DDoS in the IoT: Mirai and Other Botnets

Constantinos Kolias, George Mason University
Georgios Kambourakis, University of the Aegean
Angelos Stavrou, George Mason University
Jeffrey Voas, IEEE Fellow

The Mirai botnet and its variants and imitators are a wake-up call to the industry to better secure Internet of Things devices or risk exposing the Internet infrastructure to increasingly disruptive distributed denial-of-service attacks.

The ubiquity and increasing popularity of the Internet of Things (IoT) have made IoT devices a powerful amplifying platform for cyberattacks. Given the recent headline-making severity and frequent recurrence of security incidents involving such devices, they've clearly become the new weakest link in the security chain of modern computer networks. IoT devices might be the feeble brother of desktop systems, yet what they lack in computational capabilities they make up for in numbers. Moreover, because they're constantly connected to the Internet and seemingly permeated with flaws—in many cases the outcome of naive security configurations—they constitute low-hanging fruit for hackers. The large volume, pervasiveness, and high vulnerability of IoT devices have attracted many bad actors, particularly those orchestrating distributed denial-of-service (DDoS) attacks.

"THE FUTURE" IS HERE

A recent prominent example is the Mirai botnet. First identified in August 2016 by the whitehat security research group MalwareMustDie,[1] Mirai—Japanese for "the future"—and its many variants and imitators have served as the vehicle for some of the most potent DDoS attacks in history.

0018-9162/17/$33.00 © 2017 IEEE

EDITOR **JEFFREY VOAS**
NIST; j.voas@ieee.org

In September 2016, the website of computer security consultant Brian Krebs was hit with 620 Gbps of traffic, "many orders of magnitude more traffic than is typically needed to knock most sites offline."[2] At about the same time, an even bigger DDoS attack using Mirai malware—peaking at 1.1 Tbps—targeted the French webhost and cloud service provider OVH.[3]

In the wake of the public release of Mirai's source code by its creator soon afterward,[4] hackers offered Mirai botnets for rent with as many as 400,000 simultaneously connected devices.[5] More Mirai attacks followed, notably one in October 2016 against service provider Dyn that took down hundreds of websites—including Twitter, Netflix, Reddit, and GitHub—for several hours.[6]

Mirai primarily spreads by first infecting devices such as webcams, DVRs, and routers that run some version of BusyBox (busybox.net). It then deduces the administrative credentials of other IoT devices by means of brute force, relying on a small dictionary of potential username–password pairs.

Today, Mirai mutations are generated daily, and the fact that they can continue to proliferate and inflict real damage using the same intrusion methods as the original malware is indicative of IoT device vendors' chronic neglect in applying even basic security practices.

Surprisingly, IoT botnets have received only sporadic attention from researchers.[7,8] If the security community doesn't respond more quickly and devise novel defenses, however, ever-more sophisticated attacks will become the norm and might disrupt the Internet infrastructure itself.

MIRAI THROUGH THE LOOKING GLASS

Mirai causes a DDoS against a set of target servers by constantly propagating to weakly configured IoT devices.

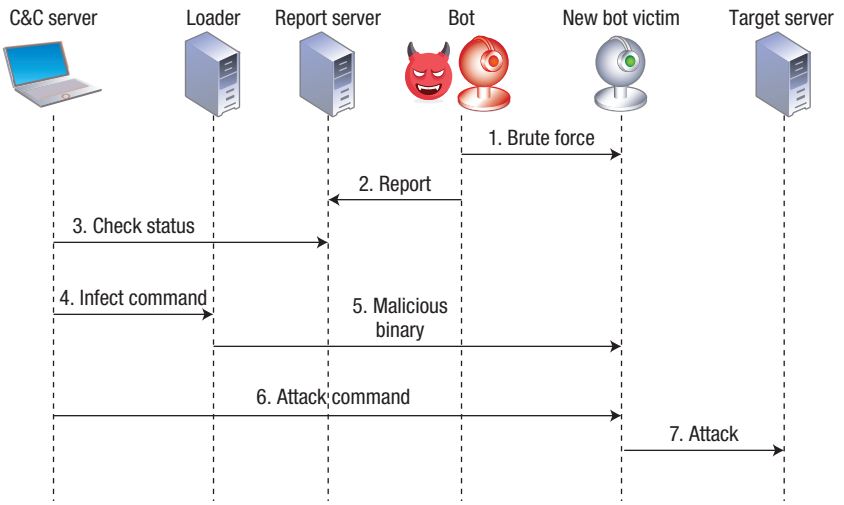

Figure 1. Mirai botnet operation and communication. Mirai causes a distributed denial of service (DDoS) to a set of target servers by constantly propagating to weakly configured Internet of Things (IoT) devices.

