo_Jstnk
date: 2023/30/05
tags:
    - attack.discovery
    - attack.t1082
logsource:
    category: process_creation
    product: linux
detection:
    selection_process:
        Image|endswith: '/grep'
        CommandLine|contains: '-iq'
    selection_architecture:
        CommandLine|contains:
            - 'aarch64'
            - 'armv7'
            - 'x86_64'
            - 'arm*'
            - 'mips*'
            - 'i386'
            - 'i686'
            - 'AArch64'
            - 'ARMv7'
            - 'arm'
            - 'mips'
            - 'mipsel'
            - 'mips64'
            - 'mips64el'
    condition: all of selection_*
falsepositives:
    - Unknown
level: low
```

Persistence Via Cron Files

There is a `crontab` execution of a crontab file which was dropped. During the execution of `crontab` to load that file, there is a file creation that is interesting to see that there was a crontab execution to persist.

```
<Event>
    <System>
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>11</EventID>
        <Version>2</Version>
        <Level>4</Level>
        <Task>11</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
        <EventRecordID>16873</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
    </System>
```

```
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff13-6475-f59c-766f18560000}
        <Data Name="ProcessId">26604</Data>
        <Data Name="Image">/usr/bin/crontab</Data>
        <Data Name="TargetFilename">/var/spool/cron/crontabs/tmp.0iiGrT
        <Data Name="CreationUtcTime">2023-05-30 13:50:11.585</Data>
        <Data Name="User">-</Data>
    </EventData>
</Event>
```

It was used to create persistence in the system. In order to detect that behavior there is a public sigma rule that can help us.

```
title: Persistence Via Cron Files
id: 6c4e2f43-d94d-4ead-b64d-97e53fa2bd05
status: test
description: Detects creation of cron file or files in Cron directories
references:
    - https://github.com/microsoft/MSTIC-Sysmon/blob/f1477c0512b0747c14
author: Roberto Rodriguez (Cyb3rWard0g), OTR (Open Threat Research), MS
date: 2021/10/15
modified: 2022/12/31
tags:
    - attack.persistence
    - attack.t1053.003
logsource:
    product: linux
    category: file_event
detection:
    selection1:
        TargetFilename|startswith:
            - '/etc/cron.d/'
            - '/etc/cron.daily/'
            - '/etc/cron.hourly/'
            - '/etc/cron.monthly/'
            - '/etc/cron.weekly/'
            - '/var/spool/cron/crontabs/'
    selection2:
        TargetFilename|contains:
            - '/etc/cron.allow'
            - '/etc/cron.deny'
            - '/etc/crontab'
    condition: 1 of selection*
falsepositives:
    - Any legitimate cron file.
level: medium
```

Disable Or Stop Services

Stop the firewall service is one of the goals during the execution. They have used `disable` and `stop` options.

```
<Event>
    <System>
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>1</EventID>
        <Version>5</Version>
        <Level>4</Level>
        <Task>1</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
```

```
        <EventRecordID>16755</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
    </System>
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff05-6475-9d3b-864902560000}
        <Data Name="ProcessId">26548</Data>
        <Data Name="Image">/usr/bin/systemctl</Data>
        <Data Name="FileVersion">-</Data>
        <Data Name="Description">-</Data>
        <Data Name="Product">-</Data>
        <Data Name="Company">-</Data>
        <Data Name="OriginalFileName">-</Data>
        <Data Name="CommandLine">systemctl stop firewalld.service</Data
        <Data Name="CurrentDirectory">/opt</Data>
        <Data Name="User">-</Data>
        <Data Name="LogonGuid">{46700b68-0000-0000-ffff-ffffffffffff}</
        <Data Name="LogonId">65535</Data>
        <Data Name="TerminalSessionId">48</Data>
        <Data Name="IntegrityLevel">no level</Data>
        <Data Name="Hashes">-</Data>
        <Data Name="ParentProcessGuid">{46700b68-ff05-6475-f547-38be1a5
        <Data Name="ParentProcessId">26546</Data>
        <Data Name="ParentImage">/usr/bin/dash</Data>
        <Data Name="ParentCommandLine">/bin/sh</Data>
        <Data Name="ParentUser">-</Data>
    </EventData>
</Event>
```

Whether the option is to `disable` or `stop`, the following public sigma rule can help you detect this behavior.

```
title: Disable Or Stop Services
id: de25eeb8-3655-4643-ac3a-b662d3f26b6b
status: experimental
description: Detects the usage of utilities such as 'systemctl', 'servi
references:
    - https://www.trendmicro.com/pl_pl/research/20/i/the-evolution-of-m
author: Nasreddine Bencherchali (Nextron Systems)
date: 2022/09/15
tags:
    - attack.defense_evasion
logsource:
    category: process_creation
    product: linux
detection:
    selection:
        Image|endswith:
            - '/service'
            - '/systemctl'
            - '/chkconfig'
        CommandLine|contains:
            - 'stop'
            - 'disable'
    condition: selection
falsepositives:
    - Legitimate administration activities
level: medium
```

Crontab Enumeration

The use of `crontab -l` is not malicious for itself, however is interesting to identify this behavior. They used it during the infection.

