WHEN THE BOTS COME MARCHING IN

A Closer Look at the Evolving Threat from Botnets, Web Scraping and IoT Zombies

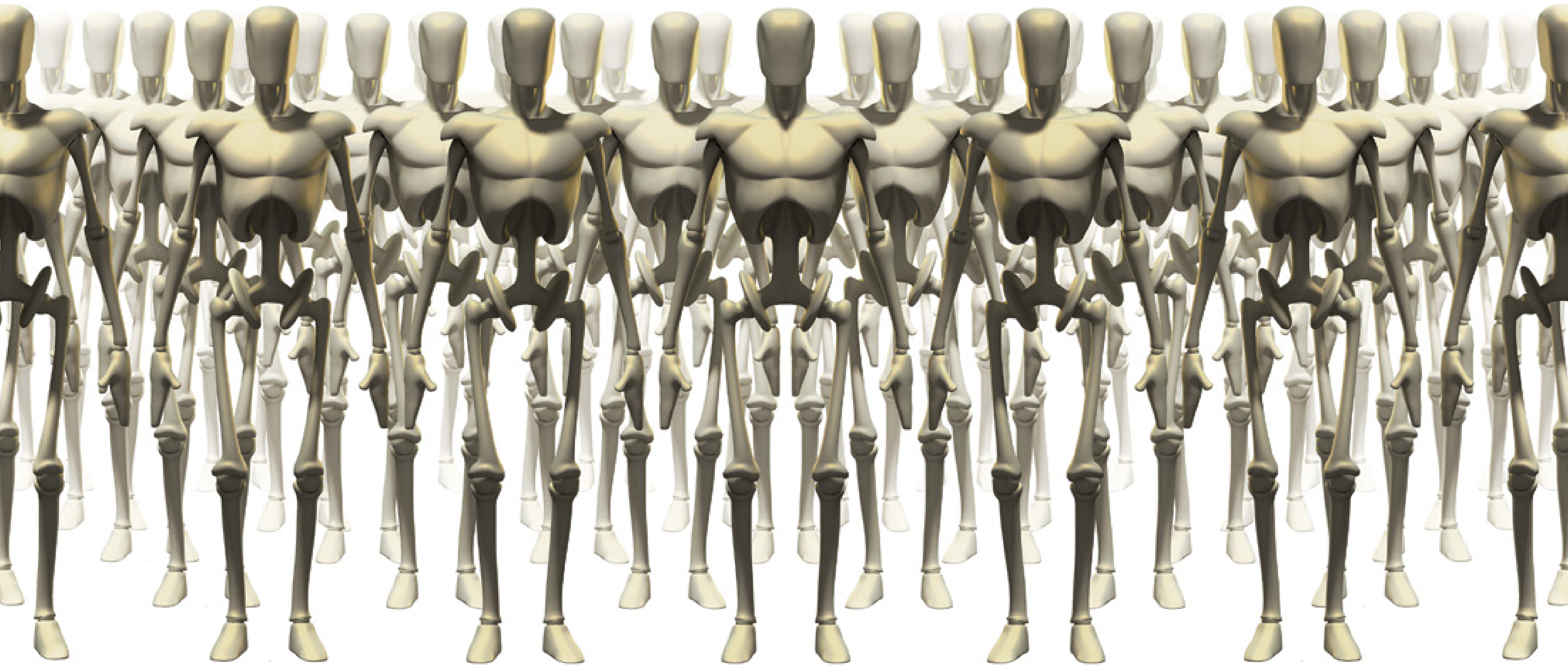
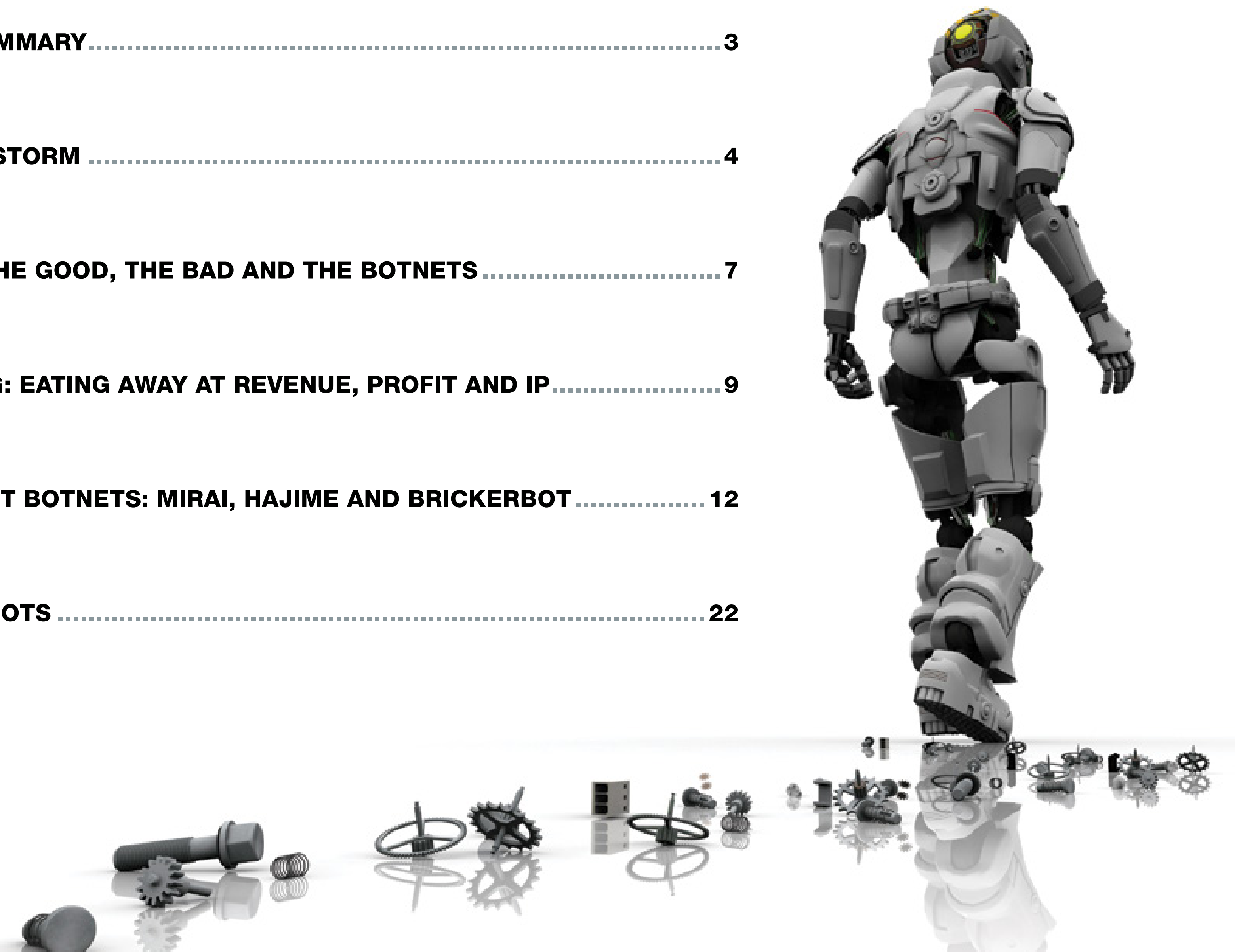


TABLE OF CONTENTS



EXECUTIVE SUMMARY

The information in this e-book may be out of date by the time you read it. That is how quickly and dramatically bots, botnets, web scraping and IoT threats are evolving.

