radware

INTERNET OF THREATS: IoT BOTNETS AND THE ECONOMICS OF DDOS PROTECTION

In more ways than one, IoT botnets transformed cyber security forever. They introduced the industry to the 1Tbps cyber-attack and sophisticated vectors like GRE floods and DNS water torture. Mirai, the 2016 posterchild for bot attacks, rewrote the rules as the world's first open source botnet that can be customized.

As a result, the battle of the bots is on everybody's mind. According to Radware's *2016-2017 Global Application and Network Security Report*, 55% of security professionals believe that that Internet of Things complicates mitigation and detection requirements.

2016 was the year that this long-feared DDoS threat came to fruition. Cyber-attacks were launched from thousands of connected devices turned bot. These high profile assaults gained worldwide notoriety and placed cyber security on front page of mainstream media outlets. This included:

June 28, 2016: PCWorld reports that "25,000 digital video recorders and CCTV cameras were compromised and used to launch distributed denial-of-service (DDoS) attacks, flooded targets with about 50,000 HTTP requests per second."[1] Though impressive and startling, this attack said nothing about what was still to come.

September 20, 2016: Around 8:00 pm, KrebsOnSecurity.com becomes the target of a record-breaking 620Gbps[2] volumetric DDoS attack from a botnet designed to take the site offline.

1 http://www.pcworld.com/article/3089346/security/thousands-of-hacked-cctv-devices-used-in-ddos-attacks.html
2 https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/

September 21, 2016: The same type of botnet is used in a 1Tbps attack targeting the French web host OVH.[3] A few days later, the IoT botnet source code goes public—spawning what would become the "marquee" attack of the year.

October 21, 2016: Dyn, a U.S.-based DNS provider that many Fortune 500 companies rely on, is attacked by the same botnet in what is publicly known as a "water torture" attack (see below). The attack renders many services unreachable and causes massive connectivity issues—mostly along the East Coast of the United States.

The Appeal of Internet of Things (IoT) Devices

For hackers, IoT devices are attractive targets for several reasons:

- IoT devices usually fall short when it gets to endpoint protection implementation.
- Unlike PCs and servers, there are no regulations or standards for secure use of IoT devices. Such regulations help ensure secured configurations and practices. Among them: changing default passwords and implementing access control restrictions (for example, to disable remote access to administrative ports).
- IoT devices operate 24x7 and can be in use at any moment.

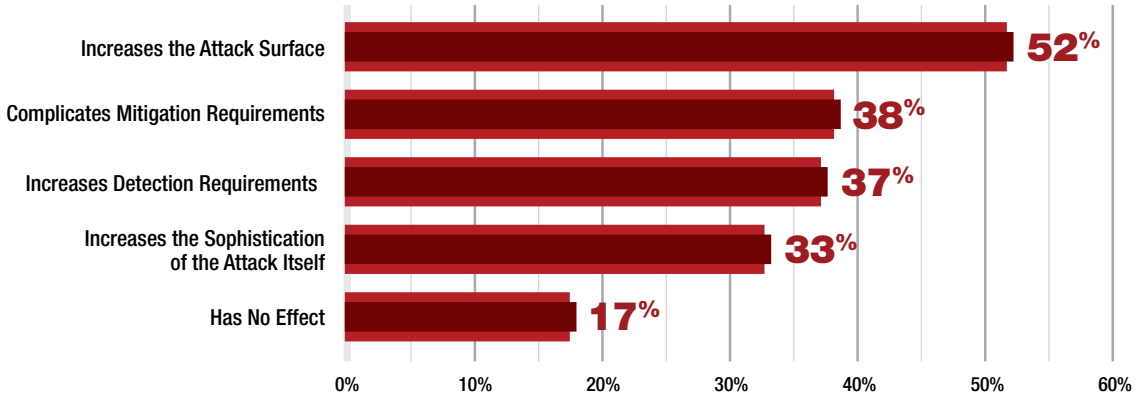

Figure 1: IoT threat impact as perceived by cyber security professionals

Mirai Under the Microscope

As an open-source attack program, Mirai is fueling justifiable fears that hackers will create countless customizations and evolutions of the tool. To help understand the risks, Radware's security research team conducted a thorough study of the infamous botnet.

Figure 2: Infection map provided by botnets researcher @ MalwareTechBlog

We can all thank a user named "Anna-senpai" for publishing the Mirai source code to a public and easily accessible forum. In short order, the code spread to numerous locations, including several GitHub repositories, where hackers began taking a closer look. Since then, the Mirai botnet has been infecting hundreds of thousands of IoT devices—turning them into a "zombie army" capable of launching powerful volumetric DDoS attacks. Security researchers estimate that there are millions of vulnerable IoT devices actively taking part in these coordinated attacks.