```
<Event>
    <System>
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>1</EventID>
        <Version>5</Version>
        <Level>4</Level>
        <Task>1</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
        <EventRecordID>16863</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
    </System>
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff13-6475-f53c-414b9d550000}
        <Data Name="ProcessId">26600</Data>
        <Data Name="Image">/usr/bin/crontab</Data>
        <Data Name="FileVersion">-</Data>
        <Data Name="Description">-</Data>
        <Data Name="Product">-</Data>
        <Data Name="Company">-</Data>
        <Data Name="OriginalFileName">-</Data>
        <Data Name="CommandLine">crontab -l</Data>
        <Data Name="CurrentDirectory">/opt</Data>
        <Data Name="User">-</Data>
        <Data Name="LogonGuid">{46700b68-0000-0000-ffff-ffffffffffff}</
        <Data Name="LogonId">65535</Data>
        <Data Name="TerminalSessionId">48</Data>
        <Data Name="IntegrityLevel">no level</Data>
        <Data Name="Hashes">-</Data>
        <Data Name="ParentProcessGuid">{46700b68-ff05-6475-f547-38be1a5
        <Data Name="ParentProcessId">26546</Data>
        <Data Name="ParentImage">/usr/bin/dash</Data>
        <Data Name="ParentCommandLine">/bin/sh</Data>
        <Data Name="ParentUser">-</Data>
    </EventData>
</Event>
```

And for that reason, I've created the next rule for crontab enumeration.

```
title: Crontab Enumeration
id: 403ed92c-b7ec-4edd-9947-5b535ee12d46
status: experimental
description: Detects usage of crontab to list the tasks of the user
references:
    - https://blogs.jpcert.or.jp/en/2023/05/gobrat.html
    - https://www.virustotal.com/gui/file/60bcd645450e4c846238cf0e7226d
author: Joseliyo Sanchez, @Joseliyo_Jstnk
date: 2023/30/05
tags:
    - attack.discovery
    - attack.t1007
logsource:
    product: linux
    category: process_creation
detection:
    selection:
        Image|endswith: '/crontab'
        CommandLine|contains: '-l'
```

```
    condition: selection
falsepositives:
    - Legitimate use of crontab
level: low
```

Chmod Suspicious Directory

The payloads dropped on the system, which are located in the `/tmp/` directory, must have the appropriate permissions to be executed, for that reason `chmod` is used.

This command is executed 4 times for the payloads dropped.

- /tmp/env/.qnapd/frpc
- /tmp/env/.qnapd/apached
- /tmp/env/.qnapd/sshd.sh
- /tmp/env/.qnapd/waitd.sh

```
<Event>
    <System>
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>1</EventID>
        <Version>5</Version>
        <Level>4</Level>
        <Task>1</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
        <EventRecordID>16856</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
    </System>
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff13-6475-5109-ead5eb550000}
        <Data Name="ProcessId">26597</Data>
        <Data Name="Image">/usr/bin/chmod</Data>
        <Data Name="FileVersion">-</Data>
        <Data Name="Description">-</Data>
        <Data Name="Product">-</Data>
        <Data Name="Company">-</Data>
        <Data Name="OriginalFileName">-</Data>
        <Data Name="CommandLine">chmod +x /tmp/env/.qnapd/apached</Data
        <Data Name="CurrentDirectory">/opt</Data>
        <Data Name="User">-</Data>
        <Data Name="LogonGuid">{46700b68-0000-0000-ffff-ffffffffffff}</
        <Data Name="LogonId">65535</Data>
        <Data Name="TerminalSessionId">48</Data>
        <Data Name="IntegrityLevel">no level</Data>
        <Data Name="Hashes">-</Data>
        <Data Name="ParentProcessGuid">{46700b68-ff05-6475-f547-38be1a5
        <Data Name="ParentProcessId">26546</Data>
        <Data Name="ParentImage">/usr/bin/dash</Data>
        <Data Name="ParentCommandLine">/bin/sh</Data>
        <Data Name="ParentUser">-</Data>
    </EventData>
</Event>
```

Since this behavior is suspicious, there was also a sigma rule to detect this behavior and help us in this case.

```
title: Chmod Suspicious Directory
id: 6419afd1-3742-47a5-a7e6-b50386cd15f8
status: experimental
description: Detects chmod targeting files in abnormal directory paths.
references:
    - https://www.intezer.com/blog/malware-analysis/new-backdoor-sysjok
    - https://github.com/redcanaryco/atomic-red-team/blob/f339e7da7d05f
author: 'Christopher Peacock @SecurePeacock, SCYTHE @scythe_io'
date: 2022/06/03
tags:
    - attack.defense_evasion
    - attack.t1222.002
logsource:
    product: linux
    category: process_creation
detection:
    selection:
        Image|endswith: '/chmod'
        CommandLine|contains:
            - '/tmp/'
            - '/.Library/'
            - '/etc/'
            - '/opt/'
    condition: selection
falsepositives:
    - Admin changing file permissions.
level: medium
```

Disabling Security Tools

Same behavior as we saw in the `Disable Or Stop Services` section. However, there is another sigma rule that is detecting it.