Does that mean you shouldn't waste your time reading this? Far from it.

Nefarious bots—including those that exploit Internet of Things (IoT) devices as weapons of attack—are one of the fastest-growing and changing threats in the security landscape. The Mirai attacks in 2016 demonstrated what can happen when hackers deploy a bot army to take control of insecure IoT devices. Yet, those kinds of attacks may no longer be the worst-case scenario. Radware recently discovered a new type of bot—the BrickerBot, which aims to "protect" already infected devices through Permanent Denial-of-Service (PDoS). That's right: To ensure that infected IoT devices won't be drafted into an IoT botnet, BrickerBot completely shuts them down. They become nothing more than "bricks", devoid of both utility and value.

Make no mistake: This will be an ongoing tale as threats and detection and mitigation solutions continuously change. Build your knowledge and defenses by reading about the existing status quo, including:

- Technology trends and forces that made these threats possible
- Web scraping and how it can eat away at revenue, profit and IP
- Basics of bots (not all are bad)
- A closer look at three of the world's most noteworthy bots—at least so far

THE PERFECT STORM

The year 2016 brought attacks on Krebs, OVH and Dyn by the IoT botnet known as Mirai. Mirai was neither the first nor the most sophisticated IoT botnet, but it was highly effective in taking down its targets. These attacks represented a milestone in IoT botnet and DDoS history and served as a wakeup call to anyone responsible for safeguarding networks, systems and data.

Mirai prompted an inflection point in the IoT and DDoS threat landscape. However, it was a number of other trends and forces that enabled Mirai's "success:" large numbers of unsecure, connected devices, dependence on the cloud among businesses and individuals, and a thriving hacker economy that offers convenient access to a host of inexpensive tools for launching DDoS and other cyber-attacks.

Let's take a closer look at this "perfect storm" of trends that have fueled risks and are poised to raise the stakes even higher:

IOT DEVICES: RAPID GROWTH...

The term "Internet of Things" was coined in 1999. It wasn't until 2014-2015 that IoT reached its own inflection point, with component costs falling and business and consumer demand accelerating. From light bulbs and washing machines to medical devices, a growing array of "things" connected to the Internet. Meanwhile, machine-to-machine solutions started going mainstream.[1]

By 2016, the number of connected things was nearly double the number of connected users and the volume of connected devices grows much more quickly than the Internet population. Depending on which source is consulted, the number of IoT units installed could reach as many as 20 billion by 2020.

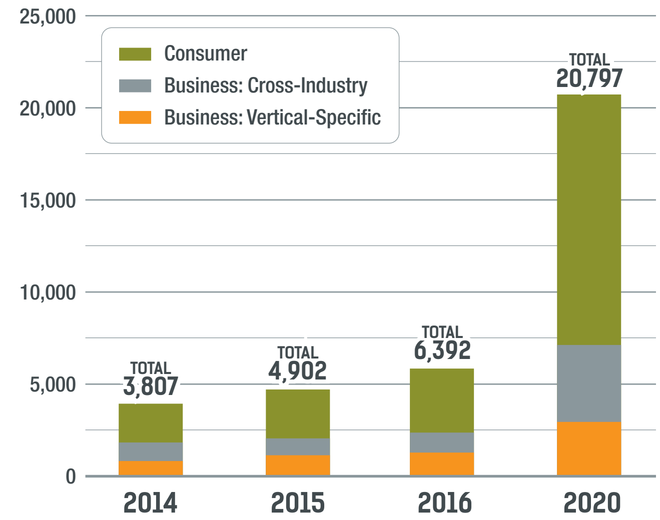


Figure 1: Internet of Things Units Installed Base by Category (Millions of Units)
source: http://www.gartner.com/newsroom/id/3165317

...AND LACKLUSTER SECURITY

What makes IoT devices desirable targets for cyber-attacks? It comes down to four main factors:

1. **Stripped-down OS.** These devices often run on the Linux operating system—but use an embedded or stripped-down version that is comparatively easy to compromise with malware.
2. **Unfettered access.** When "things" are Internet accessible, their access is usually unfettered by filtering or limitations on bandwidth.
3. **Lack of basic security.** With their barebones OS and processing power, these devices simply don't have enough capacity for standard security capabilities like auditing. The result? Device owners won't even notice most compromises.
4. **Reused components.** Device manufacturers often reuse portions of hardware and software in various devices. Though intended to save engineering time, this practice also results in default passwords and vulnerabilities being shared across not just device classes, but also manufacturers.

CLOUD MIGRATION AND EMERGING SERVERLESS COMPUTING

As more businesses migrate to the cloud, more core applications are hosted in public clouds, continuously increasing the number of targets available for attackers. As the world moved from virtualized machines (Infrastructure-as-a-Service) to applications (Software-as-a-Service), the next big thing will be serverless computing (Function-as-a-Service).

Serverless computing/FaaS is really just the natural evolution of the API economy, which is already upon us. As hyper-scale cloud applications came to leverage micro-service architectures, the API economy is about externalizing these internal micro-services as publicly consumable APIs.

1 Source: http://radar.oreilly.com/2014/06/the-internet-of-things-inflection-point.html

As serverless computing sets in, more cloud applications will become inherently dependent on a multitude of APIs. That, in turn, will give rise to a complex, interconnected world of functions—a level of interdependency that exceeds even those that made it possible for the Dyn attack to inflict collateral damage.

In short, turning a single point of public access to a cloud application into a modular set of functions will increase the number of targets. Further, it will increase the blast radius of a single compromised function or service from a few to many impacted services and businesses. It gives attackers more to shoot at—and bigger payoffs when they hit their targets.

MATURING OF THE HACKER ECONOMY

Today there are vibrant online marketplaces where just about anyone—even those with very limited technical knowhow—can buy tools to execute an attack. Cryptographic currencies enable untraceable digital payments, while old-fashioned economics is driving the growth of these marketplaces. Demand for services now outpaces supply, and DDoS-as-a-Service providers can bring in more than $100,000 annually.

