IoT Botnets

Pamela Beltrán-García, Eleazar Aguirre-Anaya(✉),
Ponciano Jorge Escamilla-Ambrosio, and Raúl Acosta-Bermejo

Laboratory of Cybersecurity, Centro de Investigación en Computación,
Instituto Politécnico Nacional, Mexico City, Mexico
pam.belt.g@gmail.com,
{eaguirre,pescamilla}@cic.ipn.mx, racostab@ipn.mx,
http://www.cic.ipn.mx

Abstract. This paper presents a comprehensive state-of-art review that discusses the IoT botnet behaviour, including topology and communication between botmaster and bots, thus is possible to make a comparison of IoT botnets, based on their topology, type of attack, target, kind of propagation and operation. In several investigations, it is explained that a significant problem is an increase in the development of IoT botnets, such as attacks like DDoS. To this aim, understanding the behaviour of the IoT botnets could be helpful to prevent them.

Keywords: IoT · Botnet · Bot

1 Introduction

The concept of Internet of Things was initially expressed as "computers everywhere," formulated by Ken Sakamura at the University of Tokyo in 1984. In 1999, Kevin Ashton was the first to devise the term "Internet of Things" [1]. The phrase "Internet of Things," which is also shortly well-known as IoT composed of two words: first is "Internet" and second is "Things."

The Internet is a global system of interconnected networks that use the standard Internet protocol suite (TCP/IP) to serve billions of users worldwide [3]. Devices with few resources in computation and energy capacity characterize the term "Things" in IoT; these can have sensors, actuators, and processing unit; which have an IP address assigned for Internet connectivity, either wired or wireless networks [2, 4]. Due to the few IoT devices resources, complicates the task of installing security controls and causes them to be vulnerable to be infected and execute DDoS attacks by flooding a service with legitimate requests [27].

With the increase in the development of IoT devices, security has become an essential factor due to attacks that are multiplying; one way is the use of botnets. Silva et al. [24] discuss that approximately 16–25% computers connected to the Internet are members of botnets. Other studies report that at the beginning of the year 2008, e-mail spam was generated only by six botnets [25]. Also, Symantec Internet Security Threat Report indicates that 5.06 million distinct botnet computers; 61,940 active computers per day, and 4,091 bot command servers have been observed [26].

M. F. Mata-Rivera et al. (Eds.): WITCOM 2019, CCIS 1053, pp. 247–257, 2019.
https://doi.org/10.1007/978-3-030-33229-7_21

Actually, according to Spamhaus Malware Labs, 10,263 botnets hosted on 1,121 different networks in 2018 were identified and blocked. That is an 8% increase from the number of botnet C&C in 2017 [32].

2 Botnets Background

The botnet term is a combination of two main words, Bot as a Robot abbreviation and Net for Network; so, it can be defined as a "network of infected hosts called bots, which are controlled by a human operator, better known as the Botmaster."

Botnets recruit vulnerable hosts by using a type of malware that exploits vulnerabilities; an infrastructure is created between the now infected hosts and the Botmaster who takes the control of these hosts remotely using a Command and Control server (C&C), through the sending of commands to perform malicious activities, such as DDoS, spam, or information theft.

In the context of IoT, botnet is a network of compromised IoT devices, such as cameras, modems, DVR, sensors and other devices that use the IP protocol with the characteristic to transmit data over the Internet, infected with malware. Such networks allow an attacker to control the devices performing malicious tasks, as well as propagate their malware [5].

2.1 Botnet Stages

Botnets perform their actions in three main stages [7, 8]:

- Infection: The malware used for recruit new bots can be propagated by exploiting vulnerabilities, downloading by web/mail, installing software from unreliable sources, then executed, and the host became part of the botnet.
- Command and Control: Once infected, the bot communicates with the host that controls the botnet to receive commands.
- Malicious Activities: Execution of attacks such as DDoS, spam, etc.