```
<Event>
    <System>
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>1</EventID>
        <Version>5</Version>
        <Level>4</Level>
        <Task>1</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
        <EventRecordID>16755</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
    </System>
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff05-6475-9d3b-864902560000}
        <Data Name="ProcessId">26548</Data>
        <Data Name="Image">/usr/bin/systemctl</Data>
        <Data Name="FileVersion">-</Data>
        <Data Name="Description">-</Data>
        <Data Name="Product">-</Data>
        <Data Name="Company">-</Data>
        <Data Name="OriginalFileName">-</Data>
        <Data Name="CommandLine">systemctl stop firewalld.service</Data
        <Data Name="CurrentDirectory">/opt</Data>
        <Data Name="User">-</Data>
        <Data Name="LogonGuid">{46700b68-0000-0000-ffff-ffffffffffff}</
        <Data Name="LogonId">65535</Data>
        <Data Name="TerminalSessionId">48</Data>
```

```
        <Data Name="IntegrityLevel">no level</Data>
        <Data Name="Hashes">-</Data>
        <Data Name="ParentProcessGuid">{46700b68-ff05-6475-f547-38be1a5
        <Data Name="ParentProcessId">26546</Data>
        <Data Name="ParentImage">/usr/bin/dash</Data>
        <Data Name="ParentCommandLine">/bin/sh</Data>
        <Data Name="ParentUser">-</Data>
    </EventData>
</Event>
```

Public sigma that can help to detect it.

```
title: Disabling Security Tools
id: e3a8a052-111f-4606-9aee-f28ebeb76776
status: test
description: Detects disabling security tools
references:
    - https://github.com/redcanaryco/atomic-red-team/blob/f339e7da7d05f
author: Ömer Günal, Alejandro Ortuno, oscd.community
date: 2020/06/17
modified: 2022/10/09
tags:
    - attack.defense_evasion
    - attack.t1562.004
logsource:
    category: process_creation
    product: linux
detection:
    selection_iptables_1:
        Image|endswith: '/service'
        CommandLine|contains|all:
            - 'iptables'
            - 'stop'
    selection_iptables_2:
        Image|endswith: '/service'
        CommandLine|contains|all:
            - 'ip6tables'
            - 'stop'
    selection_iptables_3:
        Image|endswith: '/chkconfig'
        CommandLine|contains|all:
            - 'iptables'
            - 'stop'
    selection_iptables_4:
        Image|endswith: '/chkconfig'
        CommandLine|contains|all:
            - 'ip6tables'
            - 'stop'
    selection_firewall_1:
        Image|endswith: '/systemctl'
        CommandLine|contains|all:
            - 'firewalld'
            - 'stop'
    selection_firewall_2:
        Image|endswith: '/systemctl'
        CommandLine|contains|all:
            - 'firewalld'
            - 'disable'
    selection_carbonblack_1:
        Image|endswith: '/service'
        CommandLine|contains|all:
            - 'cbdaemon'
            - 'stop'
    selection_carbonblack_2:
        Image|endswith: '/chkconfig'
        CommandLine|contains|all:
```

```
            - 'cbdaemon'
            - 'off'
    selection_carbonblack_3:
        Image|endswith: '/systemctl'
        CommandLine|contains|all:
            - 'cbdaemon'
            - 'stop'
    selection_carbonblack_4:
        Image|endswith: '/systemctl'
        CommandLine|contains|all:
            - 'cbdaemon'
            - 'disable'
    selection_selinux:
        Image|endswith: '/setenforce'
        CommandLine|contains: '0'
    selection_crowdstrike_1:
        Image|endswith: '/systemctl'
        CommandLine|contains|all:
            - 'stop'
            - 'falcon-sensor'
    selection_crowdstrike_2:
        Image|endswith: '/systemctl'
        CommandLine|contains|all:
            - 'disable'
            - 'falcon-sensor'
    condition: 1 of selection*
falsepositives:
    - Legitimate administration activities
level: medium
```

Nohup Execution

`nohup` can be used to execute binaries in the system. In fact, it was used to execute a fake `apached` service which is the GobRAT payload.

```
<Event>
    <System>
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>1</EventID>
        <Version>5</Version>
        <Level>4</Level>
        <Task>1</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
        <EventRecordID>16945</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
    </System>
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff13-6475-5131-7d1f45560000}
        <Data Name="ProcessId">26638</Data>
        <Data Name="Image">/usr/bin/nohup</Data>
        <Data Name="FileVersion">-</Data>
        <Data Name="Description">-</Data>
        <Data Name="Product">-</Data>
        <Data Name="Company">-</Data>
        <Data Name="OriginalFileName">-</Data>
        <Data Name="CommandLine">nohup /tmp/env/.qnapd/apached -d</Data
        <Data Name="CurrentDirectory">/tmp/env/.qnapd</Data>
        <Data Name="User">-</Data>
        <Data Name="LogonGuid">{46700b68-0000-0000-ffff-ffffffffffff}</
        <Data Name="LogonId">65535</Data>
```

```
        <Data Name="TerminalSessionId">48</Data>
        <Data Name="IntegrityLevel">no level</Data>
        <Data Name="Hashes">-</Data>
        <Data Name="ParentProcessGuid">{46700b68-ff13-6475-f587-0ded3d5
        <Data Name="ParentProcessId">26633</Data>
        <Data Name="ParentImage">/usr/bin/dash</Data>
        <Data Name="ParentCommandLine">/bin/sh</Data>
        <Data Name="ParentUser">-</Data>
    </EventData>
</Event>
```

The following sigma rule was created to identify the execution of `nohup`.

