A Survey: IoT-based DDoS Attack and Defense

Kenan Krijestorac
Department of Math and Computer Science
University of Missouri – St. Louis
St. Louis, United States
kkdd6@umsystem.edu

***Abstract*— The continuous growth of Internet of Things (IoT) devices and the security implications associated with the freshly developed technologies make them a breeding ground for a plethora of attacks. Distributed Denial of Service (DDoS) attacks in IoT refers to the targeting of servers to impede the availability of communication channels between the servers and IoT devices by generating false requests. Within recent years, various attack vectors have been devised in order to launch large scale attacks against a myriad of technologies. In this paper, we will discuss common IoT botnets, the classification of DDoS attacks, and potential defenses for IoT devices including detection, prevention, and mitigation. Additionally, we elaborate on tools that can be used to protect IoT devices and research challenges that need to be addressed for better IoT DDoS defense. This paper takes a different comprehensive approach compared to other various survey papers. Unlike other papers, our paper will provide a deep analysis of IoT botnets, DDoS attacks used by the botnets as well as on them, and defensive techniques for detection, prevention, and mitigation of DDoS attacks.**

Keywords—IoT, Distributed Denial of Service, DDoS, detection, mitigation, prevention, botnet

I. Introduction

The Internet of Things is an extensive network of physical devices connected to the Internet which allows devices to connect each other another and exchange data and services without any human interaction [3]. The IoT revolution has given way for devices dubbed as “dumb” to become “smart,” leading to the facilitation of monitoring, tracking, collection, etc., of user data. From healthcare monitoring devices to agricultural sensors, every residential, commercial, and industrial businesses is capable of benefitting from the information that IoT devices gather. Global businesses contest each other for consumers attention, with the sole purpose of selling their IoT devices and little to no regard to device security. Consequently, IoT security has vastly suffered and the market has been flooded with a deluge of insecure devices. As a result, more attack vectors have surfaced, giving attackers more ways to potentially target our sensitive data and devices.

The lack of security implementations has made IoT devices very susceptible to Distributed Denial of Service (DDoS) attacks. DDoS attacks aim to overwhelm the resources of the centralized architectures of servers and/or networks rendering them unavailable. Within recent years, the majority of IoT network traffic has been accounted for by botnets due to the vulnerabilities presented in devices [9]. Botnets are a massive network of IoT devices infected by malware that allow attackers to successfully launch DDoS attacks on servers or networks. Awareness about the insufficient security in IoT devices has grown tremendously after the 2016 Mirai botnet which was able to launch an incredibly large DDoS attack.

The motivation behind this paper is to address the quickly growing threats and vulnerabilities apparent in millions of IoT devices. This survey discusses the newest strains of the most common IoT botnets and how they are able to take control of devices. The classification of DDoS attacks utilized by the common IoT botnets and attacks used on IoT devices. This paper will also discuss defensive measures that can be taken to detect, prevent, and mitigate IoT DDoS attacks as well as research challenges.

This paper is structured as follows, in Section II, we describe IoT botnets in detail and a few tools attackers use to spread malware across IoT devices and how DDoS attacks are conducted. In Section III, we illustrate and elaborate on the classification of DDoS attacks carried out on and by IoT devices. In Section IV, we provide detection, prevention, and mitigation strategies for IoT devices. In Section V, we describe the current research challenges in IoT DDoS attacks. Lastly, in Section VI, we provide a conclusion of the paper as a whole.

II. Botnets in IoT

The rapid development of IoT devices has created a conundrum for cybersecurity specialists throughout the world. Weak security implementations within the growing market of IoT has resulted in larger and more complex DDoS attacks. Attackers are able to inject malware into a large amount of IoT devices by scanning networks for known vulnerable IoT devices allowing them to exploit those vulnerabilities. Thus, creating a massive network of infected IoT devices, or bots, known as botnets.

Some botnets are based on a centralized architecture which are controlled by a master bot that has complete control of all bots within the botnet, also known as the command-and-control (C&C). On the other hand, there are some botnets that are based on a decentralized architecture making them harder to discover. Several architectural models for botnets are described below:

Agent-Handler Model. This model is made of up clients, handlers, and agents. Clients are a device utilized by a malicious user to control the handlers, and agents. Clients can update agents or interact with handlers on the procedure that is going to be used to carry out a DDoS attack. A handler is a software package that compromises a machine and provides a means of communication between the client and the agents. An agent is the code used to carry out an attack [6].