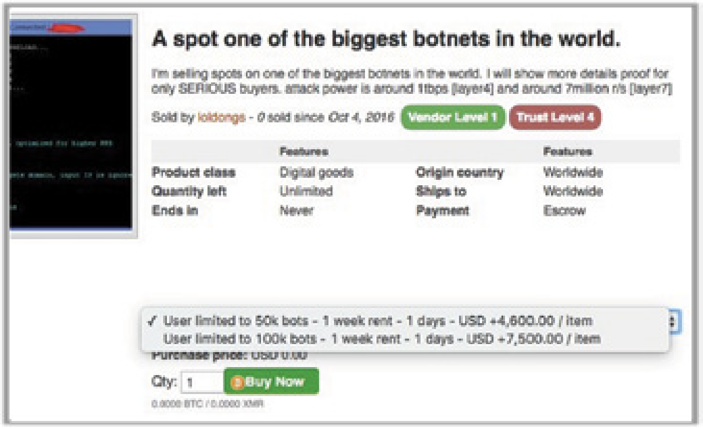


Figure 2: Example of an attack available for purchase

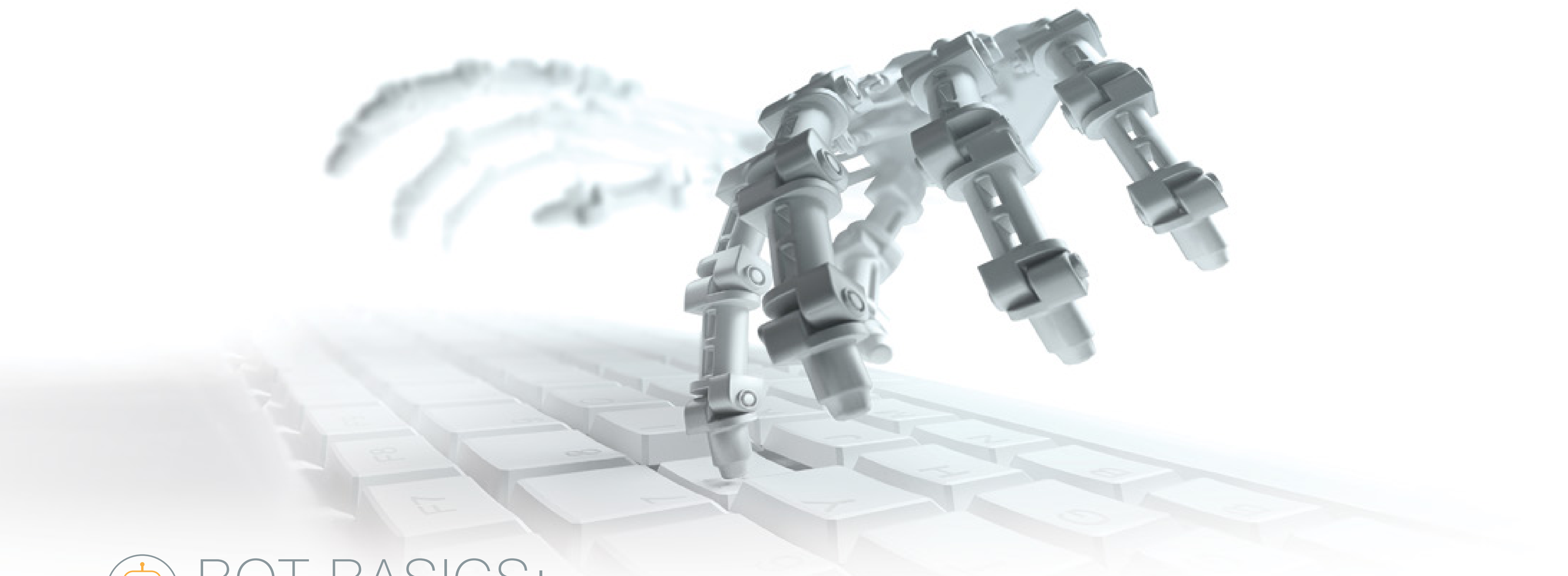
Purchasing an attack can be surprisingly inexpensive. On the Clearnet, for as little as $19.99 a month, an attacker can run 20-minute bursts for 30 days utilizing a number of attack vectors like DNS, SNMP, SYN and slow GET/POST application-layer DoS attacks. All an attacker has to do is create an account, select a plan, pay in Bitcoin and access the attack hub to target the victim by port, time and method. More advanced and larger botnets are also available for sale on the Darknet.[2]

What motivates people to pay for attacks? Radware has identified three primary drivers:

- **Profit.** Launching ransom campaigns is a way to make quick money.
- **Distraction.** In a multi-vector attack, a DDoS attack provides a valuable smokescreen to hide more targeted attacks and/or data exfiltration.
- **Disruption and defacements.** Other attacks are launched by hacktivists that are seeking to apply heat to organizations with whom they disagree—or by organizations looking to take down their competitors.

For more insights about cyber ransom and the hacking economy, see Radware's publication, *Cyber Ransom Survival Guide: The Growing Threat of Ransomware and RDoS – and What to Do About It*.

2 For more on the attack marketplace, please see this Radware blog post.



BOT BASICS: THE GOOD, THE BAD AND THE BOTNETS

WHAT'S A "BOT"?

Generally speaking, "bot" is short for "robot". In the context of the Internet, the full term is "Web robot" or "Internet robot"—a computer system programmed to perform a set of automated tasks. Bots may also be referred to as "zombies."

As Figure 3 illustrates, some bots perform positive, helpful tasks, such as gathering intelligence on websites and services; others are created by threat actors who use them to take over or infect compute devices. Targets of bot attacks can range from personal computers, smartphones and tablets to servers and connected "things."

BOTS ARE BIG PLAYERS IN THE ONLINE WORLD [3]

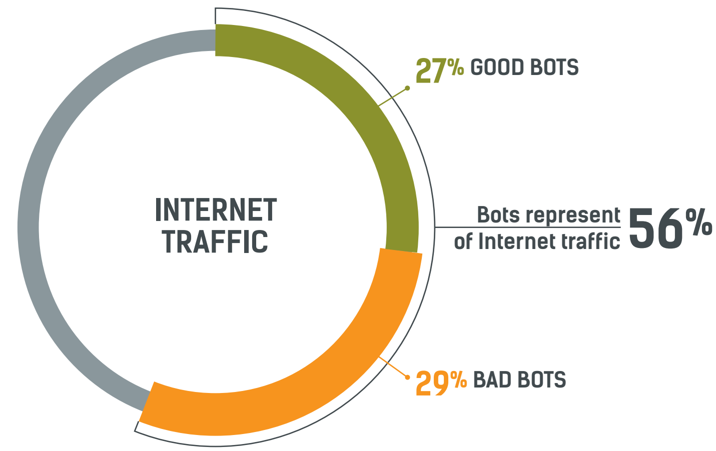
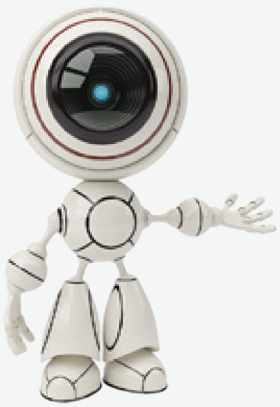


Figure 3: Bots in the online world

3 https://areyouahuman.com/downloads/GoodBotsvBadBots_FINAL.pdf



GOOD BOTS

Help make the Internet better.

Good bots support:

- Web indexes
- Travel industry aggregation sites
- News aggregation
- Automated trading
- Protection from bad bots

Examples:

- Spider bots
- Media/RSS/data bots
- Vigilante™ bot
- Trading bots
- Chatterbots

BAD BOTS

Create personal and business risk on the Internet.