Main components

A Mirai botnet is comprised of four major components. The *bot* is the malware that infects devices. Its twofold aim is to propagate the infection to misconfigured devices and to attack a target server as soon as it receives the corresponding command from the person controlling the bot, or botmaster. The *command and control (C&C) server* provides the botmaster with a centralized management interface to check the botnet's condition and orchestrate new DDoS attacks. Typically, communication with other parts of the infrastructure is conducted via the anonymous Tor network. The *loader* facilitates the dissemination of executables targeting different platforms (18 in total, including ARM, MIPS, and x86) by directly communicating with new victims. The *report server* maintains a database with details about all devices in the botnet. Newly infected ones typically directly communicate with it.

Botnet operation and communication

Initially, Mirai scans random public IP addresses through TCP ports 23 or 2323. Some addresses including those of the US Postal Service, the Department of Defense, the Internet Assigned Numbers Authority, General Electric, and Hewlett-Packard are excluded, probably to avoid attracting government attention.[9] Figure 1 shows the key steps in botnet operation and communication.

Step 1. The bot engages in a brute-force attack to discover the default credentials of weakly configured IoT devices. There are 62 possible username–password pairs hardcoded in Mirai.

Step 2. Upon discovering the correct credentials and gaining a shell (a command-line or graphical user interface), the bot forwards various device characteristics to the report server through a different port.

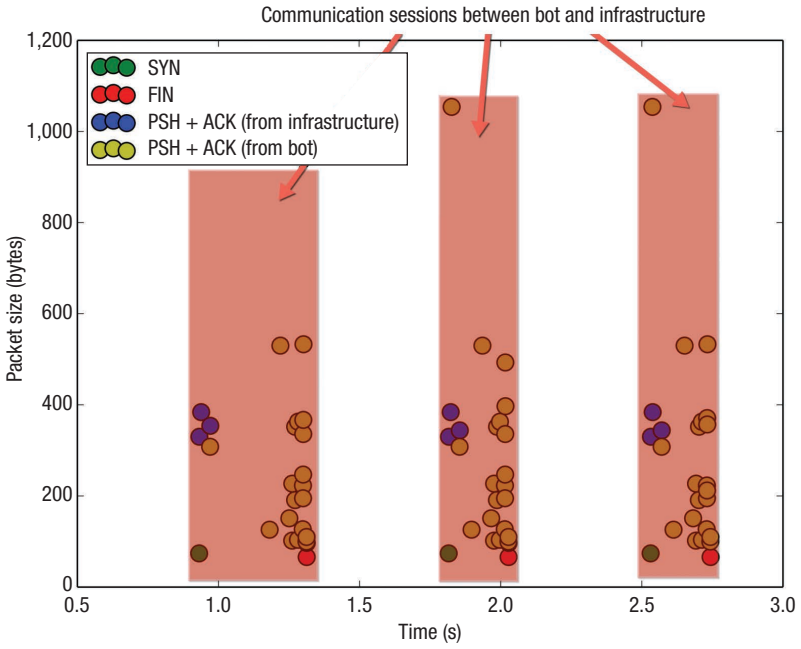

Figure 2. Distinctive communication patterns between an infected IoT device and Mirai's loader component. SYN (synchronize), FIN (finish), PSH (push), and ACK (acknowledge) are standard TCP packet types.

Step 3. Via the C&C server, the botmaster frequently checks new prospective target victims as well as the botnet's current status by communicating with the report server, typically through Tor.

Step 4. After deciding which vulnerable devices to infect, the botmaster issues an infect command in the loader containing all necessary details—for example, IP address and hardware architecture.