```
title: Nohup Execution
id: e4ffe466-6ff8-48d4-94bd-e32d1a6061e2
status: experimental
description: Detects usage of nohup which could be leveraged by an atta
references:
    - https://gtfobins.github.io/gtfobins/nohup/
    - https://en.wikipedia.org/wiki/Nohup
    - https://www.computerhope.com/unix/unohup.htm
author: 'Christopher Peacock @SecurePeacock, SCYTHE @scythe_io'
date: 2022/06/06
logsource:
    product: linux
    category: process_creation
detection:
    selection:
        Image|endswith: '/nohup'
    condition: selection
falsepositives:
    - Administrators or installed processes that leverage nohup
level: medium
```

Wget Creating Files in Tmp

Another behavior related to the creation of files is the use of `wget` to download a file into `/tmp/` folder.

```
<Event>
    <System>
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>11</EventID>
        <Version>2</Version>
        <Level>4</Level>
        <Task>11</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
        <EventRecordID>16795</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
    </System>
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff05-6475-9525-20d06e550000}
        <Data Name="ProcessId">26571</Data>
        <Data Name="Image">/usr/bin/wget</Data>
        <Data Name="TargetFilename">/tmp/env/.qnapd/apachedtmp</Data>
        <Data Name="User">-</Data>
```

```
    </EventData>
</Event>
```

The next sigma rule that I've created detect files created by `wget` in `/tmp/` folders, something that is interesting to know.

```
title: Wget Creating Files in Tmp
id: 35a05c60-9012-49b6-a11f-6bab741c9f74
status: experimental
description: Detects the use of wget to download content
references:
    - https://blogs.jpcert.or.jp/en/2023/05/gobrat.html
    - https://www.virustotal.com/gui/file/60bcd645450e4c846238cf0e7226d
author: Joseliyo Sanchez, @Joseliyo_Jstnk
date: 2023/30/05
tags:
    - attack.command_and_control
    - attack.t1105
logsource:
    product: linux
    category: file_event
detection:
    selection:
        Image|endswith: '/wget'
        TargetFilename|contains: '/tmp/'
    condition: selection
falsepositives:
    - Legitimate downloads files in tmp folder.
level: high
```

Execution From Tmp Folder

Probably the most important execution for the threat actors during this infection. The GobRAT payload is loaded from the `/tmp/` folder, something that is suspicious and allow us to detect it with a new sigma rule.

```
<Event>
    <System>
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>1</EventID>
        <Version>5</Version>
        <Level>4</Level>
        <Task>1</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
        <EventRecordID>16949</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
    </System>
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff13-6475-2f02-6b0100000000}
        <Data Name="ProcessId">26638</Data>
        <Data Name="Image">/tmp/env/.qnapd/apached</Data>
        <Data Name="FileVersion">-</Data>
        <Data Name="Description">-</Data>
        <Data Name="Product">-</Data>
        <Data Name="Company">-</Data>
        <Data Name="OriginalFileName">-</Data>
        <Data Name="CommandLine">/tmp/env/.qnapd/apached -d</Data>
        <Data Name="CurrentDirectory">/tmp/env/.qnapd</Data>
```

```
        <Data Name="User">-</Data>
        <Data Name="LogonGuid">{46700b68-0000-0000-ffff-ffffffffffff}</
        <Data Name="LogonId">65535</Data>
        <Data Name="TerminalSessionId">48</Data>
        <Data Name="IntegrityLevel">no level</Data>
        <Data Name="Hashes">-</Data>
        <Data Name="ParentProcessGuid">{46700b68-ca62-6475-65ce-a2aae05
        <Data Name="ParentProcessId">1</Data>
        <Data Name="ParentImage">/usr/lib/systemd/systemd</Data>
        <Data Name="ParentCommandLine">/sbin/init splash</Data>
        <Data Name="ParentUser">root</Data>
    </EventData>
</Event>
```

The sigma rule that I've created is looking for processes loaded in the `/tmp/` folder.

```
title: Execution From Tmp Folder
id: 312b42b1-bded-4441-8b58-163a3af58775
status: experimental
description: Detects suspicious executions from tmp folder
references:
    - https://blogs.jpcert.or.jp/en/2023/05/gobrat.html
    - https://www.virustotal.com/gui/file/60bcd645450e4c846238cf0e7226d
author: Joseliyo Sanchez, @Joseliyo_Jstnk
date: 2023/30/05
logsource:
    product: linux
    category: process_creation
detection:
    selection:
        Image|startswith: '/tmp/'
        ParentImage|startswith: '/usr/'
    condition: selection
falsepositives:
    - Unknown
level: high
```

Wget Download File To Tmp Folder in silent Mode

If we saw a file event to identify file creation of `wget` in `/tmp/`, now is time to detect the use of `wget` in silent mode.