Bad bots:

- Cause damage through DDoS
- Steal data through web crawling and scraping
- Scan applications, servers and networks for exploits
- Execute spam and phishing campaigns
- Enable ad click fraud

Examples:

- Clickerbot
- Hacker bot
- Download bot
- Bitcoin mining bot
- Spam bot
- Malware/virus bot
- Scraper bot

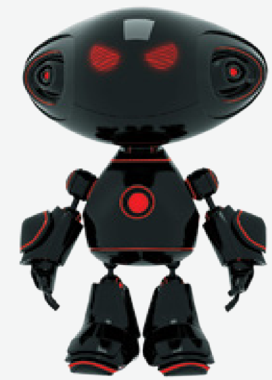


Figure 4: Examples of good and bad bots

WHAT'S A BOTNET?

A botnet is a group of bots that are overseen by a command and control server (we'll shorten that to "CnC," though it's also referred to as "C&C "or "C2"). Each botnet has one CnC (or more for availability) that orchestrates and controls all of the bots within the same botnet. Each bot in the botnet is programmed to "call home" to the CnC, which then provides instructions, or commands, to them. Botnets are the ultimate attack tools. Here's why:

- Bots aren't directly associated with the attackers
- Bots can be automated via CnC servers
- Bots are geographically distributed
- Bots are disposable and easy to replace if needed
- Bots are flexible and can be used in a wide range of nefarious activities

As Figure 5 shows, botnets are growing in size—with larger botnets and smarter devices—resulting in more sophisticated attacks.

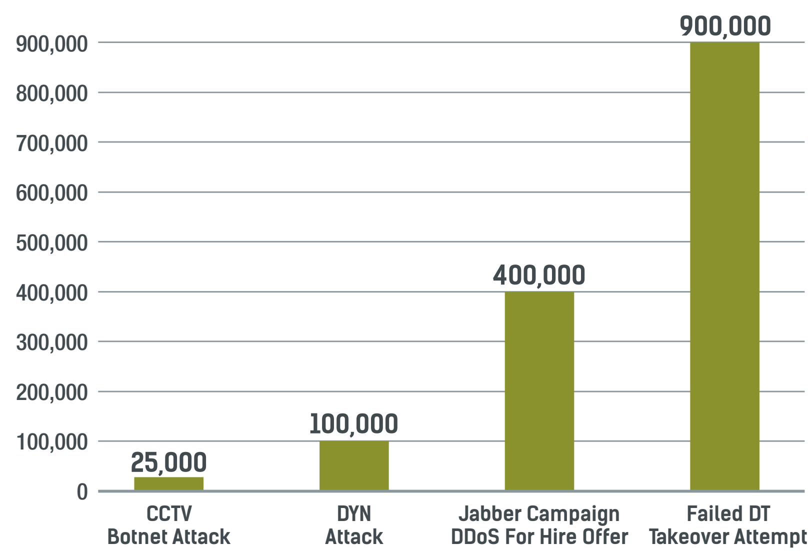


Figure 5. Estimated IoT botnet sizes 2016
Based on research from Radware's Emergency Response Team

WEB SCRAPING: EATING AWAY AT REVENUE, PROFIT AND IP

Web scraping refers to a type of software tool that harvests data from websites and uses it for a variety of purposes. There are five main use cases for web scraping:

- Content scraping (the practice of lifting original content from a legitimate website and posting it on another without the owner's knowledge or permission)
- Research
- Price comparison
- Data monitoring (weather, stocks, etc.)
- Website change detection

Content scraping poses tremendous risks to an organization, including theft of intellectual property and/or data. In addition, the bot performing the content scraping could perform so many continuous requests that it leads to a Denial-of-Service (DoS) situation. Also, a company could lose profit thanks to aggregators and price comparison websites, or due to information leakage.

IF A BROWSER CAN RENDER IT, IT CAN BE SCRAPED

What can be done to prevent scraping of online assets? Some information that can be considered sensitive has to be public in order to be useful. Common examples include airfares, hotel rates and lists of physicians. In some cases, sites try to obfuscate their data. Using dynamic grids, AJAX and/or WebSockets to download the actual data, they aim to make it much more difficult to scrape data records.

Whatever a browser can render to the screen will be retained in memory as part of a structured Document Object Model (DOM) and that content will be accessible from scripting or programming libraries.

Ironically, most of the commonly used scraping tools were designed for another purpose entirely: quality assurance. Selenium and similar tools are used for web application testing. They enable developers to simulate and automate user interactions, allowing test responses from the web app. But that same functionality makes it possible to use Selenium and similar tools to automate scraping of any data that's publicly available. Headless or real browser clients can also be used to make the bot detection even harder. These techniques help in mimicking user behavior, passing challenges and thwarting other bot-detection algorithms.

SCRAPING SERVICES: JUST A GOOGLE SEARCH AWAY

Just as it has become easy to acquire DDoS-as-a-Service, it is also fast and simple to access online services for web scraping.

ABUSED BY BOTS: SCRAPING STORIES FROM THE FRONT LINES

Securing online bargains is among the most common uses for web scraping. Web-scraping tools make it relatively easy to track online prices and create numerous requests once a price drop is identified. Compared to humans, bots are far more efficient in generating requests, producing multiple requests per minute (whether real or fake). A possible result: emptying the online store inventory so gray marketers can resell the goods at a higher price.[4]

Chances are, you've experienced this firsthand. Think about the last time you heard about an upcoming concert. The very moment tickets were available online, you tried to buy some. Yet all the good seats were already gone! Later, you found those seats—at five to eight times the cost—on ticket-broker websites. You can thank web scraping for that.

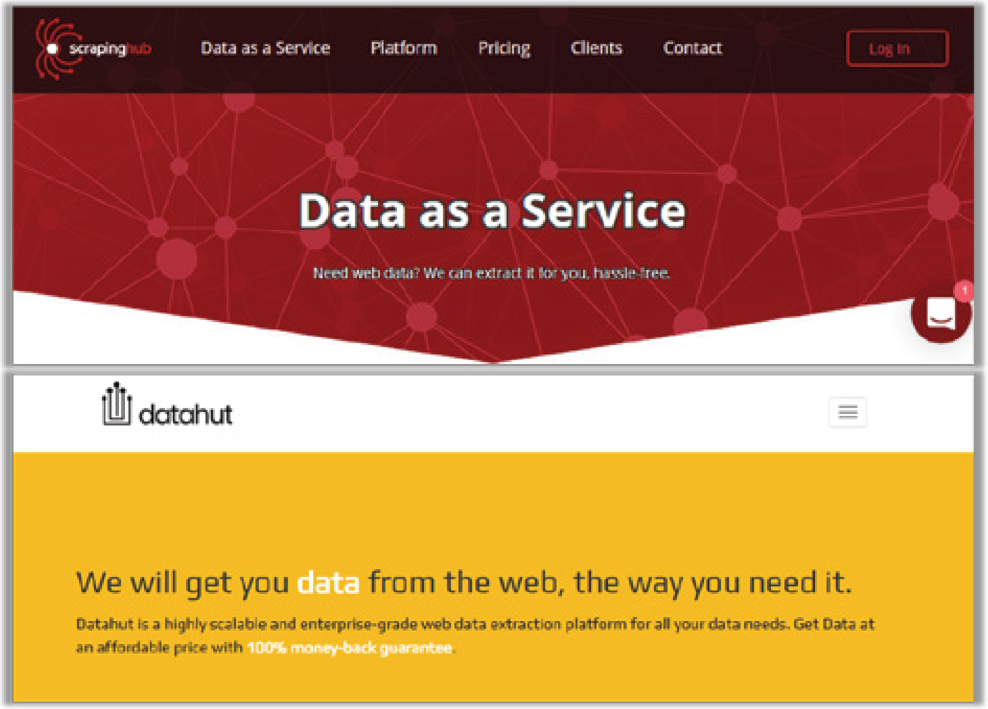


Figure 6: Example of web scraping services

4 Gray marketers buy and sell products outside of the manufacturer's authorized trading channels.

SOME MORE EXAMPLES...