Step 5. The loader logs into the target device and instructs it to download and execute the corresponding binary version of the malware, typically via GNU Wget (www.gnu.org/software/wget/manual/wget.html) or the Trivial File Transport Protocol. Interestingly, as soon as the malware is executed it will attempt to protect itself from other malware by shutting down points of intrusion such as Telnet and Secure Shell (SSH) services. At this point, the newly recruited bot instance can communicate with the C&C server to receive attack commands. It does so by resolving a domain name hardcoded in the executable (by default, the value of this entry is cnc.changeme.com in Mirai's source code) rather than a static IP address. Thus, the botmaster has the luxury of changing his IP address over time without modifying the binary and without extra communication.

Step 6. The botmaster instructs all bot instances to commence an attack against a target server by issuing a simple command through the C&C server with the corresponding parameters such as the type and duration of attack and the IP addresses of the bot instances and target server.

Step 7. The bot instances will start attacking the target server with one of 10 available attack variations such as Generic Routing Encapsulation (GRE), TCP, and HTTP flooding attacks.

Mirai signatures

Compared to other similar malware,[10] Mirai doesn't try to avoid detection. Almost all stages of infection leave a footprint that can be recognized through basic network analysis. Mirai signatures include

- sequentially testing specific credentials in specific ports,
- sending reports that generate distinctive patterns,
- downloading a specific type of binary code,
- exchanging keep-alive messages,
- receiving attack commands that have a specific structure, and
- generating attack traffic with very few random elements.

Figure 2 shows some standard communication patterns between an IoT device that's already infected but not actively launching any kind of attack and Mirai's loader component. Although the communication session times vary, the type of messages, their packet sizes, and the sequence of messages form a characteristic pattern indicative of the malware's infection.

MIRAI VARIANTS

One would have expected the public release of Mirai's source code, coupled with its relatively noisy network presence, to quickly lead to effective detection and defense mechanisms. However, the opposite occurred: within only two months of the source code's release, the number of bot instances more than doubled, from 213,000 to 493,000, and a wide range of Mirai variants emerged.[11] Even today—nearly a year after Mirai's appearance—bots continue to exploit the same weak security configurations in the same types of IoT devices.

Although most Mirai infections occur through TCP ports 23 and 2323, Mirai strains identified in November 2016 rely on other TCP ports to commandeer devices—for example, port 7547, which ISPs use to remotely manage customers' broadband routers. That same month, one such Mirai variant knocked nearly a million Deutsche Telekom subscribers offline.[12]

In February 2017, a Mirai variant launched a 54-hour-long DDoS attack against a US college.[13] The following month, yet another novel variant appeared with bitcoin miner functionality, although it's doubtful that compromising IoT devices would yield significant revenue.[14]

Active since April 2017, Persirai[15] is another IoT botnet that shares Mirai's code base. Discovered by Trend Micro researchers and named for its likely Iranian origin (the name is a portmanteau of Persian and Mirai), it attempts to access the interface of specific vendors' webcams through TCP port 81. If successful, it then worms its way into the client's router through a universal plug and play (UPnP) vulnerability, downloads the malicious binaries, and, after execution, deletes them. Rather than deducing webcam credentials via a brute-force attack, the malware proliferates by exploiting a documented zero-day flaw that lets attackers directly obtain the password file. The DDoS attack armory includes User Datagram Protocol flooding attacks. An estimated 120,000 devices in the wild are vulnerable to Persirai.

OTHER IOT BOTNETS

Following Mirai's example, other IoT botnets have recently emerged. While relying on the same basic principles, the authors of this malware are exploring increasingly sophisticated mechanisms to make their botnets more powerful than the competition as well as to obfuscate their activity.

The first IoT botnet written in the Lua programming language was reported by MalwareMustDie in late August 2016.[16] Most of its army is composed of cable modems with ARM CPUs and using Linux. This malware incorporates sophisticated features such as an encrypted C&C communication channel and customized iptables rules to protect infected devices.