```
<Event>
    <System>
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>1</EventID>
        <Version>5</Version>
        <Level>4</Level>
        <Task>1</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
        <EventRecordID>16794</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
    </System>
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff05-6475-9525-20d06e550000}
        <Data Name="ProcessId">26571</Data>
        <Data Name="Image">/usr/bin/wget</Data>
```

```
        <Data Name="FileVersion">-</Data>
        <Data Name="Description">-</Data>
        <Data Name="Product">-</Data>
        <Data Name="Company">-</Data>
        <Data Name="OriginalFileName">-</Data>
        <Data Name="CommandLine">wget -q -t 1 -T 8 --limit-rate 200k -O
        <Data Name="CurrentDirectory">/opt</Data>
        <Data Name="User">-</Data>
        <Data Name="LogonGuid">{46700b68-0000-0000-ffff-ffffffffffff}</
        <Data Name="LogonId">65535</Data>
        <Data Name="TerminalSessionId">48</Data>
        <Data Name="IntegrityLevel">no level</Data>
        <Data Name="Hashes">-</Data>
        <Data Name="ParentProcessGuid">{46700b68-ff05-6475-f547-38be1a5
        <Data Name="ParentProcessId">26546</Data>
        <Data Name="ParentImage">/usr/bin/dash</Data>
        <Data Name="ParentCommandLine">/bin/sh</Data>
        <Data Name="ParentUser">-</Data>
    </EventData>
</Event>
```

The sigma rule that can help us to detect it is the next one, which can look for the parameters `-q` for silent, `-O` to download and `/tmp/` in the command line, which means that the file could be downloaded in the `tmp` folder.

```
title: Wget Download File To Tmp Folder in Silence Mode
id: cf610c15-ed71-46e1-bdf8-2bd1a99de6c4
status: experimental
description: Detects the use of wget to download content
references:
    - https://blogs.jpcert.or.jp/en/2023/05/gobrat.html
    - https://www.virustotal.com/gui/file/60bcd645450e4c846238cf0e7226d
author: Joseliyo Sanchez, @Joseliyo_Jstnk
date: 2023/30/05
tags:
    - attack.command_and_control
    - attack.t1105
logsource:
    category: process_creation
    product: linux
detection:
    selection:
        Image|endswith: '/wget'
        CommandLine|contains|all:
            - '-q' # Turn off Wget's output.
            - '-O' # output
            - '/tmp/'
            #- '--limit-rate' # Limit the download speed
            #- '--no-check-certificate'
    condition: selection
falsepositives:
    - Legitimate downloads files in tmp folder.
level: high
```

Discovery Information About Files Created by GobRAT

We could consider this behavior to be specific to GobRAT, and it could be. But perhaps it could be related to other families in the future.

```
<Event>
    <System>
```

```
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>1</EventID>
        <Version>5</Version>
        <Level>4</Level>
        <Task>1</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
        <EventRecordID>16812</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
    </System>
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff13-6475-815f-0cee5e550000}
        <Data Name="ProcessId">26575</Data>
        <Data Name="Image">/usr/bin/grep</Data>
        <Data Name="FileVersion">-</Data>
        <Data Name="Description">-</Data>
        <Data Name="Product">-</Data>
        <Data Name="Company">-</Data>
        <Data Name="OriginalFileName">-</Data>
        <Data Name="CommandLine">grep sshd.sh</Data>
        <Data Name="CurrentDirectory">/opt</Data>
        <Data Name="User">-</Data>
        <Data Name="LogonGuid">{46700b68-0000-0000-ffff-ffffffffffff}</
        <Data Name="LogonId">65535</Data>
        <Data Name="TerminalSessionId">48</Data>
        <Data Name="IntegrityLevel">no level</Data>
        <Data Name="Hashes">-</Data>
        <Data Name="ParentProcessGuid">{00000000-0000-0000-0000-0000000
        <Data Name="ParentProcessId">26573</Data>
        <Data Name="ParentImage">-</Data>
        <Data Name="ParentCommandLine">-</Data>
        <Data Name="ParentUser">-</Data>
    </EventData>
</Event>
```

To detect it, the next sigma rule can be used.

```
title: Discovery Information About Files Created by GobRAT
id: e34cfa0c-0a50-4210-9cb3-5632d08eb041
status: experimental
description: Detects the use of grep to discover specific files created
references:
    - https://blogs.jpcert.or.jp/en/2023/05/gobrat.html
    - https://www.virustotal.com/gui/file/60bcd645450e4c846238cf0e7226d
    - https://www.virustotal.com/gui/file/3e44c807a25a56f4068b5b8186eee
author: Joseliyo Sanchez, @Joseliyo_Jstnk
date: 2023/30/05
tags:
    - attack.discovery
    - attack.t1082
logsource:
    category: process_creation
    product: linux
detection:
    selection:
        Image|endswith: '/grep'
        CommandLine|contains:
            - 'apached'
            - 'sshd.sh'
            - 'zone.arm'
            - 'frpc'
    condition: selection
```

```
falsepositives:
    - Unkwnon
level: high
```

Nohup Tmp File Execution

Although there is a rule that simply detects the use of nohup, I thought it might be interesting to have another one with different severity to detect the use of the same process but on a `/tmp/` file. Therefore, I have created the following rule

```
<Event>
    <System>
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>1</EventID>
        <Version>5</Version>
        <Level>4</Level>
        <Task>1</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
        <EventRecordID>16945</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
    </System>
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff13-6475-5131-7d1f45560000}
        <Data Name="ProcessId">26638</Data>
        <Data Name="Image">/usr/bin/nohup</Data>
        <Data Name="FileVersion">-</Data>
        <Data Name="Description">-</Data>
        <Data Name="Product">-</Data>
        <Data Name="Company">-</Data>
        <Data Name="OriginalFileName">-</Data>
        <Data Name="CommandLine">nohup /tmp/env/.qnapd/apached -d</Data
        <Data Name="CurrentDirectory">/tmp/env/.qnapd</Data>
        <Data Name="User">-</Data>
        <Data Name="LogonGuid">{46700b68-0000-0000-ffff-ffffffffffff}</
        <Data Name="LogonId">65535</Data>
        <Data Name="TerminalSessionId">48</Data>
        <Data Name="IntegrityLevel">no level</Data>
        <Data Name="Hashes">-</Data>
        <Data Name="ParentProcessGuid">{46700b68-ff13-6475-f587-0ded3d5
        <Data Name="ParentProcessId">26633</Data>
        <Data Name="ParentImage">/usr/bin/dash</Data>
        <Data Name="ParentCommandLine">/bin/sh</Data>
        <Data Name="ParentUser">-</Data>
    </EventData>
</Event>
```

Besides to identify `nohup` execution, it checks if the command line contains `/tmp/` as well.