3 https://twitter.com/olesovhcom/status/779297257199964160

In a surprising departure from previous record-holding amplification attacks, attackers did not use DNS and NTP. Instead, these attacks consisted mainly of TCP-SYN, TCP-ACK and TCP-ACK + PSH along with HTTP and non-amplified UDP floods. In the case of KrebsOnSecurity, the biggest chunk of attack traffic came in the form of GRE, which is highly unusual.[4] In the OVH attack, more than 140,000 unique IPs were reported in what seemed to be a SYN and ACK flood attack followed by short bursts over 100Gbps each over a four-day period.[5]

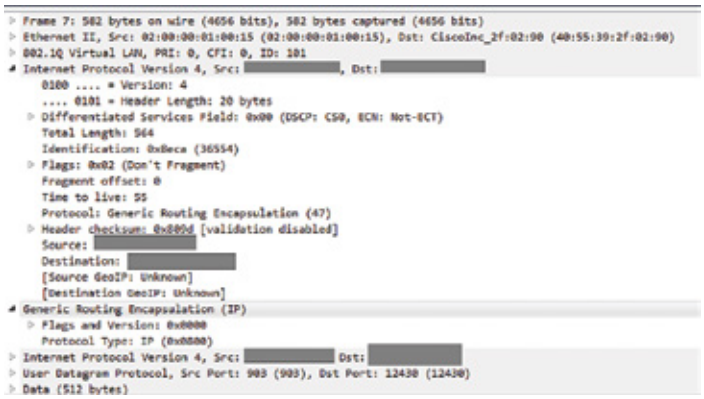

Figure 3: The bot sends GRE packets with encapsulated UDP packet containing 512 bytes of random data

Outstanding Attack Vectors

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 amounts of encapsulated data may lead to resource consumption, with the victim attempting to de-encapsulate them until exhaustion.

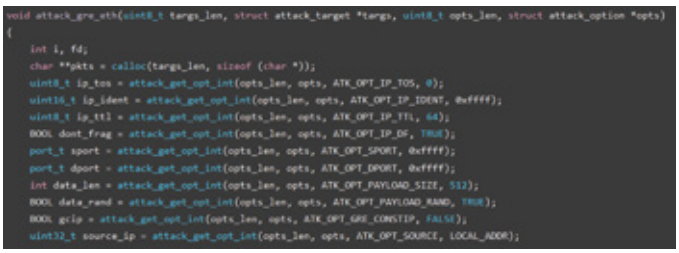

Figure 4: A function creates a GRE packet and includes it within a GRE flood attack

TCP STOMP Attack

Consider this akin to the classic ACK flood attack—with a twist. Most network security solutions will easily block simple botnets as they send large volumes of ACK packets. Thus, Mirai starts with the ACK flood only after 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.

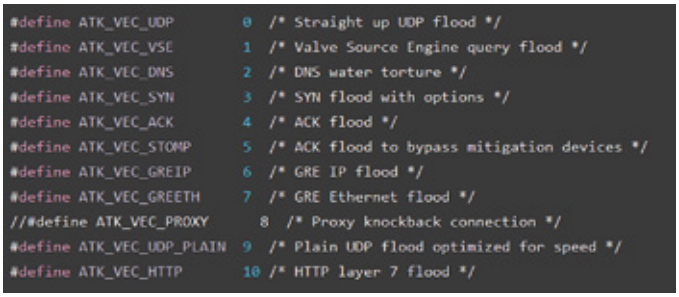

Figure 5: Menu of all Mirai's attack vectors

DNS Water Torture Attack

With this technique, the attacker sends a pre-crafted DNS query to the service provider's 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 name server repeatedly with no success. Sending different false strings with the victims' domain name will eventually increase the DNS server's CPU utilization until it is no longer available.

What follows is a concise overview of how Mirai operates:

1. Connects to victim machines via a brute-force attack against Telnet servers, using 60+ factory default credentials of BusyBox.[6]
2. Every infected device locks itself against additional bots.
3. Mirai sends the victim's IP and credentials to a centralized ScanListen service.
4. The new victim then helps in harvesting new bots, spawning a self-replicating pattern.
5. Once all devices are ready, Mirai launches the attack.

4 https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/
5 https://twitter.com/olesovhcom/status/779297257199964160
6 https://en.wikipedia.org/wiki/BusyBox

What makes Mirai so powerful? Consider that:

1. Setup is fast and easy; in fact, it can be completed within an hour.
2. Distribution is rapid. The infection recurrence mechanism leads to exponential growth in the botnet's size. In fact, perpetrators can have a botnet of 100,000+ infected devices in 24 hours.
3. Leveraging an efficient Communicating Sequential Processes (CSP) design, this distributed micro-service architecture allows for scalable control of bots and attack execution in very large botnets.
4. This piece of malware has a low detection rate. It is very difficult to retrieve samples because the malicious code lives in the device's memory and is wiped out once the device is restarted.
5. Mirai also offers configurable attack features, including the ability to specify packet size, randomize packet size, use Tos/idnt/ttl in IP header, force the source and destination ports and use TCP urg/ack/psh.rst/syn/fin.

