radware

THE RISE OF THE BOTNETS: MIRAI & HAJIME

The Internet of things is fraught with connected devices offering a staggeringly low level of security. Depending on which source is consulted, the number of IoT devices could reach as many as 20 billion by 2020. With hackers using an array of new malware to commandeer these digitized soldiers into botnet armies, it was only a matter of time until hackers unleashed these devices into a massive, distributed DDoS attack.

It's a concept that Radware predicted for years and that was vividly brought to life by the Mirai botnet in October of 2016. Ominously named after the Japanese phrase for "the future," Mirai showed just how much damage even simple, unsophisticated bots could cause. Think of Mirai as the brute-force bot: big, dumb and dangerous.

Soon after, another IoT botnet emerged. Called Hajime, this botnet brings more sophistication to some of the techniques used by Mirai. Rather than corralling an army of bots to wage attacks, Hajime seems to be designed more for staking a claim to IoT devices. So far, Hajime has booted existing bots, closing ports and hunkering down in devices. Its ultimate goal is still unknown—but the potential for global damage looms. Think of Hajime as a covert-operator bot: sneaky and mysterious with the potential to affect great harm.

Based on research and analysis by Radware's Emergency Response Team and presented in the eBook *When the Bots Come Marching In*, let's analyze these two botnets and how they inflict their damage. Although they are sure to evolve and gain competition from new entrants, understanding these two bots provides important insights into the war for IoT market share and will shed light on future DDoS trends and IoT-based threats.

MIRAI: BIG AND SIMPLE. INCREDIBLY LETHAL.

Worries about IoT security and the threat of lethal IoT zombies had been buzzing for years. But Mirai is what finally prompted an inflection point—demonstrating that hundreds of thousands of infected IoT devices could generate massive DDoS attacks. Here's how it unfolded in 2016:

- **September 20** – Around 8:00 pm, KrebsOnSecurity.com becomes the target of a record-breaking 620Gbps volumetric DDoS attack designed to take the site offline. Large portions of the attack traffic consisted of GRE payloads—very unusual in large-volume attacks.
- **September 21** – The same type of botnet was used in a large volumetric attack targeting the French Web host OVH. What distinguished this attack was the enormous number of devices used to perform it. The target of the attack was not OVH, but a Minecraft™ gaming server hosted in OVH.
- **September 30** – The Mirai botnet source code was published on HackForums.net by a person using the online name of Anna-Sempai—spawning what became the "marquee" tool of the year.
- **October 21** – Dyn, a US-based DNS provider that many Fortune 500 companies rely on, was attacked by the same botnet in what is publicly known as a "water torture" attack. The attack rendered many services unreachable and caused massive connectivity issues—mostly along the East Coast of the United States.

SIMPLE YET LETHAL

Though unsophisticated compared to its Windows cousins, Mirai proved that a simple IoT botnet can be both efficient and effective in taking down targets. Indeed, the original Mirai didn't use any sophisticated infection vectors. Instead, it applied brute force on Telnet with a limited dictionary of 61 username/password combinations. It employed a simple, clear-text TCP-based protocol on port 23 for CnC communications. It omitted domains or Domain Generation Algorithms (DGA) to protect its CnC from being discovered and easily blacklisted. And it had no "upgrade" features, underscoring that IoT bots don't require fancy features to do their jobs. In fact, IoT botnets like Mirai can be considered disposable. If an old botnet gets compromised, it can be instantly tossed out and a new one easily obtained.

In addition to having the ability to easily generate traffic volumes above 1Tbps, Mirai features a selection of 10 predefined attack vectors, some of which have proven effective in taking down the infrastructure of service providers by attacking them using GRE floods. Among the 10 vectors are highly sophisticated attack vectors, such as TCP STOMP and DNS resolver flood ("water torture") attacks. Mirai's DDoS attacks also highlight the challenges organizations face when it comes to visibility into the legitimacy of GRE traffic or recursive DNS queries.

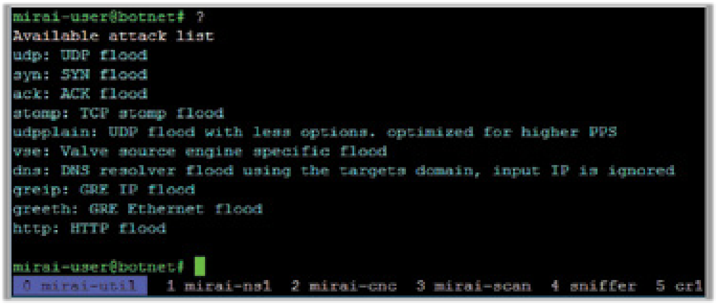
Mirai might be simple and unsophisticated, but it has succeeded at rewriting the rules and affirming new risks from IoT DDoS botnets. As the first IoT open-source botnet, Mirai shook up the status quo around real-time mitigation and made security automation a must. It isn't just that IoT botnets can facilitate sophisticated L7 attack launches that adapt themselves continuously to evade any protective measures while keeping record high volumes. The fact that Mirai is open-source means hackers can potentially mutate, customize and improve it—resulting in an untold variety of new attack tools that can be detected only through intelligent automation.