```
title: Nohup Tmp File Execution
id: 457df417-8b9d-4912-85f3-9dbda39c3645
status: experimental
description: Detects the use of grep to identify information about the
references:
    - https://blogs.jpcert.or.jp/en/2023/05/gobrat.html
    - https://www.virustotal.com/gui/file/60bcd645450e4c846238cf0e7226d
author: Joseliyo Sanchez, @Joseliyo_Jstnk
date: 2023/30/05
```

```
logsource:
    product: linux
    category: process_creation
detection:
    selection:
        Image|endswith: '/nohup'
        CommandLine|contains: '/tmp/'
    condition: selection
falsepositives:
    - System Administrators using nohup
level: high
```

Execution of Shell With Tmp Parent Process

The GobRAT payload makes use of `/bin/bash` from `/tmp/` folder, something that is insteresting.

```
<Event>
    <System>
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>1</EventID>
        <Version>5</Version>
        <Level>4</Level>
        <Task>1</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
        <EventRecordID>16950</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
    </System>
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff14-6475-0507-2fb57a550000}
        <Data Name="ProcessId">26649</Data>
        <Data Name="Image">/usr/bin/bash</Data>
        <Data Name="FileVersion">-</Data>
        <Data Name="Description">-</Data>
        <Data Name="Product">-</Data>
        <Data Name="Company">-</Data>
        <Data Name="OriginalFileName">-</Data>
        <Data Name="CommandLine">/bin/bash -c uptime</Data>
        <Data Name="CurrentDirectory">/tmp/env/.qnapd</Data>
        <Data Name="User">-</Data>
        <Data Name="LogonGuid">{46700b68-0000-0000-ffff-ffffffffffff}</
        <Data Name="LogonId">65535</Data>
        <Data Name="TerminalSessionId">48</Data>
        <Data Name="IntegrityLevel">no level</Data>
        <Data Name="Hashes">-</Data>
        <Data Name="ParentProcessGuid">{46700b68-ff13-6475-2f02-6b01000
        <Data Name="ParentProcessId">26638</Data>
        <Data Name="ParentImage">/tmp/env/.qnapd/apached</Data>
        <Data Name="ParentCommandLine">/tmp/env/.qnapd/apached</Data>
        <Data Name="ParentUser">-</Data>
    </EventData>
</Event>
```

Any binary from `/tmp/` using `/bin/bash` or `/bin/sh`, it is interesting to know. Therefore, I have generated the following rule.

```
title: Execution of Shell With Tmp Parent Process
id: 2fade0b6-7423-4835-9d4f-335b39b83867
```

```
status: experimental
description: Detects suspicious executions of shells with tmp parents
references:
    - https://blogs.jpcert.or.jp/en/2023/05/gobrat.html
    - https://www.virustotal.com/gui/file/60bcd645450e4c846238cf0e7226d
author: Joseliyo Sanchez, @Joseliyo_Jstnk
date: 2023/30/05
logsource:
    product: linux
    category: process_creation
detection:
    selection:
        ParentImage|startswith: '/tmp/'
        CommandLine|contains:
          - '/bin/bash'
          - '/bin/sh'
    condition: selection
falsepositives:
    - Unknown
level: high
```

Execution of Script In Tmp Folder

Very similar to the above section, however, in this case they executing a script stored in `/tmp/` using `/bin/sh`.

```
<Event>
    <System>
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>1</EventID>
        <Version>5</Version>
        <Level>4</Level>
        <Task>1</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
        <EventRecordID>16981</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
    </System>
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff45-6475-f597-3cd5db550000}
        <Data Name="ProcessId">26665</Data>
        <Data Name="Image">/usr/bin/dash</Data>
        <Data Name="FileVersion">-</Data>
        <Data Name="Description">-</Data>
        <Data Name="Product">-</Data>
        <Data Name="Company">-</Data>
        <Data Name="OriginalFileName">-</Data>
        <Data Name="CommandLine">/bin/sh -c /tmp/env/.qnapd/sshd.sh</Da
        <Data Name="CurrentDirectory">/root</Data>
        <Data Name="User">-</Data>
        <Data Name="LogonGuid">{46700b68-0000-0000-ffff-ffffffffffff}</
        <Data Name="LogonId">65535</Data>
        <Data Name="TerminalSessionId">49</Data>
        <Data Name="IntegrityLevel">no level</Data>
        <Data Name="Hashes">-</Data>
        <Data Name="ParentProcessGuid">{00000000-0000-0000-0000-0000000
        <Data Name="ParentProcessId">26664</Data>
        <Data Name="ParentImage">-</Data>
        <Data Name="ParentCommandLine">-</Data>
        <Data Name="ParentUser">-</Data>
```

```
    </EventData>
</Event>
```

And if you want to detect this kind of behavior, use the next sigma rule.

