The Research on IoT Botnet

– Use Mirai Botnet as an Example

Fang Lin
Freie Universität Berlin
IoT & Security Seminar Report

***Abstract*—The Internet of Things (IoT) is becoming an indispensable part of our daily lives, playing an increasingly important role in health, the environment, the family, the military and so on. The Internet of Things has grown tremendously in recent years. However, IoT devices still suffer from basic security vulnerabilities. Hackers use their computing and communication advantages to perform different types of attacks, IoT botnet is one of them. Mirai Botnet is the most typical type of Botnets. This article uses Mirai Bot as the main body of analysis. By analysing the development line, structure and propagation form of Mirai Bot, it analyses how Mirai Bot attacks IoT devices, and analyses how to detect and prevent botnets from technical and non-technical aspects, hoping to gain a deeper understanding of IoT botnets and how to prevent it.**

***Index Terms*—IoT, Botnet, Mirai Botnet, DDoS, Deep Learning**

I. Introduction

The introduction part mainly introduces the background and significance of the research, some basic concepts, and the structure of this article.

A. Research Background

The IoT botnet is an important reason that makes IoT devices unable to operate normally and leads to network security issues such as the leakage of private information. IoT botnets mainly interfere with the operation of IoT devices by attacking DDoS. Looking back at the history of the IoT botnets invading the IoT devices, a series of network security incidents have occurred in the economic, smart city and so one. As a classic botnet, Mirai botnet is worthy of our in-depth research on botnets to explore the reasons for these things.

B. Research Significance

On the one hand, through the analysis and research of the Mirai botnet, I can have a deeper understanding of the botnet and also have a better understanding of the security issues of IoT devices. On the other hand, by understanding how to prevent the infringement of botnets, I can also improve my awareness of prevention when using IoT devices, that is, to be able to use IoT devices more sensibly, not to blindly believe and use without them security protection. they, so that the private information can be better protected.

C. Basic Concepts

- **IoT**
  he Internet of things (IoT) describes the network of physical objects—a.k.a. ”things”—that are embedded with sensors, software, and other technologies for the purpose of connecting and exchanging data with other devices and systems over the Internet. [16]
- **Botnet**
  A botnet is a network of infected machines or bots, also called zombies, that has a command-and-control infrastructure and is used for various malicious activities such as distributed denial-of-service (DDoS) attacks. [10]
- **Mirai Botnet**
  The Mirai botnet, composed primarily of embedded and IoT devices, took the Internet by storm in late 2016 when it overwhelmed several high-profile targets with massive distributed denial-of-service (DDoS) attacks. [1]
- **DDoS**
  In a distributed denial-of-service attack (DDoS attack), the incoming traffic flooding the victim originates from many different sources. This effectively makes it impossible to stop the attack simply by blocking a single source. [14]
- **Deep Learning**
  Deep learning (also known as deep structured learning) is part of a broader family of machine learning methods based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised. [15]

D. Structure

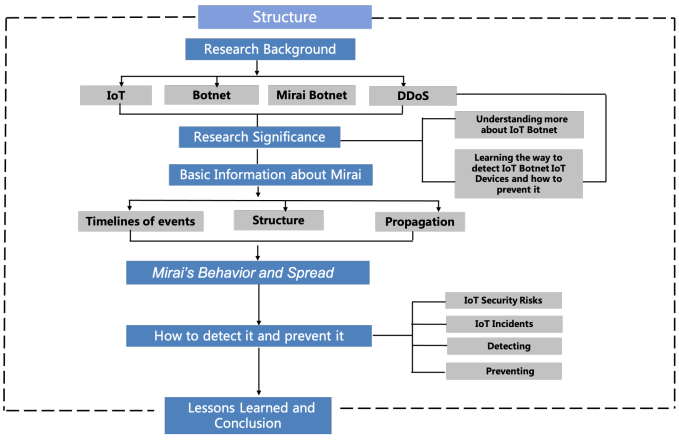

Fig. 1. Structure of this article

II. Basics of IoT Botnets and Mirai Botnet

In this section, the basics of IoT botnets and mirai botnet will be introduced. I introduce firstly the basics of IoT Botnets, and then I will mainly write the information of Mirai Botnet, including the timeline of events, the structure and the way of propagation. At the end I will introduce some other botnets and make a small conclusion about the relationships between them.