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

Figure 6: Mirai's HTTP flood program creates huge 80MB POST requests

6. 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 7: Mirai tries to bypass DDoS protection.

Open-Source Attack Tools Open Pandora's Box

The act of leaking or flat-out releasing source code of advanced hacking tools isn't new. It has happened numerous times, especially with high-profile and advanced malware families, such as Zeus, Citadel, Carberp and SpyEye, which have been responsible for losses measuring in the hundreds of millions of dollars. Once dangerous tools are released to the public, they can be downloaded—and modified and enhanced—by anyone.

Figure 8: "I made my money, there's lots of eyes looking at IoT now" –Anna-senpai

As security reporter Brian Krebs wrote, “Miscreants who develop malicious software often dump their source code publicly when law enforcement investigators and security firms start sniffing around a little too close to home.”

That can fuel copycats—and “enhanced” copycats. Radware performed a quick test to see how easy or difficult it would be for an average hacker to take the now open-sourced Mirai source code and extend its capabilities with a new, advanced attack vector.

```
#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 9: Mirai 1.0 source code showing attack vectors including UDP, DNS, SYN, GRE, HTTP

To do this, we considered implementing several advanced attacks that are NOT currently implemented in the original Mirai source code, such as:

1. SSL attacks
2. Layer-7 HTTP attacks with JavaScript support
3. HTTP 2.0 support

> These advanced Layer-7 attacks combined with the massive size and scale of IoT botnets are indeed very dangerous.

From there, we began our experiment. We were able to acquire the Mirai source code in a matter of minutes on GitHub. Compiling the bot binary and building it for the x86 platform took five minutes and did not require any programming skills.

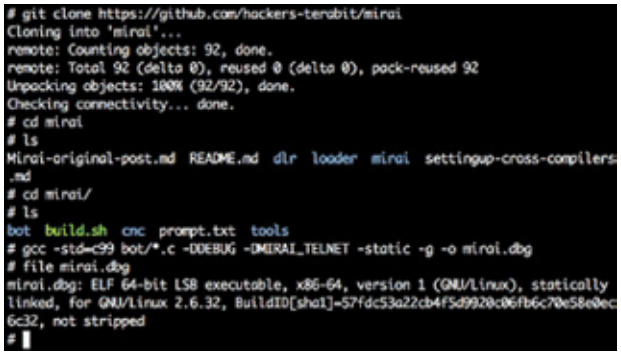

Figure 10: Radware obtains the Mirai source code from one of the GitHub repositories and builds the attack bot binary

In less than an hour, we have managed to integrate another open-source attack tool called thc-ssl-dos, which can be used to launch SSL RENEGOTIATION attacks against web servers. With some elementary coding skills, we slightly modified the code to stress servers that do not allow SSL renegotiation by rapidly establishing new TCP connection on each SSL handshake.

Benchmarking Our Code

We have performed some basic benchmarking of our new attack vector capabilities against a target low-end server (Intel Xeon E3-1245V2, 16gb RAM) running Nginx 1.10 web server (built with OpenSSL 1.0.2g). The client that was used to launch these attacks was sitting on a different remote server, with a latency of ~15 milliseconds roundtrip time.

Create Your Own Botnet Within an Hour

1. Download the Mirai code from GitHub (5 minutes)
2. Compile the bot binary (5 minutes)
3. Integrate other open-source attack tools (50 minutes)

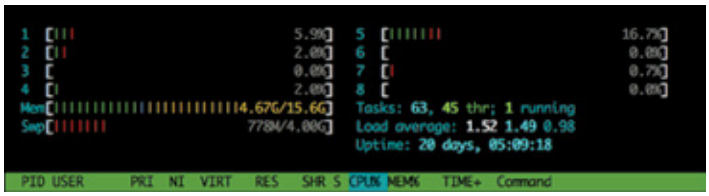

Figure 11: We can see that during "peacetime," the server CPU usage is very low (4 cores, 8 threads)

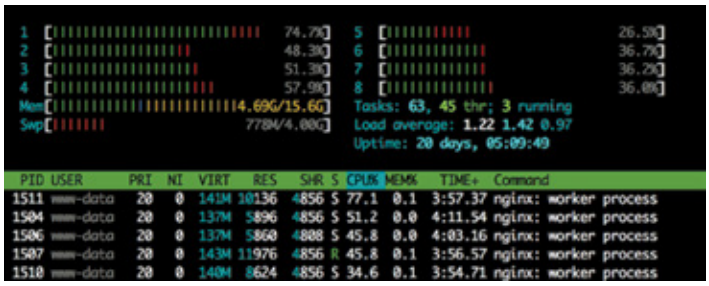

Figure 12: But when we launch an SSL attack using our "improved" Mirai bot, our server starts to get "busy" handling the incoming SSL connections

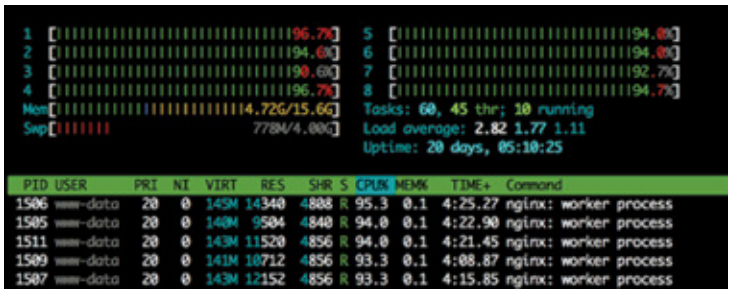

Figure 13: Running as few as two simultaneous attacks now puts our server under real stress at nearly 100% CPU on all cores

In our test landscape, we have observed that a single instance of our new Mirai code is capable of generating 350 SSL connections per second, which takes 50% of our server CPU resources. Multiple instances easily bring the server to full CPU utilization—dramatically hurting system performance and availability.

For large enterprises with high-end backend servers, load balancers, proxies and the like, 350 SSL connections per second is negligible. However, if we extrapolate this value to 100,000 instances—or even 1,000,000 instances—the resulting numbers are large enough to take down, in theory, every major website.