Airlines are another common target of web scraping. For one Radware customer—a major US-based airline—this type of cyber-attack occurred with alarming frequency. Bots were programmed to "scrape" certain flights, routes and classes of tickets. With the bots acting as faux buyers—continuously creating but never completing reservations on those tickets—the airline was unable to sell the seats to real customers. In essence, the airline's inventory was held hostage, and a growing number of flights were taking off with empty seats that could have been sold.[5]

In the UK, a name-brand website was operating from behind a paywall. The company was not concerned about web scraping until they found their entire website was being scraped and offered for free in a Chinese hosting site.

Many years ago, Radware helped an online store that was plagued by competitors putting $99,000 of merchandise into their shopping carts and going to the checkout phase. The snag? Even though the competitors didn't actually complete the checkout process, the inventory appeared depleted. For real customers, everything was showing as "out of stock" and had to be back-ordered. Beyond that, this online store was finding that its competitors were visiting its site to do price comparisons. After Radware's web performance optimization solution blocked the scraper bots, the site's web traffic bandwidth decreased by 66%. That demonstrates how much bandwidth had been serving bad bots! With those bots out of the mix, website page speed and performance doubled.

Figure 7: Another example of a web scraping service

5 Source: Radware's 2015-2016 Global Application & Network Security Report

THE RISE OF IOT BOTNETS: MIRAI, HAJIME AND BRICKERBOT

As described earlier, bots can be commandeered into armies known as botnets, and the Internet of Things is fraught with connected devices offering a staggeringly low level of security. It's not hard to imagine what happens when the two are combined—infecting tens or hundreds of thousands of IoT devices with the same bot and then transforming those devices into a massive, distributed fleet of IoT zombies.

It's a concept that Radware predicted for years and that was vividly brought to life by the Mirai botnet in October 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 effect great harm.**

Most recently, Radware's own Pascal Geenens discovered an entirely new breed of IoT botnet. Coined BrickerBot, this bot has another objective entirely. While Mirai quickly harnessed an army for attacks and Hajime seems intent on quietly building, but not taking action with, its own army, BrickerBot has a "nobler" purpose. Its author, known as the Janit0r, purports to be protecting insecure IoT devices through Permanent Denial-of-Service (PDoS). Rather than simply kicking out other bots and taking over devices, BrickerBot "bricks" them. It thereby eliminates the risk that they'll be drafted a part of an IoT zombie army. Of course, it also means they can no longer function as anything other than paperweights. **Think of BrickerBot as the vigilante bot. It bricks infected IoT devices—issuing a stark wakeup call about the need for IoT security.**

Let's analyze these botnets and how they inflict their damage. Although they are sure to evolve and gain competition from new entrants, understanding these three bots provides important insights into the war for IoT market share that is ongoing on the Internet and will help us better understand 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 OVH3. 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.

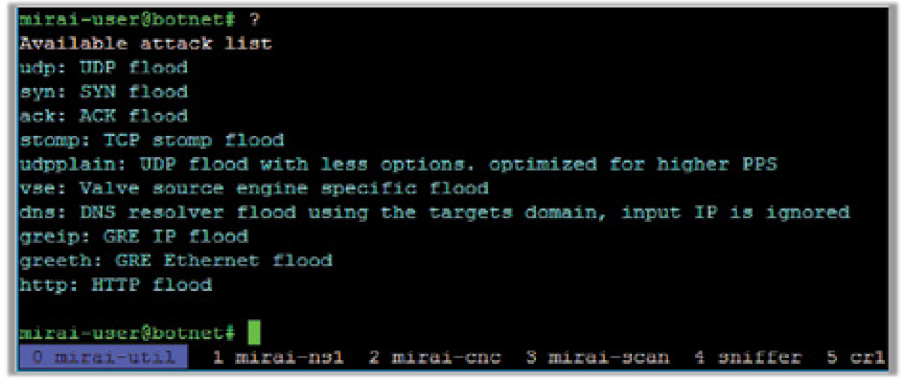


Figure 8: Mirai botnet’s command set.

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.[6] (While Mirai is the Japanese word for “the future,” Hajime can mean “beginning.”[7]) 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.

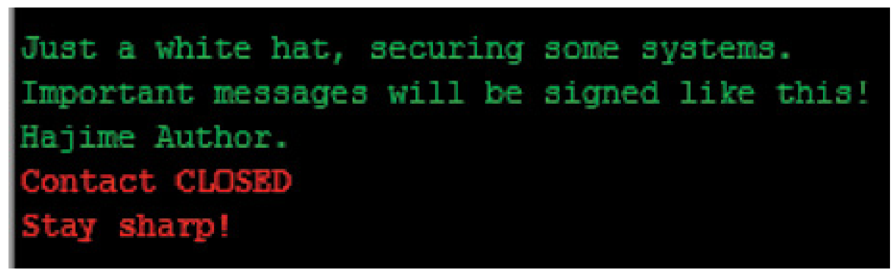


Figure 9: Message periodically displayed on the terminal by Hajime.

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.”

6 https://security.rapiditynetworks.com/publications/2016-10-16/hajime.pdf

7 https://www.extremetech.com/internet/248087-meet-hajime-iot-botnet-built-vaccinate-devices-mirai

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 10). According to Kaspersky, the peer-to-peer network had reached almost 300,000 devices by April 25, 2017.[8]

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.

Figure 10. Hajime's geographic reach, with every dot accounting for a unique IP that Radware was able to identify as an infected device

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

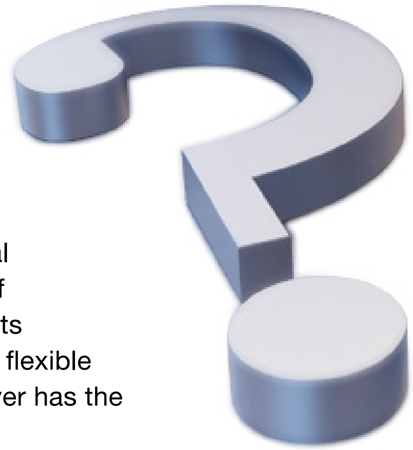
8 https://securelist.com/blog/research/78160/hajime-the-mysterious-evolving-botnet/

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.

BRICKERBOT: A CURE WORSE THAN THE DISEASE?