2.2 Botnet Topologies

According to Vormayr et al. [9] and Dhinnesh et al. [10], the topology of the botnets is defined by the command and control process they execute:

- Centralized: The Botmaster controls and monitors all bots from a single central point, which makes the latency low, that is, all bots receive commands and reports to the center point (C&C server). Likewise, there are two centralized topologies: Star and hierarchical, in which the Protocols Internet Relay Chat (IRC) and Hyper Text Transfer Protocol (HTTP) are mostly used; like in the Chuck Norris or Aidra IoT botnet (Fig. 1(a)).
- Decentralized: Also known as Peer-to-Peer (P2P), in which the bot acts as a server and client, each one is connected to another bot at least. The commands can reach each bot only if all the bots are connected (Fig. 1(b)).

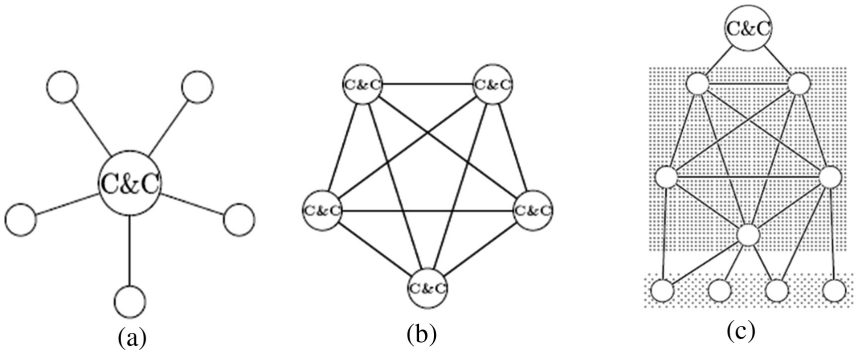

Fig. 1. (a) Centralized [9], (b) Decentralized [9] and Hybrid [9]

- Hybrid: It´s a combination of centralized and decentralized topologies since there are two types of bots, some of them have the functionality as servers and clients others just as clients, makes the message latency high (Fig. 1(c)).

Authors in [9, 11, 12, 28], described the communication between the C&C server and bots, as shown in Table 1.

Table 1. Communication C&C server and bots

	Advantages	Disadvantages	IoT Botnet application
IRC	Channel protected by authentication mechanisms	Use of a centralized topology, this means a single failure point for the botnet	Close ports that were used to gain access to IoT device, i.e., like in the IoT Linux/Hydra botnet
HTTP		Use of a centralized or P2P topology, in which case has to prevent loops or replicated messages	Such as the Mirai botnet gained remote access to the device over Telnet, SSH, or HTTP
SMB		Often blocked by a gateway, so is mainly used for local networks	Bashlite IoT Botnet makes use of this channel by downloading two scripts to gain remote access, then inject malicious code into files, generating keep-alive messages exchanged every 60 s between the botnet C&C and the bot

As mentioned in Sect. 2.1, for an IoT botnet the main botnet stages are the communication between the botmaster and the bots due to the botmaster does not have direct communication with the bots, it makes use of the C&C server.

According to the topology and communication channel, botnet detection can be attached by building a C&C server detector.

2.3 Botnet Communication

Therefore, an exchange of messages sequence is needed to achieve a specific task:

- Coordination: If the particular task is not automatic, it is necessary to instruct the bot what to do.
- Scan: Network scan, can be an ICMP echo request, UDP or TCP scan for vulnerable services.
- Data: It can be a binary bot, arbitrary files, or network data that communicates to the botnet or other bots.
- Register: Message required for a record.
- Execute: Tasks executed by bots.

As well, it can be separated as presented in Table 2.

Table 2. Botnet stages

Infection	Operation
• It is used to recruit new bots • It can be active (when the botnet tries to infect additional hosts) or passive (when the binary bot is distributed by other means)	• The C&C sends commands to the active bots • The stage in which botnet performs the task

According to Vormayr et al. [9] botnets continuously recruit new bots by infecting hosts for launching a specific attack, this infection stage occurs in two different ways as exposed in Table 3.