Reflector Model. Similar to the Agent-Handler Model, the Reflector Model is composed of clients, handlers, and agents. However, handlers instruct agents to send packets to devices that are not infected, also known as reflectors, rather than directly sending them to victim directly. Reflectors have to be capable of responding to IP requests [6].

Internet Relay Chat (IRC)-based Model. IRC is a protocol used in the application layer to aid in communication via text. The IRC-based model is also similar to the Agent-Handler model but is instead based on a C&C infrastructure so that it connects the client to the bot(s) [6].

Web-based Model. Similar to the IRC-based model, the web-based model uses a website instead of IRC for communication. Some agents are responsible for communicating statistics back to the website, whereas other agents are used to carry out attacks that are accomplished through scripts and encrypted communication, such as HTTP and HTTPS protocols [6].

P2P-based model. The aforementioned models are based on a centralized architecture, whereas the P2P-based model is based on a decentralized architecture. This model allows bots to become both clients and servers. The workload is distributed among all peers within the network making it difficult to shut down [6].

There is a plethora of IoT botnets in the wild that use the aforementioned architectural models. One of the more famous botnets, Mirai, is based on the Agent-Handler model, and newly developed botnets are beginning to utilize the P2P model. P2P models are fault tolerant making it much more difficult to put out of commission. The following subsections will discuss large-scale botnets that have been discovered by researchers in recent years.

A. Mirai

The infamous Mirai botnet is responsible for bringing the lack of security within IoT devices to the attention of security professionals. Mirai malware was used in several large-scale DDoS attacks on various organizations including OVH, KrebsOnSecurity, DYN, etc. [2] The attack used over 500,000 IoT devices, such as CCTV cameras, DVRs, routers, etc., with an attack volume of 1.2 Tbps [5]. The Mirai source code was posted on various Internet locations paving the way for new child malwares.

Mirai botnet is based on an Agent-Handler architecture and can be used to launch a variety of DDoS including but not limited to SYN flood, UDP flood, ACK Flood, DNS Water Torture, and HTTP Layer 7 Flood. The Mirai infrastructure is composed of a C&C, reporting, and loading server as well as the bots (infected IoT devices). This malware targeted Linux-based IoT devices. Mirai was able to quickly infect vulnerable IoT devices by using already infected devices to scan IP addresses for similarly vulnerable devices. Once a device is discovered, a bot will prove Telnet port 23 and every ten attempts the bot will probe TCP port 23 [5]. When the target asks for user credentials, the bot will perform a brute-force dictionary attack using 62 common username/password combinations in order to successfully infect the IoT device [2], [6]. **Figure 1** provides an overview of communication in the infrastructure.

Mirai has been one of the most powerful botnets in recent years and was successfully used in various DDoS attacks and disrupt the availability of very well-known companies. However, a new botnet, Mozi, has developed within the past year and has accounted for the majority of IoT internet traffic.

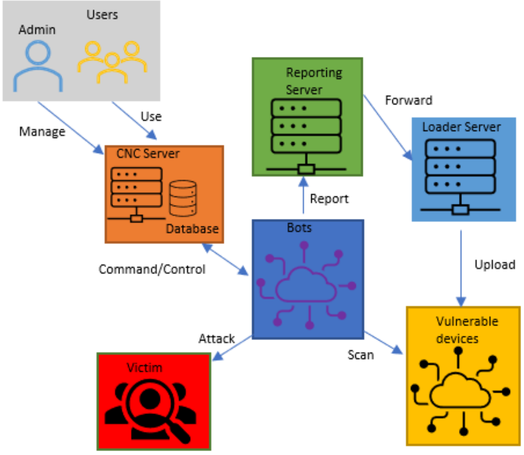

Figure 1: Mirai Infrastructure [6]

B. Mozi

Mozi botnet has been developed within the past couple years and has become a massive network of vulnerable and infected IoT devices. Unlike the Mirai botnet, Mozi is based on decentralized architecture. Mozi is a P2P malware that has infected several routers by Netgear, Huawei, and Eir as well as CCTV cameras and DVRs [8]. This botnet established its P2P network with the use of a custom DHT protocol and utilizes Mirai, Gafgyt, and Reaper/IoTrooper variants to launch its DDoS attacks [7].