The Hajime botnet, discovered in October 2016 by Rapidity Networks,[17] uses a method of infection similar to that of Mirai. However, rather than having a centralized architecture, Hajime relies on fully distributed communications and makes use of the BitTorrent DHT (distributed hash tag) protocol for peer discovery and the uTorrent Transport Protocol for data exchange. Every message is RC4 encrypted and signed using public and private keys. So far, Hajime hasn't evidenced malicious behavior; in fact, it actually closes potential sources of vulnerabilities in IoT devices that Mirai-like botnets exploit, causing some researchers to speculate that it was created by a whitehat.[18] But its true purpose remains a mystery.

A BusyBox-based IoT botnet like Mirai, BrickerBot was unearthed by Radware researchers in April 2017.[19] By leveraging SSH service default credentials, misconfigurations, or known vulnerabilities, this malware attempts a permanent denial-of-service (PDoS) attack against IoT devices using various methods that include defacing a device's firmware, erasing all files from its memory, and reconfiguring network parameters.

LESSONS LEARNED

The dramatic impact of DDoS attacks by Mirai, its variants, and other similar botnets highlight the risks IoT devices pose to the Internet. Currently, even naive approaches can gain control of such devices and create a massive and highly disruptive army of zombie devices. The ease of infection and stability of the generated bot population are alluring factors for any attacker.

There are five main reasons IoT devices are particularly advantageous for creating botnets:

- *Constant and unobtrusive operation*. Unlike laptop and desktop computers, which have frequent on–off cycles, many IoT devices such as webcams and wireless routers operate 24/7 and in many cases aren't properly recognized as computing devices.
- *Feeble protection*. In their rush to penetrate the IoT market, many device vendors neglect security in favor of user-friendliness and usability.
- *Poor maintenance*. Most IoT devices fall under the setup-and-forget umbrella—after initially setting them up, users and network administrators forget about them unless they stop working properly.
- *Considerable attack traffic*. Contrary to common belief, IoT devices are powerful enough and well situated to produce DDoS attack traffic comparable to that of modern desktop systems.
- *Noninteractive or minimally interactive user interfaces*. Because IoT devices tend to require minimum user intervention, infections are more likely to go unnoticed. Even when they're noticed, there's no easy way for the user to address them short of replacing the device.

Two years ago we correctly predicted the emergence of IoT-powered DDoS attacks,[20] and today increasingly sophisticated Mirai variants and imitators are appearing at an alarming rate. This malware typically runs on multiple platforms and is usually lightweight enough to execute in a tiny amount of RAM. In addition, the infection process is relatively simple, making every vulnerable device a bot candidate even with frequent rebooting. Although most existing IoT malware is easy to profile and detect, newer bots are stealthier.

Much of the responsibility for DDoS attacks often lies with users who practice poor security behaviors and system administrators who fail to deploy adequate safeguards. In the case of IoT botnets, however, it's device vendors who should assume the responsibility for naively distributing products with weak security, including default credentials and remote access capabilities. IoT vendors are also in a unique position to provide the automated

security updates that would address the problem. Solutions that require manual intervention—for example, frequently changing passwords—are unrealistic in the IoT realm, where many devices must be self-regulating. What we need now is the technical means to enforce security best practices in computer networks as well as robust security standards for IoT devices and distributors.