HAJIME: SMART, SOPHISTICATED TECHNIQUES. ELUSIVE MOTIVES.

On October 16, 2016, Sam Edwards and Ioannis Profetis from Rapidity Networks published a report on a new malware they had discovered and named Hajime.[1] (While Mirai is the Japanese word for “the future,” Hajime can mean “beginning.”[2]) The malware’s discovery followed the release of the Mirai source code and Mirai’s attacks on Krebs and OVH. And before this new malware could make any headlines of its own, Mirai was attributed as the source of the October 21 attacks that took down Dyn and its “who’s who” list of customers that included Amazon, CNN, Netflix, Spotify and Twitter.

The new malware may have evaded the limelight, but it didn’t go away. In fact, it continues to grow—steadily and silently.

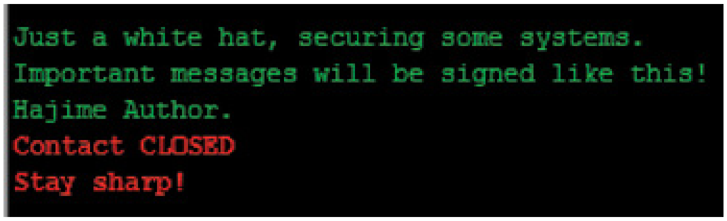
After the Rapidity Networks report, which cited a vulnerability in the encryption code of the CnC channels, the author of the malware updated his botnet with a newer, improved version. This version, no longer vulnerable, also adopted the name “Hajime,” a message written periodically to the terminal revealed “Hajime Author.”



```
mirai-user@botnet# ?
Available attack list
udp: UDP flood
syn: SYN flood
ack: ACK flood
stomp: TCP stomp 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
greip: GRE IP flood
greeth: GRE Ethernet flood
http: HTTP flood

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

Figure 1: Mirai botnet’s command set.



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

Figure 2: Message periodically displayed on the terminal by Hajime.

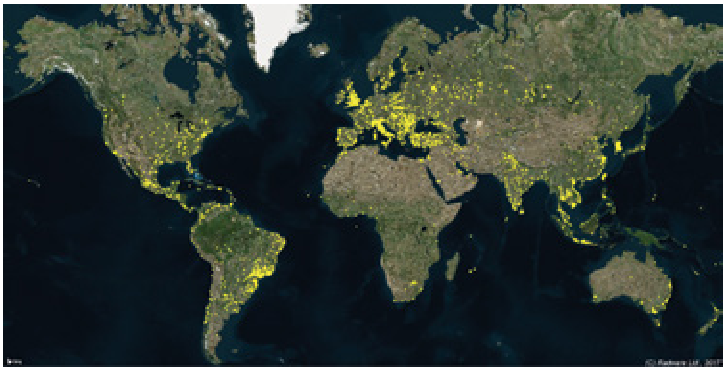


Figure 3. Hajime’s geographic reach, with every dot accounting for a unique IP that Radware was able to identify as an infected device

Notably, Hajime has been gaining considerable IoT market share since its discovery. Infection attempts by Hajime account for nearly half of the total IoT bot activity in Radware’s honeypots (which we use to lure hackers and attacks for the purpose of studying them). Radware discovered that upon infecting, the Hajime bot sometimes leverages other infected nodes to download its malware. That increased the coverage—bringing the total number of Hajime-compromised devices identified by our honeypots to nearly 19,000 (see figure 3). According to Kaspersky, the peer-to-peer network had reached almost 300,000 devices by April 25, 2017.[3]

A NEXT-GENERATION IOT BOT

Unlike Mirai, Hajime is sophisticated, flexible, thoughtfully designed and future proof. At the same time, Hajime leverages the infection method and brute force exploits of Mirai. Capable of updating itself, Hajime can quickly and efficiently extend its member bots with “richer” functionality. The distributed bot network used for command and control and updating uses trackerless torrents. These torrent channels sit on top of the well-known public BitTorrent peer-to-peer network using several dynamic info_hashes that change daily to better conceal CnC activity. All communications through BitTorrent are further signed and encrypted using RC4 and private/public keys.

[1] https://security.rapiditynetworks.com/publications/2016-10-16/hajime.pdf
[2] https://www.extremetech.com/internet/248087-meet-hajime-iot-botnet-built-vaccinate-devices-mirai
[3] https://securelist.com/blog/research/78160/hajime-the-mysterious-evolving-botnet/

Hajime is a modular malware that provides support for extension modules. The current extension module provides scan and loader services to discover and infect new victims. The efficient SYN scanner implementation seeks new victims through open ports TCP/23 (Telnet) and TCP/5358 (WSDAPI). Upon discovering open Telnet ports, the extension module tries to exploit the victim using brute force shell login much the same way Mirai did. For this purpose, Hajime uses a list consisting of the 61 factory default passwords from Mirai and adds two new entries—"root/5up" and "Admin/5up"—the factory defaults for Atheros wireless routers and access points. In addition, Hajime is capable of exploiting ARRIS modems using the password-of-the-day "backdoor" with the default seed.

Hajime does not follow a fixed sequence of credentials. From Radware's honeypot logs, we could conclude that the credentials used during an exploit change depending on the login banner of the victim. In doing so, Hajime increases its efficiency in successfully exploiting devices within a limited set of attempts. It thereby avoids the system account being locked or its IP being blacklisted for a set amount of time by the device.

Upon execution, Hajime prevents further access to the device through filtering ports known to be abused by IoT bots like Mirai:

- **TCP/23 (Telnet)** – the primary exploit vector of Mirai and most IoT botnets
- **TCP/7547 (TR-069)** – as first used in the DT attack by a Mirai variant
- **TCP/5555 (TR-069)** – alternate port commonly used in TR-069