Of course, we need to remember that an IoT device is running on very low power and with limited CPU/network capabilities. Even so, if we take a factor of x1,000, then an IoT botnet with 20,000 zombies will generate an attack that is 20 times higher than the one we have measured.

The Economics of Botnets

While much has been discussed around Mirai, IoT, "the rise of the machines" and other catchy buzz-phrases, we believe one of the most disruptive changes is the new economics model of IoT botnets.

Not so long ago, hackers were investing a great deal of money, time and effort to scan the Internet for vulnerable servers, build their zombie bots army and then safeguard it against other hackers who might also want to claim ownership of them. All the while, hackers would keep continual watch for new infection targets that could join their zombie army.

Things have changed: Now we see millions of vulnerable devices sitting with default credentials. Bot masters—the authors and owners of the botnets—do not even bother to secure their bots after infection. After all, as Mirai demonstrates, it does not even persist infection to disk, so a simple device reboot brings it back to a clean and healthy state.

Nevertheless, this will not prevent re-infection. As we now know, it takes less than six minutes to scan the entire IPv4 space—and the time-to-infection of vulnerable devices is constantly dropping. It is now estimated to take less than an hour.

For a bot master, gaining control of powerful servers with 1Gbit cards or 10Gbit cards was considered to be the ultimate goal—the "Holy Grail." Sometimes a hacker would pay hundreds of dollars every month for it. Often he or she would gain illegal access to it and work very diligently to hide it from others. And finding these servers—then gaining access and maintaining exclusive control—was and still is difficult and expensive.

Now with IoT botnets, we see a different picture. Instead of spending months of effort and hundreds of dollars to control a few powerful servers and several hundred infected PCs, bot masters can take control of millions of IoT devices with near zero cost.

What Now?

To date, the number of connected devices is estimated at 6 billion, while the estimated Internet user count is just 3.5 billion (though expected to grow to 13 billion by 2020). This shift points to a different economy—and requires changes in thought and action.

The botnet attacks of 2016 also underscore the need to move beyond security as an IoT afterthought. IoT platforms and devices need to be designed—from the ground up—to be secure. Right now it is far too simple to victimize IoT devices; all it takes is telnet and a limited list of factory default usernames and passwords to generate botnets of unimaginable proportions. And this is only the beginning.

Reducing the potential impact of IoT botnets should be a combined effort by all IoT stake holders:

1. **"Smart appliances" manufacturers** need to be mindful of producing resilient products with robust security components.
2. To protect enterprise customers, **network carriers** need the ability to detect and manage traffic that originates from such devices.
3. **Enterprise customers** should understand that when making a security investment to protect their infrastructure and assets, they need to be able to protect not only from today's threats, but also from those that will arise in the next three to five years.

The bottom line: The effort and money we've been expending to build defenses is no longer proportional to attackers' investments. It is time to review the attack landscape, re-evaluate the architecture of defense mechanisms and consider how best to defend against higher-order-of-magnitude attacks.

Download the 2016-2017 Global Application & Network Security Report to learn more.

www.radware.com/ert-report-2016

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.