Table 3. Infection stage

Active infection	Passive infection
• Makes use of existing vulnerabilities • Through commands or automatically 1. Coordination: Scan and parameters configuration to be exploited 2. Scanning: Host detection achievable with an ICMP echo request or directly in the vulnerabilities of the UDP and TCP ports 3. Infection: Vulnerabilities exploitation 4. Download additional data 5. Registration: Monitoring the botnet status, as well as the size and location of the bots	• Via email, web pages, or storage media • Users infect their host by a click or an action • The binary bot is executed on the infected host • Downloading additional data 1. Register: Monitor the botnet status, as well as the size and location of the bots

These two infection ways are applied to IoT devices due to, i.e., infected cameras by exploiting existing vulnerabilities such as default credentials; or like tablets where the user infects their device by a click allowing the binary bot download, these being helpful to understand botnet communication and therefore install detection mechanisms.

2.4 Communication Hiding and Obfuscation

To increase the probability of survival, botnets tries to evade detection using hiding and obfuscation techniques in the operation stage [9, 12]:

- Covert channels: They are used to hide the existence of communication.
- Encryption: Used to hide transmitted commands as well as the protocol used. If a botnet uses a predefined key and fixed commands, the botnet can be disabled.
- Compression: A compression algorithm can hide the data.
- Steganography: Hiding information in non-communicating containers, similar to covert channels with the difference of being used as carriers.

Knowing botnet communication and obfuscation techniques, network-based signature detection can be applied. According to Vormayr et al. [9], the operation stage can be categorized, as shown in Table 4.

Table 4. Operation stage

Operation stage	Actions
Data Upload: Data collected from bots	• C&C sends commands to bots with specific instructions • Instructions are executed • Report results to C & C
Data Download: storing data onto bots, i.e., botnet binary update	• C&C indicates the bot which file to download and where to find it • Download file in the bot • Store or install the download
Forward Proxy: Uses the botnet to hide the actual origin of the communication	• Unidirectional: It consists of data sent from the bot to a target, used to overload a single service with requests • Bidirectional: It consists of requests and responses that transmit through the bot, used if the Botmaster or a third entity needs the results of a request
Reverse proxy: Reverse for retransmission to a specific source host	• C&C indicates the bot which port to open and which internal address should connect so that other hosts or bots can connect to the source
Instruction: Execute tasks on behalf of the Botmaster	• The botmaster indicates the bots what tasks must be executed and provides the necessary parameters • The current command executes

During the operation stage, the sequence of exchanging messages depends on the botnet specific task, IoT botnets examples are in Sect. 3.

Studying the exchanging messages could be possible to collect information packets and determine the type of botnet used so that detection mechanisms can be used.

3 IoT Botnet Attacks

The use of a botnet is for several purposes such as malicious activities, better known as attacks. In a DDoS attack, the network bandwidth is consumed by the compromised IoT devices injecting malicious packets into the network targeting a particular server; this means it floods the traffic with service request and processed by a server [29, 30].

Specifically, the most executed IoT botnets flooding attacks are:

- **TCP SYN Flood:** exploits a known weakness in the TCP connection sequence, the requester sends multiple SYN requests, but either does not respond to the host's SYN-ACK response or sends the SYN requests from a spoofed IP address. Either way, the host system continues to wait for acknowledgment for each of the requests, making binding resources until no new connections, and ultimately resulting in a denial of service.
- **UDP Flood:** a large number of User Datagram Protocol (UDP) packets are sent to a targeted server to overwhelm that device's ability to process and respond.
- **HTTP Flood:** the attacker exploits seemingly-legitimate HTTP GET or POST requests to attack a web server or application.
- **ICMP Flood:** overwhelms the target resource with ICMP Echo Request (ping) packets, generally sending packets as fast as possible without waiting for replies.
- **ACK/ACK PUSH Flood:** receives no legitimate ACK packets that do not belong to any of the sessions on the server's list of transmissions. The server under attack then wastes all its system resources (RAM, processor, etc.) trying to define where the packets belong. The results in productivity loss and partial server unavailability.
- **TCP XMAS Flood:** Send a very explicitly crafted TCP packet to a device on the network with FIN, URG and PUSH flags set.
- **DNS Flood:** Attackers send validly but spoofed DNS request packets at a very high packet rate and from a large group of source IP addresses. The DNS server can be overwhelmed by the number of requests.

