CSI4139
Assignment 2
Name: NamChi Nguyen
Student #: 7236760

Mirai Botnet DDoS Attack

Botnet description

A bot is a program on a machine that has been infected by malware and can infect other machines similar to a zombie that spreads the virus to other humans, hence can also be referred to as a zombie. However, the bot is only activated after receiving a command from the command-and-control (C2) server network which is managed remotely by the attacker. So, a botnet is a network of bots with a size that can range from hundreds to thousands of bots that act in an organized fashion under the control of the attacker.

In September 2016, the Mirai botnet launched an enormous distributed denial of service (DDoS) attack. A DDoS attack floods the target (eg. a website or a server) with an unusually high volume of network traffic from multiple devices that will disrupt its service and make it unavailable to users. Mirai accomplished this by infecting and gaining control of approximately 200,000 to 300,000 Internet of Things (IoT) devices initially before reaching as high as 600,000 IoT devices by November 2016 and then dropping to 100,000 devices in February 2017. IoT devices are devices that have access to the Internet but are not strictly computers. Some of these targeted devices included IP cameras, routers, printers and digital video recorders (DVRs). The botnet rendered high profile targets out of service such as the security blog Krebs on Security, domain name system (DNS) provider Dyn and a Liberian telecom called Lonestar Cell as well as game servers, and many other sites [1].

The Mirai botnet derived from a DDoS malware family called BASHLITE in which this family of malware infected Linux systems utilizing a brute force attack using default login credentials. A brute force login attack tries every possible combination of a username and password. It had four major operations during its proliferation which included 1) rapid scanning, 2) hardcoded report server, 3) loader program and 4) download/execution of malware.

Rapid scanning sends transmission control protocol synchronize (TCP SYN) probes asynchronously to IPv4 addresses on Telnet TCP ports (TCP/23 and TCP/2323) which omits hard-coded IP addresses on the blacklist. Once a prospective target is found, then a brute force login of a username/password pair occurs to create a Telnet connection using the set of 62 credentials it already owned from BASHLITE's set. When the login is successful, the target's IP and credentials are relayed to a hardcoded report server which then dispatches a loader program that asynchronously is activated to infect the IoT device. Afterwards, the attacker can send commands to the C2 server which will govern the bots and collectively launch the DDoS attack [1].

Furthermore, the geographic location of infected IoT devices were prevalent in Brazil, Colombia, and Vietnam which accumulated to approximately 41.5% of bots worldwide. Along with these top three countries and several more in South America and Southeast Asia, the reason for this large concentration of bots in these areas were due to devices that had reduced computational power and low bandwidth [1].

Main features

The main features of the Mirai botnet included the ability to add new login credentials (username/passwords) to its dictionaries from the original 62 username/password pairs it had from BASHLITE's collection, scan new ports and close infected ports, alter the source code for blacklists that omitted Department of Defense (DoD) attacks and terminate competing malware [1].

Evolution of Mirai

The DDoS attack lasted from September 2016 to February 2017 and during this timeframe, Mirai evolved considerably. Initially, Mirai utilized an IP-based command-and-control (C2) infrastructure but elevated to a domain-based C2 in mid-September in which the malware could now delete its executable binary and disguise its process ID. Later on, in November 2016, packed binaries made an appearance, and the botnet could scan for the ports TCP/7547 and TCP/5555 which were related to CWMP and it was capable of utilizing domain generation algorithms (DGA) in a hardcoded C2 domain [1].

Vulnerabilities (in hardware, software and humans)

The successful DDoS attack were due to factors within hardware, software and by humans.

For hardware, during the designing of the IoT devices, there is the possibility that the hardware didn't apply the Principle of Least Privilege because once the bot was successfully logged in, it could download the malicious code and execute it. If the bot had minimal permissions, it would have required administrator privileges to be able to run the malicious commands sent from the attacker on the device [1].

For software, there was the lack of automatic updates in the IoT devices to patch any known vulnerabilities and repair bugs. When the IoT device rebooted, it was no longer infected, however since auto updates weren't available to mitigate the threat, the device could easily be infected again since the credentials weren't changed. Also, the botnet was capable to scanning a vast amount of open ports and close infected ports [1].

For humans, the usage of weak passwords in IoT devices were exploited and was the main vulnerability that led to Mirai's success. These passwords weren't changed by the user from the default configuration provided by the manufacturer or the criteria of creating a strong password wasn't required in which users could create insecure passwords that could be easily guessed. Some sample passwords included 123456, pass, admin, etc. which demonstrates that the password didn't have to be an assortment of uppercase/lowercase letters, numbers and special characters and no minimum password length was specified. Also, the manufacturers played a role in which there was an inadequate amount of security practices that were considered or reinforced while constructing IoT devices (such as the password policy). Some manufacturers that produced infected IoT devices included Dahua, Huawei, ZTE and many more [1].