```
title: Execution of Script In Tmp Folder
id: 30bcce26-51c5-49f2-99c8-7b59e3af36c7
status: experimental
description: Detects executions of scripts stored in tmp folder using b
references:
    - https://blogs.jpcert.or.jp/en/2023/05/gobrat.html
    - https://www.virustotal.com/gui/file/60bcd645450e4c846238cf0e7226d
author: Joseliyo Sanchez, @Joseliyo_Jstnk
date: 2023/30/05
logsource:
    product: linux
    category: process_creation
detection:
    selection_shell:
        CommandLine|contains:
          - '/bin/bash'
          - '/bin/sh'
    selection_parameters:
        CommandLine|contains|all:
          - '/tmp/'
          - '-c'
    condition: all of selection_*
falsepositives:
    - Unknown
level: high
```

File Deletion

Multiple files are deleted during the execution, probably to avoid be detected by software and analysts. Most of the actity is under the path `/tmp/env/.qnapd/`. Next sysmon event is just one of the files deleted during the exeution.

```
<Event>
    <System>
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>1</EventID>
        <Version>5</Version>
        <Level>4</Level>
        <Task>1</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
        <EventRecordID>16774</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
    </System>
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff05-6475-e1f4-c3d0eb550000}
        <Data Name="ProcessId">26560</Data>
        <Data Name="Image">/usr/bin/rm</Data>
        <Data Name="FileVersion">-</Data>
        <Data Name="Description">-</Data>
        <Data Name="Product">-</Data>
        <Data Name="Company">-</Data>
        <Data Name="OriginalFileName">-</Data>
        <Data Name="CommandLine">rm -rf /tmp/env/.qnapd/bot.log</Data>
```

```
        <Data Name="CurrentDirectory">/opt</Data>
        <Data Name="User">-</Data>
        <Data Name="LogonGuid">{46700b68-0000-0000-ffff-ffffffffffff}</
        <Data Name="LogonId">65535</Data>
        <Data Name="TerminalSessionId">48</Data>
        <Data Name="IntegrityLevel">no level</Data>
        <Data Name="Hashes">-</Data>
        <Data Name="ParentProcessGuid">{46700b68-ff05-6475-f547-38be1a5
        <Data Name="ParentProcessId">26546</Data>
        <Data Name="ParentImage">/usr/bin/dash</Data>
        <Data Name="ParentCommandLine">/bin/sh</Data>
        <Data Name="ParentUser">-</Data>
    </EventData>
</Event>
```

And yes, there is a public sigma rule to detect files deleted in the system! Although it is a little generic, it can help us

```
title: File Deletion
id: 30aed7b6-d2c1-4eaf-9382-b6bc43e50c57
status: stable
description: Detects file deletion using "rm", "shred" or "unlink" comm
references:
    - https://github.com/redcanaryco/atomic-red-team/blob/f339e7da7d05f
author: Ömer Günal, oscd.community
date: 2020/10/07
modified: 2022/09/15
tags:
    - attack.defense_evasion
    - attack.t1070.004
logsource:
    product: linux
    category: process_creation
detection:
    selection:
        Image|endswith:
            - '/rm'     # covers /rmdir as well
            - '/shred'
            - '/unlink'
    condition: selection
falsepositives:
    - Legitimate administration activities
level: informational
```

Process Discovery

The use of `ps` command is another capability implemented by this loader during the execution.

```
<Event>
    <System>
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>1</EventID>
        <Version>5</Version>
        <Level>4</Level>
        <Task>1</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
        <EventRecordID>16813</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
```

```
    </System>
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff13-6475-f1bb-6c01f7550000}
        <Data Name="ProcessId">26574</Data>
        <Data Name="Image">/usr/bin/ps</Data>
        <Data Name="FileVersion">-</Data>
        <Data Name="Description">-</Data>
        <Data Name="Product">-</Data>
        <Data Name="Company">-</Data>
        <Data Name="OriginalFileName">-</Data>
        <Data Name="CommandLine">ps -ef</Data>
        <Data Name="CurrentDirectory">/opt</Data>
        <Data Name="User">-</Data>
        <Data Name="LogonGuid">{46700b68-0000-0000-ffff-ffffffffffff}</
        <Data Name="LogonId">65535</Data>
        <Data Name="TerminalSessionId">48</Data>
        <Data Name="IntegrityLevel">no level</Data>
        <Data Name="Hashes">-</Data>
        <Data Name="ParentProcessGuid">{00000000-0000-0000-0000-0000000
        <Data Name="ParentProcessId">26573</Data>
        <Data Name="ParentImage">-</Data>
        <Data Name="ParentCommandLine">-</Data>
        <Data Name="ParentUser">-</Data>
    </EventData>
</Event>
```

Public sigma rule that can help us to detect the use of `ps` with information level.

```
title: Process Discovery
id: 4e2f5868-08d4-413d-899f-dc2f1508627b
status: stable
description: |
  Detects process discovery commands. Adversaries may attempt to get in
  Information obtained could be used to gain an understanding of common
references:
    - https://github.com/redcanaryco/atomic-red-team/blob/f339e7da7d05f
author: Ömer Günal, oscd.community
date: 2020/10/06
modified: 2022/07/07
tags:
    - attack.discovery
    - attack.t1057
logsource:
    product: linux
    category: process_creation
detection:
    selection:
        Image|endswith:
            - '/ps'
            - '/top'
    condition: selection
falsepositives:
    - Legitimate administration activities
level: informational
```

System Information Discovery

GobRAT Loaders execute multiple discovery commands during the infection. Just to mention a few of them, next two sysmon events are related to system information discovery.