A. *Basics of an IoT Botnet*

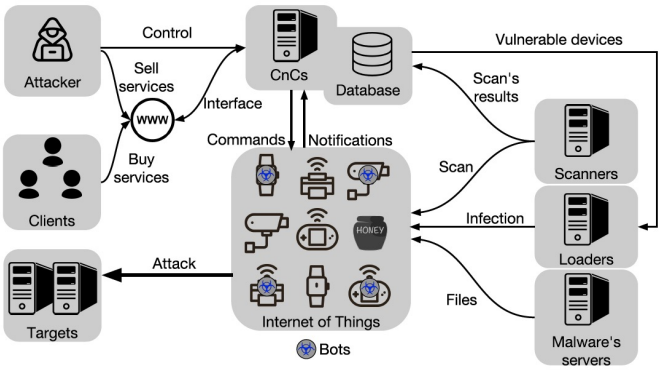

Fig. 2. Overview of an IoT Botnet [3]

- The left part includes attackers, clients, and targets which are easy to understand.
  - Attackers
    Attackers are those who try to attack the IoT devices to gain profits.
  - Clients
    Clients are the groups who bought IoT devices and use them as a part of normal life, in the context they are also the victims.
  - Targets
    Targets are the IoT devices which suffered from the attacks from the attackers and then the information of the clients may be exposed.
- The center part includes three domains, which are CnCs, Database and Bots, these should be explained more detailed in the following.
  - CnCs
    Command and control servers (C&C) are the operators' interface to the botnet. C&Cs receive commands from operators and maintain connections with infected devices to broadcast commands. [3]
  - Database
    Database (potentially distributed) stores information collected by the botnet, e.g., active bots and scan results. [3]
  - Bots
    Bots are infected devices that are part of the botnet. Bots report their state to C&Cs and execute the received commands. [3]
- In the right part there are scanners, loaders and malware's servers.
  - Scanners
    Scanners probe devices to find telnet and SSH servers to attempt login and identify vulnerable devices. [3]
  - Loaders
    Loaders login to vulnerable devices to download and run the botnet malware, creating a new bot. [3]
  - Malware's servers
    Malware servers host resources used by the botnet such as shell scripts and executable binaries. [3]
- How IoT devices will be infected?
  - The scanner first identifies vulnerable devices and reports to the central database.
  - The loader then connects to the vulnerable device to download and run the malware. During the infection process, the loader accesses the server to download and run the malware binary file on the vulnerable device.
  - Once infected, the bot will connect to the C&C of the botnet and wait for commands. To prevent subsequent infection attempts from other botnets, the IoT botnet disables the telnet and SSH services of the infected device.
  - Finally, operators may sell botnet services (for example, denial of service attacks), which are usually accessible through the client's web interface

B. *The Timeline of Mirai Botnet*

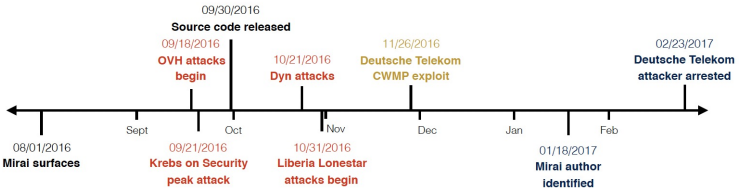

Fig. 3. Timeline of events: Major attacks (red), exploits (yellow), and events (black) related to the Mirai botnet. [1]

From Figure 3 [1], we can see that on August 1, 2016, Mirai's surfaces were released. Mirai's first attack was on September 18, 2016, which was OVH attacks. On the 21st, it attacked DNS provider Dyn. During the two attacks, the source code was released on the Internet. At the end of November, it exploited the vulnerability of Deutsche Telekom CWMP. On January 18, 2017, the author of Mirai was confirmed and arrested at the end of February of the same year.

C. *The Structure and Propagation of Mirai Botnet*

This chapter introduces the structure and propagation of Mirai Botnet mainly by figure 4 [1], which basically follows the structure of an Iot Botnet, but will be more specific and orientational just about Mirai Botnet.