Protection against botnets

Some steps listed by the authors to protect against future botnet attacks are as follows (from most important to least which is the same ordering in the paper):

1. *Security hardening*:
The authors mentioned utilizing default-closed ports instead of default-open ports, and during the designing of IoT devices, to apply address space layout randomization (ASLR), isolation boundaries, and the principle of least privilege [1].

The usage of default-open ports can permit attackers to configure services to spread content, exploit old versions of software that are no longer in use and acquire information about the network [2]. ASLR "is a memory-protection process for operating systems that guards against buffer-overflow attacks by randomizing the location where system executables are loaded into memory" [3] in which zero-day attacks can occur when the attacker correctly guesses the position and processes in memory [3].

This is the most crucial step to take because there are collectively many tasks to consider in security hardening that will attempt to maximize the protection of the device and reduce the possibilities of an attack being successful.

2. *Automatic updates*:
The software architecture must be able to securely overwrite and rollback when a failure occurs along with cryptographic resources and PKI infrastructure. This step is especially important in order for the devices to patch up vulnerabilities and repair bugs that attackers can exploit. An example mentioned in the paper was that Deutsche Telekom's routers had been infected and could infect other devices with a vulnerability found in its update mechanism. However, the automatic update patched the device and mitigated the threat [1].

3. *Notifications*:
Notifications are a mechanism to restore the security compliances of devices and indicate to users that something may be wrong with the device. This is also an important step to include, however in IoT devices, the ability to notify the user or admin that the device has been compromised is not trivial. These devices would need a public indication of the owner and a secured channel to communicate with the user. The authors suggested that a mandatory email is set up with the manufacturer or there is a platform in which users can be notified when a device has problems [1].

4. *Facilitating device identification*:
The task to identify the model or firmware of the device is challenging. To try and solve this challenge, the authors proposed that the manufacturers consider identifying the model or firmware version in a standard way in the network such as encrypting it in part of the device's MAC address. This would make information visible at the network access layer for network operators. This step could help to mitigate the disclosure of important information found on the network [1].

5. *Defragmentation*:
Fragmentation stores data or processes non-contiguously in memory which can slow down performance and cause storage to be used inefficiently. So, defragmentation would find and combine the non-contiguous fragments of the data in memory to increase the efficiency of the storage [5]. This could be accomplished by implementing operating systems such as Android

Thing, Windows for IoT and several others on the IoT devices [1]. This step would increase the effectiveness of the storage and provide more storage to install an IDS or firewall for example.

6. *End-of-life*:
When a device reaches its end of life, this will make it vulnerable when the product is no longer supported and kept up-to-date, thus can be exploited by attackers. This step is not as important and is not as easy to reinforce as the consumer would have to be knowledgeable about the device's lifecycle and be aware of when they would have to replace or take the device offline to ensure security [1].

Some additional steps that weren't mentioned include the installation of an intrusion detection system (IDS) or a firewall. An IDS can detect threats or anomalies and alert the user of suspicious activity while a firewall can allow and block incoming and outgoing traffic. Also, using encryption when applicable would aid to keep data that is transmitted secured and this would help to prevent attackers from accessing sensitive data easily [4].

Conclusion
To summarize, the Mirai botnet launched a massive DDoS attack on several sites and companies using infected IoT devices. These devices currently have inadequate protection against attackers in which weak passwords were exploited because they were not changed from the default configuration. For future productions of IoT devices, many security practices should be followed meticulously in order to provide the same amount of security that is bestowed to computers and phones. Security hardening and automatic updates are a couple methods that can augment the protection of an IoT device considerably. As IoT devices become more secure in the future, it would make it more difficult for malicious entities to exploit the vulnerabilities found presently.

References:

[1] M. Antonakakis, *et al*., Understanding the Mirai Botnet, *Proceedings of the 26th Usenix Security Symposium*, Vancouver, Canada, August 16-18, 2017, pp. 1093 – 1110. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis. [Accessed: Nov. 8, 2018]
[2] I. Muscat, "How to Close Unused Open Ports: TCP and UDP Port scan", Jun. 19, 2014, [Online]. Available: https://www.acunetix.com/blog/articles/close-unused-open-ports/. [Accessed: Nov. 11, 2018]
[3] M. Rouse, "Address space layout randomization (ASLR)", Jun. 2014, [Online]. Available: https://searchsecurity.techtarget.com/definition/address-space-layout-randomization-ASLR. [Accessed: Nov. 11, 2018]
[4] Online Trust Alliance, "The Enterprise Iot Security Checklist," 2018, [Online]. Available: https://otalliance.org/system/files/files/initiative/documents/enterprise_iot_checklist.pdf. [Accessed: Nov. 12, 2018]
[5] TechTarget, "Defragmentation", n.d., [Online]. Available: https://searchwindowsserver.techtarget.com/definition/defragmentation. [Accessed: Nov. 15, 2018]