A single IoT device as bot is not a threat, but the recruitment of several bots can attached flooding attacks, due to these devices can send connections requests affecting the availability of services. By determining the type of a botnet attack, it is possible to apply a network-anomaly detection technique.

4 Related Work

As briefly mentioned in Sect. 3, IoT botnets are performing many attacks. In this section, an IoT botnet comparison is done at Table 5 based on model architecture, attacks, target, operation, and propagation [6, 13–23, 28].

Table 5. IoT botnets

Botnet	Archit. model	Attacks	Target	Infection	Operation
Linux/Hydra (2008)	IRC	SYN & UDP Flood	Routing devices based on MIPS architecture	Active: Dictionary attack, known specific authentication vulnerability	Download data: the attacker had to edit one of the source files to provide the URL address of the C&C IRC server as well as the link to download the malicious binary
Psyb0t (2009)	IRC	SYN, UDP & ICMP Flood	Routers & DSL modems	Active: Access to Telnet and SSH using brute force with 6000 predefined user names and 13000 passwords	Download Data: Once a shell of the vulnerable device is acquired, Psyb0t downloads itself from a remote server
Chuck Norris (2010)	IRC	SYN, ICMP & ACK Flood	D-Link routers & DSL modems	Active: Brute force, as well as the authentication override vulnerability	Download Data: Download SSH
Tsunami (2010)	IRC	SYN Flood, UDP Flood & ACK Also, HTTP Flood & TCP XMAS	Linux Mint Official ISO	Passive: Modifies the DNS server setting in the configuration of the infected devices such that the traffic redirects from the IoT device to malicious servers	Download Data: Download file (s) from remote servers
LightAidra/Aidra (2012)	IRC	SYN Flood & ACK Flood	Architectures such as MIPS, ARM, and PPC	Active: Search open Telnet port using default credentials Propagates over TCP on port 23, for install a backdoor.	Download Data: Download a shell script from buonapesca.altervista.org The script downloads and executes

(*continued*)

Table 5. (*continued*)

Botnet	Archit. model	Attacks	Target	Infection	Operation
					additional malicious files, also receive a command from the IRC server to perform the attack
The Moon (2014)	P2P	Bring down websites and servers by overwhelming them with requests; obfuscate information online	Linksys, ASUS, MikroTik, and D-Link routers	Active: Exploiting a command vulnerability in the POST request parameter	Reverse proxy: download an additional proxy module that opens a SOCKS5 proxy on infected devices, who knows the web page and the parameters to which the redirection of the uplink node must be accessed, and the port no longer opens directly on the infected node
Bashlite (2014)	IRC	SYN, UDP, HTTP & ACK Flood	Linux-based IoT devices such as cameras and DVRs	Active: Brute force with default credential of devices with open Telnet ports	Upload Data: In the malware´s binary has the IP address of the C&C server. It also has the IP addresses hard-coded into it
Mirai (2016)	Centralized	Generates floods of GRE IP, GRE ETH, SYN, ACK, STOMP, DNS, UDP, & HTTP	Closed-circuit television, cameras, routers, and DVR	Active: 10 predefined attack vectors Dictionary attack based on 62 entries	Upload Data: loading the malware to the vulnerable IoT devices detected, scan the network for new victims while waiting for instructions from the C&C. Scan traffic uses random

(*continued*)

Table 5. (*continued*)

Botnet	Archit. model	Attacks	Target	Infection	Operation
					parameters to avoid identification and fingerprinting
Linux/IRCTelnet/New Aidra (2016)	IRC	SYN, ACK-PUSH, & UDP Flood	Routers, DVR, and IP cameras	Active: Brute force and code injection, list of Mirai credentials	Upload Data: In the malware´s binary has the IP address of the C&C server. It also has the IP addresses hard-coded into it

Table 5 lists the botnets analyzed in this work along with their topology, targets, types of attacks and the infection/operation stages studied in Sect. 2; each column gives information about how the IoT botnet operates, making possible the use of detection mechanisms. Some mechanisms such as signatures-bases detection are referring to the analysis of known or abnormal patterns or characteristics of threats from intruders into a system, or by honeypots, which are used as traps to collect bot´s information and activities, making possible to analyze them to detect botnets. These detections are performed when the attack is executed [31].