```
<Event>
    <System>
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>1</EventID>
        <Version>5</Version>
        <Level>4</Level>
        <Task>1</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
        <EventRecordID>16950</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
    </System>
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff14-6475-0507-2fb57a550000}
        <Data Name="ProcessId">26649</Data>
        <Data Name="Image">/usr/bin/bash</Data>
        <Data Name="FileVersion">-</Data>
        <Data Name="Description">-</Data>
        <Data Name="Product">-</Data>
        <Data Name="Company">-</Data>
        <Data Name="OriginalFileName">-</Data>
        <Data Name="CommandLine">/bin/bash -c uptime</Data>
        <Data Name="CurrentDirectory">/tmp/env/.qnapd</Data>
        <Data Name="User">-</Data>
        <Data Name="LogonGuid">{46700b68-0000-0000-ffff-ffffffffffff}</
        <Data Name="LogonId">65535</Data>
        <Data Name="TerminalSessionId">48</Data>
        <Data Name="IntegrityLevel">no level</Data>
        <Data Name="Hashes">-</Data>
        <Data Name="ParentProcessGuid">{46700b68-ff13-6475-2f02-6b01000
        <Data Name="ParentProcessId">26638</Data>
        <Data Name="ParentImage">/tmp/env/.qnapd/apached</Data>
        <Data Name="ParentCommandLine">/tmp/env/.qnapd/apached</Data>
        <Data Name="ParentUser">-</Data>
    </EventData>
</Event>
```

```
<Event>
    <System>
        <Provider Name="Linux-Sysmon" Guid="{ff032593-a8d3-4f13-b0d6-01
        <EventID>1</EventID>
        <Version>5</Version>
        <Level>4</Level>
        <Task>1</Task>
        <Opcode>0</Opcode>
        <Keywords>0x8000000000000000</Keywords>
        <EventRecordID>16784</EventRecordID>
        <Correlation/>
        <Execution ProcessID="26414" ThreadID="26414"/>
        <Channel>Linux-Sysmon/Operational</Channel>
        <Security UserId="0"/>
    </System>
    <EventData>
        <Data Name="RuleName">-</Data>
        <Data Name="ProcessGuid">{46700b68-ff05-6475-81fa-b29af3550000}
        <Data Name="ProcessId">26565</Data>
        <Data Name="Image">/usr/bin/uname</Data>
        <Data Name="FileVersion">-</Data>
        <Data Name="Description">-</Data>
        <Data Name="Product">-</Data>
        <Data Name="Company">-</Data>
```

```
        <Data Name="OriginalFileName">-</Data>
        <Data Name="CommandLine">uname -m</Data>
        <Data Name="CurrentDirectory">/opt</Data>
        <Data Name="User">-</Data>
        <Data Name="LogonGuid">{46700b68-0000-0000-ffff-ffffffffffff}</
        <Data Name="LogonId">65535</Data>
        <Data Name="TerminalSessionId">48</Data>
        <Data Name="IntegrityLevel">no level</Data>
        <Data Name="Hashes">-</Data>
        <Data Name="ParentProcessGuid">{46700b68-ff05-6475-f547-38be1a5
        <Data Name="ParentProcessId">26546</Data>
        <Data Name="ParentImage">/usr/bin/dash</Data>
        <Data Name="ParentCommandLine">/bin/sh</Data>
        <Data Name="ParentUser">-</Data>
    </EventData>
</Event>
```

For this behaviors, there is another public sigma rule to detect them.

```
title: System Information Discovery
id: 42df45e7-e6e9-43b5-8f26-bec5b39cc239
status: stable
description: Detects system information discovery commands
references:
    - https://github.com/redcanaryco/atomic-red-team/blob/f339e7da7d05f
author: Ömer Günal, oscd.community
date: 2020/10/08
modified: 2021/09/14
tags:
    - attack.discovery
    - attack.t1082
logsource:
    product: linux
    category: process_creation
detection:
    selection:
        Image|endswith:
            - '/uname'
            - '/hostname'
            - '/uptime'
            - '/lspci'
            - '/dmidecode'
            - '/lscpu'
            - '/lsmod'
    condition: selection
falsepositives:
    - Legitimate administration activities
level: informational
```

Conclusions

Although the campaign was initially aimed at routers, as we have seen, the behaviors are very generic and linux endpoints could be targeted, where in fact some of the analyzed behaviors have already been used previously by other threat actors.

This is an approach on how from a CTI point of view where new trends, campaigns and malware are analyzed, it can lead to the generation of new threats.

All the sigma rules created by myself are pushed to the official repo.

Contact

Twitter: https://twitter.com/Joseliyo_Jstnk

LinkedIn: https://www.linkedin.com/in/joseluissm/

Tags: threat hunting | detection | analysis | research | GobRAT | malware | sigma

Newer Post
« Malicious document identified in the conflict Israel & Gaza themed about terrorist organizations related to Iran

Older Post
Analyzing AsyncRAT distributed in Colombia by Blind Eagle »