On April 5, 2017, Radware announced a stunning discovery. Over a four-day period, our honeypot recorded 1,895 Permanent Denial-of-Service (PDoS) attempts performed from several locations around the world. The sole purpose: to compromise IoT devices and corrupt their storage. Besides this intense, short-lived attack (BrickerBot.1), Radware's honeypot recorded attempts from a second, very similar bot (BrickerBot.2), which started PDoS attempts on the same date. Both bots were discovered less than one hour apart—with the second one exhibiting lower intensity and enhanced thoroughness and its location(s) concealed by TOR egress nodes.

Compromising a Device

The BrickerBot PDoS attack uses Telnet brute force—yes, the same exploit vector used by Mirai and Hajime—to breach a victim's devices. BrickerBot does not try to load a binary. Rather, it scans for several well-known ports and, according to its author, the Janit0r, tries to secure the device in a less destructive way. If this fails, BrickerBot reverts to what the Janit0r calls "Plan B," with "B" meaning "brick."

Corrupting a Device

Upon successful access to the device, the PDoS bot performs a series of Linux commands that would ultimately lead to corrupted storage, followed by commands to disrupt Internet connectivity and device performance and wipe all files on the device (see figure 11 for the exact sequence of commands performed by the PDoS bots).

```
fdisk -l
busybox cat /dev/urandom >/dev/mtdblock0 &
busybox cat /dev/urandom >/dev/sda &
busybox cat /dev/urandom >/dev/mtdblock10 &
busybox cat /dev/urandom >/dev/mmc0 &
busybox cat /dev/urandom >/dev/sdb &
busybox cat /dev/urandom >/dev/ram0 &
fdisk -C 1 -H 1 -S 1 /dev/mtd0
w
fdisk -C 1 -H 1 -S 1 /dev/mtd1
w
fdisk -C 1 -H 1 -S 1 /dev/sda
w
fdisk -C 1 -H 1 -S 1 /dev/mtdblock0
w
route del default;iproute del default;ip route del default;rm -rf /* 2>/dev/null &
sysctl -w net.ipv4.tcp_timestamps=0;sysctl -w kernel.threads-max=1
halt -n -f
reboot
```

Figure 11. Command sequence of BrickerBot.1

Among the special devices targeted:

- **/dev/mtd – Memory Technology Device** – a special device type to match flash characteristics
- **/dev/mmc – MultiMediaCard** – a special device type that matches memory card standard, a solid-state storage medium

TARGETS

The use of the “Busybox” command combined with the MTD and MMC special devices means this attack is targeted specifically at Linux/BusyBox-based IoT devices that have their Telnet port open and exposed publicly on the Internet. These are matching the devices targeted by Mirai, Hajime or related IoT botnets.

The PDoS attempts originated from a limited number of IP addresses spread around the world. All devices are exposing port 22 (SSH) and running an older version of the Dropbear SSH server and outdated firmware. Most of the devices were identified by Shodan as Wireless CPE devices, Wireless Access Points and Wireless Bridges with beam directivity.

BRICKERBOT.3: THE JANIT0R COMES BACK—WITH A VENGEANCE

In May 2017, Radware discovered yet another new version of the BrickerBot PDoS attack (BrickerBot.3) with a new command sequence on a different honeypot location (see figure 12 for the new PDoS attempt sequence). An hour after finalizing the report on the first 12 hours of BrickerBot.3, the article on the potential author of BrickerBot and the comments by this author were published by Catalin Cimpanu on his blog, claiming to have bricked two million devices to date.

THE FIRST 12 HOURS

During the first 12 hours of the attack, a total of 1,118 PDoS attempts were recorded. The attacks all originated from a limited number of Clearnet IP addresses. ZoomEye and Shodan searches based on the source IPs of the attacks revealed all of them running an outdated version of the Dropbear SSH server.

The attacks started at 12:00 GMT on April 21, and in the first 12 hours, the number of bots performing the attacks grew up to 15 (see figure 13).

The devices used to perform the PDoS attacks on our honeypot do not correspond to the devices from BrickerBot.1. Although BrickerBot.1 was also abusing a limited number of Clearnet-connected devices to perform its attack, there is no immediate correlation between the two.[10]

HITTING WHERE IT HURTS: SIERRA ISP

On April 10, 2017, customers of Sierra Tel, a small ISP in California, experienced a disruption in Internet and phone service. Recognizing it as a modem issue, Sierra’s initial response was to have customers physically bring their Zyxel HN-51 modems to be swapped for new ones.

But the number of malfunctioning modems surged—and the supply of new devices dried up. Meanwhile, the Janit0r reached out to Catalin Cimpany, a journalist from Bleeping Computer, to claim credit for the incident, which had failed to make national news until that point. It took two weeks for Sierra Tel to fully address the issue and restore all customers’ service.[9]

As the Bleeping Computer article notes, “It’s quite possible that Sierra Tel’s Zyxel modems went offline as the result of a secret turf war waged among various families of IoT malware, such as Mirai, BrickerBot, Hajime, Wifatch, Gafgyt, Imeij and others. As more and more IoT malware families emerge, they will eventually cause more problems like the Sierra Tel incident.”

```
busybox cat /dev/urandom >/dev/mtdblock0 &
busybox cat /dev/urandom >/dev/sda &
busybox cat /dev/urandom >/dev/mtdblock10 &
busybox cat /dev/urandom >/dev/mmc0 &
busybox cat /dev/urandom >/dev/sdb &
busybox cat /dev/urandom >/dev/ram0 &
busybox cat /dev/urandom >/dev/mtd0 &
busybox cat /dev/urandom >/dev/mtd1 &
busybox cat /dev/urandom >/dev/mtdblock1 &
busybox cat /dev/urandom >/dev/mtdblock2 &
busybox cat /dev/urandom >/dev/mtdblock3 &
fdisk -C 1 -H 1 -S 1 /dev/mtd0
w
fdisk -C 1 -H 1 -S 1 /dev/mtd1
w
fdisk -C 1 -H 1 -S 1 /dev/sda
w
fdisk -C 1 -H 1 -S 1 /dev/mtdblock0
w
route del default;iproute del default;ip route del default;rm -rf /* 2>/dev/null &
sysctl -w net.ipv4.tcp_timestamps=0;sysctl -w kernel.threads-max=1
halt -n -f
reboot
```

Figure 12. BrickerBot 3

9 https://www.bleepingcomputer.com/news/security/us-isp-goes-down-as-two-malware-families-go-to-war-over-its-modems/
10 For complete disclosure and transparency, the attacks were detected by a different honeypot than the one that detected the BrickerBot.1 and BrickerBot.2 attacks.

The devices that perform the attack are spread around the globe, with no concentration in a specific region. Nor do they correlate to the locations of the BrickerBot.1 sources (see figure 14).

In line with BrickerBot.1 and BrickerBot.2, this bot is also using the Mirai exploit vector to compromise the target. Any Busybox-based Linux device that has Telnet exposed publicly and has factory default credentials unchanged is a potential victim.

BRICKERBOT.4

Between 5:22 pm and 8:44 pm GMT, the same honeypot detected yet another, very similar sequence of commands. The attack was only attempted from a single device located on the Clearnet. Upon investigation, it was discovered to have an outdated version of the Dropbear SSH server. This isolated bot performed 90 attacks and was not seen again after.