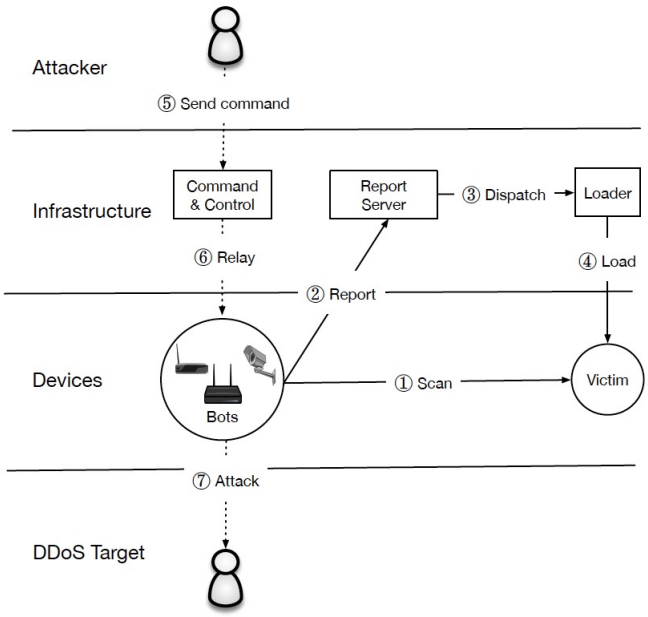

Fig. 4. **Mirai Operation**—Mirai bots scan the IPv4 address space for devices that run telnet or SSH, and attempt to log in using a hardcoded dictionary of IoT credentials. Once successful, the bot sends the victim IP address and associated credentials to a report server, which asynchronously triggers a loader to infect the device. Infected hosts scan for additional victims and accept DDoS commands from a command and control (C2) server. [1]

*a) **The Structure of Mirai Botnet**:* Compare to figure 2, the components of the structure of Mirai Botnet are almost the same, except there is a report server of mirai botnet structure. **Report Server** is a server which Bots will report the victim IP address to it. In the figure 4 [1], four layers are designed to describe the mirai botnet structure.

- **Attacker**
  The same as in figure 2, it's the hackers who want to attack the IoT devices.
- **Infrastructure**
  In Infrastructure Layer, there are three components, C&C, Roport Server and Loader.
- **Devices**
  The Devices layer is made of bots and victim, the victim here is the victim IoT device.
- **DDoS Targer**
  Easy to understand, it's the clients of an IoT devices, which their devices are attacked by Mirai Botnet.

*b) **The Propagation of Mirai Botnet**:* As we can see in figure 4, there are 7 phases about the propagation of Mirai Botnet.

- **Phase 1**
  Mirai spread by first entering a rapid scanning phase 1 where it asynchronously and "statelessly" sent TCP SYN probes to pseudorandom IPv4 addresses, excluding those in a hard-coded IP blacklist, on Telnet TCP ports 23 and 2323 (hereafter denoted TCP/23 and TCP/2323). If Mirai identifies a potential victim, it entered into a brute-force login phase in which it attempted to establish a Telnet connection using 10 username and password pairs selected randomly from a pre-configured list of 62 credentials. [1]
- **Phase 2**
  At the first successful login, Mirai sent the victim IP and associated credentials to a hardcoded report server.
- **Phase 3**
  A separate loader program asynchronously infected these vulnerable devices by logging in, determining the underlying system environment.
- **Phase 4**
  And at the end this separate loader program infected the devices by downloading and executing architecture-specific malware.
- **Phase 5**
  The attackers will send the command to C&C services.
- **Phase 6**
  The C&C services relay the instruction from the attacker to certain bots.
- **Phase 7**
  At the end, the bots will attack the DDoS Target.

- After a successful infection, Mirai attempted to conceal its presence by deleting the downloaded binary and obfuscating its process name in a pseudorandom alphanumeric string. As a consequence, Mirai infections did not persist across system reboots. In order to fortify itself, the malware additionally killed other processes bound to TCP/22 or TCP/23, as well as processes associated with competing infections. At this point, the bot listened for attack commands from the command and control server (C2) while simultaneously scanning for new victims. [1]

D. Other Botnets

In this subsection three other botnets will be shortly introduced.

- **The first IoT botnet written in the Lua programming language was reported by MalwareMustDie in late August 2016.** Most of its army is composed of cable modems with ARM CPUs and using Linux. This malware incorporates sophisticated features such as an encrypted CC communication channel and customized iptables rules to protect infected devices. [8]
- **The Hajime botnet, discovered in October 2016 by Rapidity Networks, uses a method of infection similar to that of Mirai.** However, rather than having a centralized architecture, Hijame 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.But its true purpose remains a mystery.