5 Discussion

Botnets grow in size and complexity as potential nodes in the Internet of Things time to time, making difficult to identify malicious from benign traffic even monitoring the common ports used, like 23,2323 and 22 to gain access to the IoT device.

From a diverse set of IoT attacks described in Sect. 3 and the botnet communication in Sect. 2, different botnets were chosen in the state-of-art to identify the type of infection and operation used. These botnets have been analysed according to their topology, target, and type of attack, for determining the way it infects new hosts and how it operates.

It was observed similar characteristics in most IoT botnets, such as centralized topology due to the IoT devices resources cannot function as a server the main cons of this topology is the single failure point by detecting the C&C server, although the pros are management and monitoring of the botnet by the botmaster that communicates directly with each bot, and the low latency.

As compare to decentralized and hybrid topology, there are more than one C&C server, and because of the IoT devices resources, these can not function as a server, these are cons. Although, the pros lies in the detection complexity of a failure point

because if one C&C server is taking down, the other servers can manage and monitor the botnet.

The application of a centralized topology still exists, by creating new botnets following the evolution of applications on the Internet. According to the previous research, it is provided a comprehensive state-of-art review of botnets that are evolving time to time making important and understandable point their application to a new target, IoT devices, highlighting how evolution and behaviour can be expanded to perform different attacks in the IoT context.

6 Conclusion

This paper presented a comprehensive review and analysis that discusses the IoT botnet development with distinct variation targets, attacks, and type of propagation and operation. Being helpful to have a better knowledge of the communication between different IoT botnets that use similar techniques or a combination of them, so that botnet prevention can be created or even disable the botnet, i.e., the use machine learning to extract message exchanges from network traffic for identifying possible botnet communication.

Acknowledgment. The authors would like to thank the Instituto Politécnico Nacional (IPN), the Centro de Investigación en Computación (CIC) and the Consejo Nacional de Ciencia y Tecnología (CONACYT) for the support in this research.

References

1. Escamilla-Ambrosio, P.J., Rodríguez-Mota, A., Aguirre-Anaya, E., Acosta-Bermejo, R., Salinas-Rosales, M.: Distributing computing in the internet of things: cloud, fog and edge computing overview. Stud. Comput. Intell. **731**, 87–115 (2018)
2. Madakam, S., Ramaswamy, R., Tripathi, S.: Internet of Things (IoT): a literature review. J. Comput. Commun. **03**(05), 164–173 (2015)
3. Nunberg, G.: The Advent of the Internet: 12th April, Courses (2012)
4. Stavrou, A., Voas, J., Fellow, I.: DDoS in the IoT Mirai and Other Botnets-2017-Computer (2017)
5. Botnet de IoT Homepage (botnet de internet de las cosas). https://searchdatacenter.techtarget.com/es/definicion/IoT-botnet-botnet-de-internet-de-las-cosas
6. Angrishi, K.: Turning Internet of Things (IoT) into Internet of Vulnerabilities (IoV): IoT Botnets, pp. 1–17 (2017)
7. Tyagi, A., Aghila, G.: A wide scale survey on botnet. Int. J. Comput. Appl. **34**(9), 9–22 (2011)
8. Zhaosheng, Z., Zhi, J.F., Guohan, L., Phil, R., Yan, C., Keesook, H.: Botnet research survey. In: Proceedings of the International Computer Software and Applications Conference, pp. 967–972 (2008)
9. Vormayr, G., Zseby, T., Fabini, J.: Botnet communication patterns. IEEE Commun. Surv. Tutorials **19**(4), 2768–2796 (2017)
10. Sundareswaran, N.: Botnet life cycle and topologies. Int. J. Pure Appl. Math. **119**(17), 421–429 (2018)