Compared to Bricker.Bot3, the BrickerBot.4 command sequence has fewer block devices it tries to corrupt (see figure 15).

THE JANIT0R: A WHITE KNIGHT?

A few hours after the discovery of BrickerBot.3 and BrickerBot.4, Catalin Cimpanu published his article on the author of BrickerBot. A person who went by the name of "Janit0r" on HackForums.net alluded to being the author of BrickerBot. Based on the Janit0r's discussion, BrickerBot is more complex—and very likely much larger—than we initially believed it to be.

"Like so many others I was dismayed by the indiscriminate DDoS attacks by IoT botnets in 2016. I thought for sure that the large attacks would force the industry to finally get its act together, but after a few months of record-breaking attacks it became obvious that in spite of all the sincere efforts the problem couldn't be solved quickly enough by conventional means. I consider my project a form of 'Internet Chemotherapy'. I sometimes jokingly think of myself as 'The Doctor'. Chemotherapy is a harsh treatment that nobody in their right mind would administer to a healthy patient, but the Internet was becoming seriously ill in Q3 and Q4/2016 and the moderate remedies were ineffective."

The Janit0r, who claims to have bricked 2 million devices (and counting) since November 2016

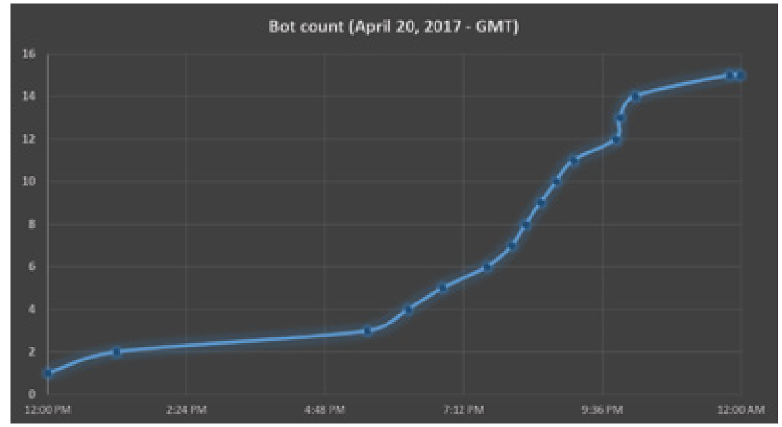


Figure 13. BrickerBot.3 growth timeline

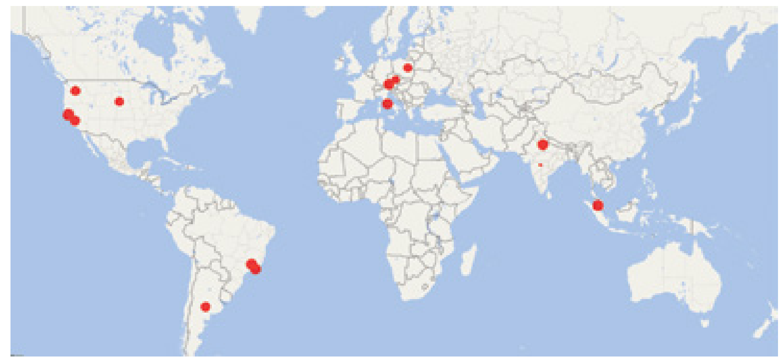


Figure 14. Geographic distribution of devices used by BrickerBot.2 to perform the attacks

```
fdisk -l
busybox cat /dev/urandom >/dev/mtdblock0 &
busybox cat /dev/urandom >/dev/sda &
busybox cat /dev/urandom >/dev/mtdblock10 &
busybox cat /dev/urandom >/dev/mmc0 &
busybox cat /dev/urandom >/dev/sdb &
busybox cat /dev/urandom >/dev/ram0 &
fdisk -C 1 -H 1 -S 1 /dev/mtd0
w
fdisk -C 1 -H 1 -S 1 /dev/mtd1
w
fdisk -C 1 -H 1 -S 1 /dev/sda
w
fdisk -C 1 -H 1 -S 1 /dev/mtdblock0
w
route del default;iproute del default;ip route del default;rm -rf /* 2>/dev/null &
sysctl -w net.ipv4.tcp_timestamps=0;sysctl -w kernel.threads-max=1
halt -n -f
reboot
```

Figure 15: BrickerBot.4 command sequence

For the time being, it seems as though the Janit0r will continue waging BrickerBot attacks, at least until officials and hardware vendors take definitive action to improve the state of IoT security.

Like BrickerBot.1, BrickerBot.3 and BrickerBot.4 are targeting Busybox-based Linux devices, typically IoT devices such as IP cameras and DVRs. As mentioned in Radware's BrickerBot blog from April 13, 2017, Radware tested the BrickerBot.1 command sequence and assessed its impact on a real-life device: a Sricam AP003 Metal Gun Type Waterproof Outdoor Bullet IP camera, which is known to be "supported" by Mirai.

Although the name connotes robustness, upon running the sequence of commands from BrickerBot.1, the camera got disconnected from the network and upon reboot was no longer responding. Factory reset through a special button on the camera did not recover the IoT device. It was effectively bricked.

Figure 16: Excerpt of a conversation that includes JanitOr

The bad news? BrickerBot can be quite effective. The good news? We now know it only works on previously infected IoT devices.

UNDERSTANDING HOW BRICKERBOT WORKS

Radware analysis has revealed insights into how BrickerBot may be identifying and targeting victims. There is no evidence of active scanning before the attacks—at least not from the same devices that were used during the attacks. From the BrickerBot data, we realized that only a limited number of devices performed attacks. What's more, most did so exactly 90 times before they simply disappeared. This magic number of 90 was even more apparent in the second BrickerBot wave.

One big question remained: How does BrickerBot detect infected devices? Radware soon realized the answer was right in front of us all the time. Radware's honeypots are built to listen on known exploit ports. What if BrickerBot does the same? What if it passively detects exploit attempts by botnets, such as Mirai variants and Hajime? Instead of actively scanning the Internet for new victims, it could just be listening on port TCP/23 (telnet) and port TCP/7457 (TR069) for scans from infected IoT devices. For sure, this technique is both stealthier and more efficient than actively scanning most of the Internet from each device.

Using one of the more recently discovered BrickerBot source IP addresses, Radware performed a TCP connection test on port TCP/23. The connection got established and was immediately closed by the server. Within seconds, the honeypot deployed on the same Internet connection started revealing BrickerBot sequences from the same BrickerBot source IP we just dialed. The same BrickerBot kept attacking until it reached exactly 90 attempts—and then left.

Further testing of several BrickerBot-infected devices from previous attack waves showed that more ports are open. Telnet to port 7547 and 19058 consistently triggered the BrickerBot attacks—suggesting that the bot is listening on most of the known ports used by IoT bot exploits. In subsequent BrickerBot attempts, Radware noticed a slightly different sequence. This might have to do with changes in our honeypot code or improved BrickerBot code by Janit0r.