- **TCP/5358 (WSDAPI)** – Web Service on Devices API is Microsoft's interoperable implementation of the open Device Profile for Web Services (DPWS) specification for embedded devices

At the same time, Hajime also tries to remove existing firewall rules with the name CWMP_CR. CWMP refers to the CPE WAN Management Protocol or TR-069. Removing any potential CWMP rules set by an ISP to allow specific management IPs or subnets that will be locked out leaving ISPs without control of the CPE device.

Besides locking down the device, Hajime opens up port UDP/1457 and a random higher port number (> 1024) for UDP and TCP. In doing so, it enables itself to use BitTorrent DHT and uTP from port UDP/1457 to build its peer-to-peer CnC network. The random higher port serves the purpose of the loader service used by the infection process to remotely download the malware onto new victims.

Hajime prefers the use of volatile file systems as its working directory, ensuring any indicator of compromise is gone after a device reboot. Hajime is not persistent. In other words, rebooting the device will clean it from infection—but only until the next one.

THAT'S HOW. BUT WHY?

Radware has explored how Hajime works. That leaves the "why:" What is the purpose of Hajime? Notably, no attacks have been attributed to Hajime, and it doesn't carry a payload to do so. Nevertheless, it is sophisticated, well designed and flexible enough to be repurposed in the blink of an eye.

There has been lots of speculation about the "grayness" of the author and the intent and purpose of Hajime. The motivation of the original author may never be known, but it's believed this botnet could be hijacked from its original owner. Sam and Ioannis from Rapidity Networks uncovered a vulnerability in the initial encryption implementation of Hajime and were able to reverse its messaging protocol, proving that such complex malware is not always without its vulnerabilities.

As previously noted, the vulnerability has been patched and updated, but a botnet of this size with a flexible backend and high potential for criminal behavior will certainly attract the attention of black hats. Translation: Whoever has the “private keys” to the botnet could decide its fate.

Because of its flexible and extensible nature, Hajime can easily be repurposed and leveraged to perform any or all of the following:

- **DDoS attacks.** In much in the same way Mirai included code for performing several DDoS attacks, an extension module for Hajime could easily be created that includes the same and more attack vectors to perform devastating DDoS attacks on command.
- **Massively distributed vulnerability scanning.** This would allow hackers to detect vulnerable, exposed public services and exploit them within hours after the disclosure of a new vulnerability. (As history has shown, most systems are not patched within a few hours) Custom exploit modules can be written in any language as long as they compile to a binary for one of the supported platforms, and can be distributed through the torrent overlay to be executed by tens or even hundreds of thousands of distributed nodes across the Internet.
- **Massive surveillance network.** The extension module could potentially tap into RTSP streams from cameras.
- **IoT bricker network.** Leveraging the work of BrickerBot, it would be a small and easy change to the atk program to perform a self-destructive sequence upon receiving a “Plan B” command through the CnC channel. A hacker could put a specific region or city in the dark by bricking all the infected devices corresponding to that region or city based on GeoIP.

For now, Hajime appears to remain under the control of its original author, and for the most part, the intentions are considered to be good. Still, questions persist as to why this supposed white knight keeps growing his botnet—searching and scanning aggressively for the next potential victim—and keeping all victims hostage. If the intentions are good, why not just leave the CWMP rules? Why not improve on what the ISP may have done inadequately? Why not make the firewall rules persistent—or keep them volatile but release the device? Instead, Hajime is used to keep devices hostage until reboot.

If Hajime offers a glimpse into the future of IoT botnets, let’s hope the IoT industry starts considering securing existing and new products. If not, our connected hopes and futures might depend on purging the threat the hard way—through grayhat vigilantes like the Janit0r of BrickerBot fame.

radware
WHEN THE BOTS COME MARCHING IN
A Closer Look at the Evolving Threat from Botnets, Web Scraping and IoT Zombies

DOWNLOAD *WHEN THE BOTS COME MARCHING IN: CLOSER LOOK AT THE EVOLVING THREAT FROM BOTNETS, WEB SCRAPING AND IOT ZOMBIES*

LEARN MORE AT DDOS WARRIORS

To know more about today’s attack vector landscape, understand the business impact of cyber-attacks orlearn 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.