- **A BusyBox-based IoT botnet like Mirai, BrickerBot was unearthed by Radware researchers in April 2017.** 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.
- **Linux/IRCTelnet is a new IRC botnet ELF malware aimed at IoT devices with IPv6 capabilities.** IRCTelnet combining the concept of Tsunami for IRC protocol, BASHLITE for the infection techniques (telnet brute force access and code injection) and using theMirai botnet's IoT credential list. The base source code of LightAidra/Aidra is used to build the new botnet malware. The botnet is using UDP, TCP flood along with other series of attack methods in both IPv4 and IPv6 protocol. The new malware features the extra IP spoof option in both IPv4 or IPv6. [2]

III. Watching Mirai's Behavior and Tracking Mirai's Spread

In Chapter II I write the basic structure about Mirai Botnet and I describe the propagation of the Mairi Botnet, it's actually the basic route how Mairi attacks IoT devices. In this Chapter I will go further about Mirai Botnet by summarising the analysis from article [1] in two aspects, one is by using the network vantage points: a large, passive network telescope, Internet-wide scanning, active Telnet honeypots, logs of C2 attack commands, passive DNS traffic, and logs from DDoS attack targets to get the information about how Mirai Botnet behavior, another aspect is by tracking Mirai's Spread to know further how it works, and may contribute the our next chapter.

A. *Mirai's Behavior*

*a) **Network Telescope**:* Mirai's indiscriminate, rapid scanning strategy lends itself to tracking the botnet's propagation to new hosts. The authors [1] monitored all network requests to a network telescope composed of 4.7 million IP address operated by Merit Network over a seven month period from July 18, 2016 to February 28, 2017.

On average, the network telescope received 1.1 million packets from 269,000 IP addresses per minute during this period. To distinguish Mirai traffic from background radiation and other scanning activity, we uniquely fingerprinted Mirai probes based on an artifact of Mirai's stateless scanning whereby every probe has a TCP sequence number—normally a random 32-bit integer—equal to the destination IP address. The likelihood of this occurring incidentally is 1=232, and we would expect to see roughly 86 packets demonstrating this pattern in our entire dataset. In stark contrast, we observed 116.2 billion Mirai probes from 55.4 million IP addresses. Prior to the emergence of Mirai, we observed only three IPs that perform scans with this fingerprint. Two of the IP addresses generated five packets; two on TCP/80 and three on TCP/1002. The third IP address belongs to Team Cymru, who conducts regular TCP/443 scans.

*b) **Active Scanning**:* In order to determine the manufacturer and model of devices infected with Mirai, the authors [1] leveraged Censys, which actively scans the IPv4 space and aggregates application layer data about hosts on the Internet.

A number of challenges make accurate device labeling difficult. First, Mirai immediately disables common outward facing services (e.g., HTTP) upon infection, which prevents infected devices from being scanned. Second, Censys scans often take more than 24 hours to complete, during which devices may churn to new IP addresses. Finally, Censys executes scans for different protocols on different days, making it difficult to increase label specificity by combining banners from multiple services.

A conclusion is that devices with open services that are not closed by Mirai (e.g., HTTPS and FTP) can appear repeatedly in Censys banner scans during our measurement window (due to churn) and thus lead to over counting when compared across protocols.

*c) **Telnet Honeypots**:* In order to track the evolution of Mirai's capabilities, the authors [1] collected binaries installed on a set of Telnet honeypots that masqueraded as vulnerable IoT devices. At the end they extracted the set of logins and passwords, IP blacklists, and C2 domains from these binaries, identifying 67 C2 domains and 48 distinct username/password dictionaries (containing a total 371 unique passwords).

*d) **Passive & Active DNS**:* Following the public release of Mirai's source code, competing Mirai botnet variants came into operation. the authors [1] disambiguated ownership and estimate the relative size of each Mirai strain by exploring passive and active DNS data for the 67 C2 domains that we found by reverse engineering Mirai binaries. We also leveraged our DNS data to map the IP addresses present in attack commands to victim domain names. In total, 33 unique DNS clusters of Mirai Botnet are identified.