Figure 17. Geographic distribution of BrickerBot (1 dot = 1 bot, with 223 bots in total)

BRICKERBOT: THE BOTTOM LINE

The BrickerBot botnet has gone global. Radware has identified 223 active nodes (see figure 17). Using Shodan.io and zoomeye.org queries, it appears that nearly all devices are running an older version of the Dropbear SSH server. Most seem to have outdated firmware—some with hostnames 'HACKED-ROUTER-HELP-SOS-DEFAULT-PASSWORD' and 'HACKED-ROUTER-HELP-SOS-WAS-MFWORM-INFECTED' indicating the devices known to be vulnerable.

Aside from its global footprint, BrickerBot is quite different from most botnets. As we've explained, the typical botnet functions like an army of soldiers marching to the orders of their "general" (the CnC). Each instance of BrickerBot is more akin to an independent sleeping bear that takes action only when aroused by an infected IoT device. Once the BrickerBot "bear" is awakened, it will make 90 attempts to brick the infected IoT device. But we believe it can only be awakened by compromised IoT devices—with typical indicators of compromise by Mirai, Hajime or another IoT bot infection serving as the wake-up call.

In other words, keep your devices clean and secure and you won't have to worry about amassing a large collection of paperweights.

BEATING THE BOTS

WHILE THEY'RE DUKING IT OUT FOR CONTROL OF THE INTERNET OF THINGS…

MIRAI: THE BAD

- Most powerful botnet seen to date
- New level of DDoS attacks
- Potential for multiple Tbps attacks
- Unsophisticated, easy-to-herd new bots, impacting the DDoS-as-a-Service economy

HAJIME: THE GOOD (AT LEAST FOR NOW!)

- Takes insecure IoT devices hostage to keep them from taking part in DDoS botnets
- Sophisticated—offering a glimpse into the future of IoT bots and botnets
- Aggressively scans and infects
- Keeps CnC channel open for updates and new extensions
- True purpose remains a mystery

BRICKERBOT: THE VIGILANTE

- Destroys insecure IoT devices to keep them from taking part in DDoS botnets
- Only attacks devices already compromised by other bots

…HERE'S HOW TO PROTECT YOUR IOT DEVICES

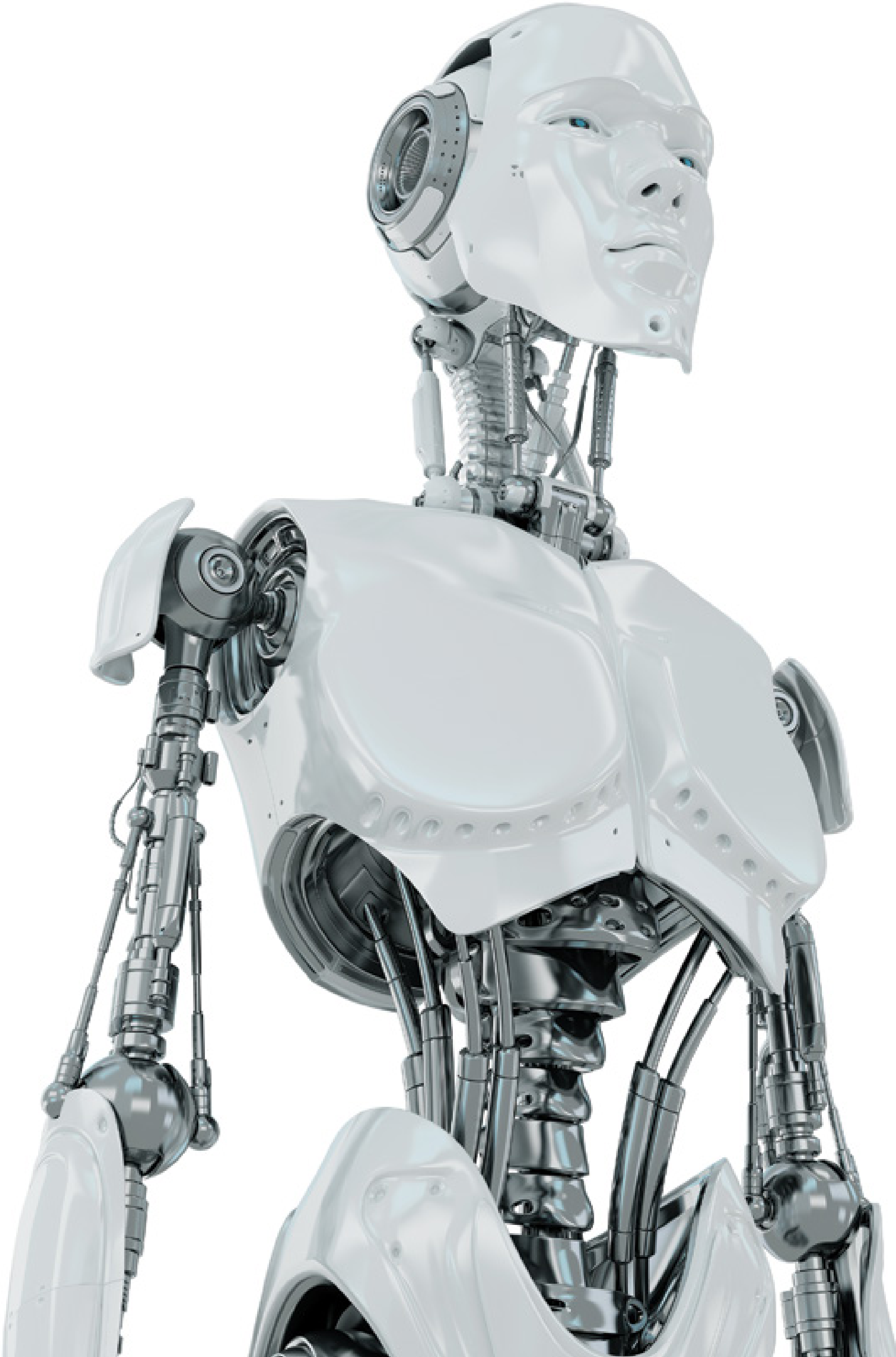
Be sure to:

- Change every device's factory default credentials.
- Disable Telnet access to every device.
- Check and upgrade firmware often.
- Brace and prepare yourself for more and larger DDoS attacks in the near future.
- Have an adequate incident response plan in place—and perform regular war games to train your people and test and asses your plan's effectiveness.
- Use Network Behavioral Analysis to detect anomalies in traffic and combine this with automatic signature generation for fast, effective mitigation.
- Use user and entity behavior analytics to spot granular anomalies in traffic early.

WHAT NOW?

The landscape of threats from botnets, web scraping and IoT zombies is dynamic and increasingly complex. It would be a disservice to provide a list of solutions, as the solutions have to remain as dynamic as the challenges they must mitigate.

KNOWLEDGE IS POWER. We encourage you to continue reading and learning and stay engaged with Radware. As our discovery of BrickerBot attests, we stay on the cutting edge of emerging and evolving cyber-security threats.



radware

This document is provided for information purposes only. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law. Radware specifically disclaims any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. The technologies, functionalities, services, or processes described herein are subject to change without notice.