REFERENCES

1. "MMD-0055-2016-Linux/PnScan; ELF Worm That Still Circles Around," blog, MalwareMustDie, 24 Aug. 2016; blog.malwaremustdie.org/2016/08/mmd-0054-2016-pnscan-elf-worm-that.html.
2. "KrebsOnSecurity Hit with Record DDoS," blog, KrebsOnSecurity, 16 Sept. 2016; krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos.
3. D. Goodin, "Record-Breaking DDoS Reportedly Delivered by >145K Hacked Cameras," *Ars Technica*, 28 Sept. 2016; arstechnica.com/security/2016/09/botnet-of-145k-cameras-reportedly-deliver-internets-biggest-ddos-ever.
4. J. Gamblin, "Mirai-Source-Code," GitHub; github.com/jgamblin/Mirai-Source-Code/blob/master/ForumPost.txt.
5. C. Cimpanu, "You Can Now Rent a Mirai Botnet of 400,000 Bots," BleepingComputer.com, 24 Nov. 2016; www.bleepingcomputer.com/news/security/you-can-now-rent-a-mirai-botnet-of-400-000-bots.
6. C. Williams, "Today the Web Was Broken by Countless Hacked Devices—Your 60-Second Summary," *The Register*, 21 Oct. 2016; www.theregister.co.uk/2016/10/21/dyn_dns_ddos_explained.
7. E. Bertino and N. Islam, "Botnets and Internet of Things Security," *Computer*, vol. 50, no. 2, 2017, pp. 76–79.
8. K. Angrishi, "Turning Internet of Things (IoT) into Internet of Vulnerabilities (IoV): IoT Botnets," arXiv preprint, 13 Feb. 2017, arXiv:1702.03681.
9. B. Herzberg, D. Bekerman, and I. Zeifman, "Breaking Down Mirai: An IoT DDoS Botnet Analysis," blog, Imperva Incapsula, 26 Oct. 2016; www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html.
10. S.S.C. Silva et al., "Botnets: A Survey," *Computer Networks*, vol. 57, no. 2, 2013, pp. 378–403.
11. *Distributed Denial of Service (DDoS) Threat Report: Q4 2016*, threat report 20170222-EN-A4, Nexusguard, 2017; news.nexusguard.com/threat-advisories/q4-2016-ddos-threat-report.
12. "New Mirai Worm Knocks 900K Germans Offline," blog, KrebsOnScurity, 16 Nov. 2016; krebsonsecurity.com/2016/11/new-mirai-worm-knocks-900k-germans-offline.
13. D. Bekerman, "New Mirai Variant Launches 54 Hour DDoS Attack against US College," blog, Imperva Incapsula, 29 Mar. 2017; www.incapsula.com/blog/new-mirai-variant-ddos-us-college.html.
14. D. McMillen and M. Alvarez, "Mirai IoT Botnet: Mining for Bitcoins?," *Security Intelligence*, 10 Apr. 2017; securityintelligence.com/mirai-iot-botnet-mining-for-bitcoins.
15. T. Yeh, D. Chiu, and K. Lu, "Persirai: New Internet of Things (IoT) Botnet Targets IP Cameras," blog, TrendLabs, 9 May 2017; blog.trendmicro.com/trendlabs-security-intelligence/persirai-new-internet-things-iot-botnet-targets-ip-cameras.
16. "MMD-0057-2016-Linus/LuaBot-IoT Botnet as Service," blog, MalwareMustDie, 6 Sept. 2016; blog.malwaremustdie.org/2016/09/mmd-0057-2016-new-elf-botnet-linuxluabot.html.
17. S. Edwards and I. Profetis, "Hajime: Analysis of a Decentralized Internet Worm for IoT Devices," Rapidity Networks; 16 Oct. 2016; security.rapiditynetworks.com/publications/2016-10-16/hajime.pdf.
18. P. Muncaster, "Mirai-Busting Hajime Worm Could Be Work of White Hat," *Infosecurity Mag.*, 20 Apr. 2017; www.infosecurity-magazine.com/news/mirai-busting-hajime-worm-could.
19. "'BrickerBot' Results in PDoS Attack," Radware, 5 Apr. 2017; security.radware.com/ddos-threats-attacks/brickerbot-pdos-permanent-denial-of-service.
20. C. Kolias, A. Stavrou, and J. Voas, "Securely Making 'Things' Right," *Computer*, vol. 48, no. 9, 2015, pp. 84–88.

CONSTANTINOS KOLIAS is a research assistant professor in the Department of Computer Science at George Mason University as well as lead engineer for the first IoT laboratory at NIST. Contact him at kkolias@gmu.edu.

GEORGIOS KAMBOURAKIS is an associate professor in the Department of Information and Communication Systems Security and director of the Laboratory of Information and Communication Systems Security (Info Sec Lab) at the University of the Aegean. Contact him at gkamb@aegean.gr.

ANGELOS STAVROU is a professor in the Department of Computer Science and director of the Center for Assurance Research and Engineering (CARE) at George Mason University. Contact him at astavrou@gmu.edu.

JEFFREY VOAS is a Fellow of IEEE as well as of the American Association for the Advancement of Science (AAAS) and the Institution of Engineering and Technology (IET). Contact him at j.voas@ieee.org.