*e) **Attack Commands**:* To track the DDoS attack commands issued by Mirai operators, Akamai ran a "milker" from September 27, 2016–February 28, 2017 that connected to the C2 servers found in the binaries uploaded to their honeypots.

The authors [1]note that a naive analysis of attack commands overestimates the volume of attacks and targets: individual C2 servers often repeat the same attack command in rapid succession, and multiple distinct C2 servers frequently issued the same command.

*f) **DDoS Attack Traces**:* For Google Shield, The authors [1] shared a list of IP addresses observed by our network telescope and in turn received aggregate statistics on what fraction matched any of 158.8K IP addresses involved in a 1-minute Mirai HTTP-flood attack on September 25, 2016. Finally, Dyn provided them with a set of 107.5K IP addresses associated with a Mirai attack on October 21, 2016.

B. *Tracking Mirai's Spread*

a) ***Bootstrapping****:* Mirai's comparatively modest initial growth may be due to the low bandwidth and computational resources of infected devices, a consequence of the low-accuracy, brute-force login using a small number of credentials, or simply attributable to a bottleneck in loader infrastructure.

b) ***Steady State Size****:* The authors [1] observed multiple phases in Mirai's life: an initial steady state of 200,000–300,000 infections in September 2016; a peak of 600,000 infections at the end of November 2016; and a collapse to roughly 100,000 infections at the end of our observation window in late February 2017 (Figure 3). Even though hosts were initially compromised via a simple dictionary attack, Mirai was able to infect hundreds of thousands of devices. This is similar in scale to historical botnets such as the prolific Srizbi spam botnet (400,000 bots), which was responsible for more than half of all global botnet spam , and the Carna botnet (420,000 bots), the first botnet of IoT devices compromised using default credentials.

c) ***Global Distribution****:* Where Mirai infections were geographically concentrated?

Country	Mirai Infections	Mirai Prevalence	Telnet Prevalence
Brazil	49,340	15.0%	7.9%
Colombia	45,796	14.0%	1.7%
Vietnam	40,927	12.5%	1.8%
China	21,364	6.5%	22.5%
S. Korea	19,817	6.0%	7.9%
Russia	15,405	4.7%	2.7%
Turkey	13,780	4.2%	1.1%
India	13,357	4.1%	2.9%
Taiwan	11,432	3.5%	2.4%
Argentina	7,164	2.2%	0.2%

Fig. 5. Geographic Distributiont [1]

In figure 5 we can see, most Mirai infections originate from Located in Brazil (15.0%), Colombia (14.0%) and Vietnam (12.5%)

d) ***Device Composition****:* While cursory evidence suggested that Mirai targets IoT devices—Mirai's dictionary of default usernames and passwords included routers, DVRs, and cameras.

After the gathering and analysing of the authors [1], the devices they identified were primarily network-attached storage appliances, home routers, cameras, DVRs, printers, and TV receivers made by dozens of different manufacturers.

The manufacturers responsible for the most infected devices we could identify are: Dahua, Huawei, ZTE, Cisco, ZyXEL, and MikroTik.

The data indicates that some of the world's top manufacturers of consumer electronics lacked sufficient security practices to mitigate threats like Mirai, and these manufacturers will play a key part in ameliorating vulnerability. Unfortunately, as discussed in the previous section, the menagerie of devices spanned both countries and legal jurisdictions, exacerbating the challenge of coordinating technical fixes and promulgating new policy to safeguard consumers in the future.

e) ***Device Bandwidth****:* Mirai was primarily powered by devices with limited computational capacity and/or located in regions with low bandwidth.

IV. How to detect and Prevent Mirai Botnet and other Botnets

A. *IoT Security Risks*

This chapter introduces some basick security risks of IoT so that we know why some attacks are easy to be taken.

B. *Some Incidents of IoT caused by Botnets*

I'm not very sure if I should write some certain examples about some events. Above I already mentioned the timeline of events which are concerned with just Mirai Botnet. However, I'm also supposed to provide some examples which are not that closely to DDoS, I mean Mirai Botnet is obviously a DDoS attacking Botnet. Somehow I think maybe to give some other incidents by not just DDoS attacking attacks maybe help to improve the width of this article. (So maybe if at the end the words number doesn't satisfy the requirement, this part is then needed.)