11. Dwivedi, S.K., Bist, A.S., Chaturvedi, P.K.: Recent trends in botnet research. Int. J. Eng. Sci. Res. Technol. **6**(7), 280–295 (2017)
12. Khattak, S., Ramay, N.R., Khan, K.R., Syed, A.A., Khayam, S.A.: A taxonomy of botnet behavior, detection, and defense. IEEE Commun. Surv. Tutorials **16**(2), 898–924 (2014)
13. De Donno, M., Dragoni, N., Giaretta, A., Spognardi, A.: DDoS-capable IoT malwares: comparative analysis and mirai investigation. Secur. Commun. Netw. **2018**, 1–30 (2018)
14. Spognardi, A., De Donno, M., Dragoni, N., Giaretta, A.: Analysis of DDoS-capable IoT malwares. In: Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, vol. 11, pp. 807–816, September 2017
15. Meidan, Y., et al.: N-BaIoT-network-based detection of IoT botnet attacks using deep autoencoders. IEEE Pervasive Comput. **17**(3), 12–22 (2018)
16. Durfina, L., Kroustek, J., Zemek, P.: PsybOt malware: a step-by-step decompilation case study. In: Proceedings of the Working Conference on Reverse Engineering WCRE, pp. 449–456 (2013)
17. Hallman, R., Bryan, J., Palavicini, G., Divita, J., Romero-Mariona, J.: IoDDoS – the internet of distributed denial of service attacks. In: IoTBDS (2017)
18. Janus, M.: Heads of the Hydra. Malware for Network Devices, 16 August 2011. https://securelist.com/heads-of-the-hydra-malware-for-network-devices/36396/
19. Barnett, R.: New Tsunami/Kaiten Variant: Propagation Status, 11 September 2018. https://blogs.akamai.com/sitr/2018/09/new-tsunamikaiten-variant-propagation-status.html
20. Symantec "Linux.Aidra" Writeup By: Kaoru Hayashi. https://www.symantec.com/security-center/writeup/2013-121118-5758-99
21. Cyware "The Moon IoT botnet is proxying traffic for Youtube ad fraud scheme", 1 February 2019. https://cyware.com/news/themoon-iot-botnet-is-proxying-traffic-for-youtube-ad-fraud-scheme-e17d6945
22. Netlab website. https://blog.netlab.360.com/themoon-botnet-a-review-and-new-features/
23. NJ Cybersecurity and Communications Integration Cell "Linux/IRCTelnet", 3 November 2016. https://www.cyber.nj.gov/threat-profiles/botnet-variants/linux-irctelnet
24. Silva, S.S.C., Silva, R.M.P., Pinto, R.C.G., Salles, R.M.: Botnets: a survey. Comput. Netw. **57**(2), 378–403 (2013)
25. AsSadhan, B., Moura, J.M.F., Lapsley, D., Jones, C., Strayer, W.T.: Detecting botnets using command and control traffic. In: Proceedings of the 2009 8th IEEE International Symposium on Network Computing and Applications NCA 2009, no. 4, pp. 156–162 (2009)
26. Symantec Internet Security Threat Report: Trends for July–December 2007 (Executive Summary), vol. XIII, April 2008
27. Makhdoom, I., Abolhasan, M., Lipman, J., Liu, R.P., Ni, W.: Anatomy of threats to the internet of things. IEEE Commun. Surv. Tutorials **21**(2), 1636–1675 (2019)
28. Ceron, J.M., Steding-Jessen, K., Hoepers, C., Granville, L.Z., Margi, C.B.: Improving IoT botnet investigation using an adaptive network layer. Sensors **19**(3), 1–16 (2019)
29. Ahmed, M.E., Kim, H.: DDoS attack mitigation in internet of things using software-defined networking. In: Proceedings of the 3rd IEEE International Conference on Big Data Computing Service and Applications, BigDataService 2017, pp. 271–276 (2017)
30. Gupta, B.B., Badve, O.P.: Taxonomy of DoS and DDoS attacks and desirable defense mechanism in a cloud computing environment. Neural Comput. Appl. **28**(12), 3655–3682 (2017)
31. Lange, T., Kettani, H.: On security threats of botnets to cyber systems. In: 6th International Conference on Signal Processing and Integrated Networks, SPIN 2019, pp. 176–183 (2019)
32. Spamhaus Malware Labs: Spamhaus Botnet Threat Report 2019, pp. 1–15 (2018)