C. *Detecting*

Here I may use two specific examples by Deep Learning based detection system. The thing now is that I don't how further I should write about this chapter, since to describe the algorithms and the structure of DL is not very important to this article

D. *Preventing*

This Chapter is about some basic techniques that we can use to avoid the attacks from Botnet and also some IoT security strategies which can prevent IoT devices from suffering the attacks from Botnets./

a) ***Protection Techniques*** *:*

- 1
- 2
- 3
- 4
- 5
- 6

b) ***IoT Security Strategy*** *:*

- 1
- 2
- 3
- 4
- 5
- 6

V. Lessoned Learned and Conclusion

A. *Lessoned Learned*

- 1
- 2
- 3
- 4
- 5
- 6

B. *Conclusion*

- 1
- 2
- 3
- 4
- 5
- 6

References

[1] Manos Antonakakis, Georgia Institute of Technology; Tim April, Akamai; Michael Bailey, University of Illinois, Urbana-Champaign; Matt Bernhard, University of Michigan, Ann Arbor; Elie Bursztein, Google; Jaime Cochran, Cloudflare; Zakir Durumeric and J. Alex Halderman, University of Michigan, Ann Arbor; Luca Invernizzi, Google; Michalis Kallitsis, Merit Network, Inc.; Deepak Kumar, University of Illinois, Urbana-Champaign; Chaz Lever, Georgia Institute of Technology; Zane Ma and Joshua Mason, University of Illinois, Urbana-Champaign; Damian Menscher, Google; Chad Seaman, Akamai; Nick Sullivan, Cloudflare; Kurt Thomas, Google; Yi Zhou, University of Illinois, Urbana-Champaign.Understanding the Mirai Botnet. (2017)

[2] Kishore Angrishi. Turning Internet of Things(IoT) into Internet of Vulnerabilities (IoV) : IoT Botnets. (2017)

[3] Artur Marzano, David Alexander, O. Fonseca, E. Fazzion, C. Hoepers, Klaus Steding-Jessen, M. H. P. Chaves, Ítalo S. Cunha, D. Guedes, W. Meira .The Evolution of Bashlite and Mirai IoT Botnets. 2018 IEEE Symposium on Computers and Communications (ISCC)

[4] Basheer Al-Duwairi, Wafaa Al-Kahla, Mhd Ammar AlRefai, Yazid Abdelqader, Abdullah Rawash, Rana Fahmawi. SIEM-based detection and mitigation of IoT-botnet DDoS attacks. (2019)

[5] Huy-Trung Nguyen, Quoc-Dung Ngo, Van-Hoang Le. A novel graph-based approach for IoT botnet detection. (2019)

[6] Stephen Herwig, Katura Harvey, George Hughey, Richard Roberts, Dave Levin. Measurement and Analysis of Hajime, a Peer-to-peer IoT Botnet. (2019)

[7] Xiaolu Zhang, Oren Upton, Nicole Lang Beebe, Kim-Kwang Raymond Choo. IoT Botnet Forensics: A Comprehensive Digital Forensic Case Study on Mirai Botnet Servers. (2020)

[8] Constantinos Kolias, George Mason University, Georgios Kambourakis, University of the Aegean Angelos Stavrou, George Mason University Jeffrey Voas, IEEE Fellow. DDoS in the IoT: Mirai and Other Botnets. (2017)

[9] Priscilla Moriuchi, Sanil Chohan. Mirai-Variant IoT Botnet Used to Target Financial Sector in January 2018. (2018)

[10] Elisa Bertino, Purdue University, Nayeem Islam, Qualcomm. Botnets and Internet of Things Security. (2017)

[11] Saleh Soltan, Prateek Mittal, and H. Vincent Poor, Princeton University. BlackIoT: IoT Botnet of High Wattage Devices Can Disrupt the Power Grid. (2018)

[12] Amaal Al Shorman, Hossam Faris, Ibrahim Aljarah. Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection. (2019)

[13] Vinayakumar R, Mamoun Alazab Senior Member, IEEE, Sriram S, Quoc-Viet Pham, Soman KP, Simran K. A Visualized Botnet Detection System based Deep Learning for the Internet of Things Networks of Smart Cities. (2020)

[14] https://en.wikipedia.org/wiki/Denial-of-service_attack

[15] https://en.wikipedia.org/wiki/Deep_learning

[16] https://en.wikipedia.org/wiki/Internet_of_things