The Mozi malware is spread via command injection (CMDi) attacks using the “wget” command. If the IoT device is vulnerable, the attacker downloads and executes the ‘mozi.a’ malware file on the device allowing it to infect the device, add it to the P2P botnet, and potentially download more malware. In order for the bot to join the DHT (P2P) network, the malware must generate an ID for the new device. The device will send an HTTP request to a URL specified in the malware so that it can be registered within the DHT and will send a query to a specified list of eight hardcoded DHT bots that allow the registering node to officially connect to the botnet [7]. Additionally, Mozi malware can brute force weak Telnet credentials using a hardcoded list to gain access. After successful execution of the malware, Mozi will synchronize the configuration file so that it will be able to carry out various instructions: DDoS attacks, update malware, collect and send bot information, etc. [8].

Mozi botnet is one of the more prominent botnets occupying the majority of internet traffic. Unlike the majority of botnets, Mozi is based on a decentralized architecture which makes it difficult to trace and shut down. Unlike Mirai, there have been no reported incidents of Mozi botnet being used in any large-scale DDoS attacks. Another similar botnet, named Hajime, uses variants of Mirai and is based on a P2P network.

C. *Hajime*

The Hajime botnet is spread through a similar means as that of the Mirai botnet, however, it has a decentralized architecture that is similar to the Mozi botnet. Hajime uses a peer-to-peer network and does not contain a C&C, but rather spreads those controls across the entire network [6]. Although Hajime is a malware that creates a botnet, it has not been used for any known attacks. Unlike the Mirai malware, the code behind Hajime conceals itself and attempts to prevent other malwares from taking over the IoT devices [6].

Hajime is a much more intricate and robust malware compared to Mirai, but the way that it is able to spread across various IoT devices is similar. The Hajime malware also attacks Telnet credentials using the same hardcoded list that Mirai uses along with an additional two credential logins. Hajime attacks on port 5358 as well as TCP port 23, a port that Mirai attacked. This malware blocks all access to various TCP ports on the IoT device in order to prevent any possible malicious use for DDoS attacks. Hajime, like Mozi, also uses a DHT protocol to facilitate the process of discovery of new IoT devices, and uTorrent Transport Protocol (uTP) for the exchanging of data. After successful download of the infection on a random high TCP port, Hajime uses UDP port 1457 so that it can use DHT and uTP [6][10].

Hajime was coined “white worm” because researchers believed it to be part of a “white hat” hacking attempt to secure a vast amount of IoT devices due to a message that it would display. However, the malware was still installed on a large number of devices without proper authorization. Hajime malware, like Mozi and Mirai, is a result of poorly configured IoT devices that are available to the public and are actively being used [10].

Mirai, Mozi, and Hajime are just a few of the IoT botnets that have been present within recent years. Mirai was an extreme powerhouse malware, based on an Agent-Handler architecture, that shocked the world of IoT and security professionals across the globe. Mirai was an awakening for developers and standard organizations around the world to develop more secure IoT devices and enforce more regulations and/or proper standards. In modern day, Mozi is responsible for the majority of IoT traffic and has infected millions of IoT devices globally. Mozi, based on a P2P architectural model, was constructed off several other malwares and infected millions and millions of devices, and become the leader in IoT traffic. These various IoT botnets can be used in launching various DDoS attacks.

III. CLASSIFICATION OF DDOS ATTACKS

Successful DDoS attacks require a large magnitude of bots to be able to take down well-known servers, such as those belonging to GitHub, KrebsOnSecurity, etc. [2]. Attackers rent or create their own botnet armies in order to launch these large-scale attacks on organizations. With the ever-growing IoT network, vulnerable IoT devices are flooding the market and offering attackers easy access into the networks to be able to launch attacks. IoT devices are now a prominent part of the modern world and can be seen in smart homes, cities, etc., and it is now more crucial than ever to secure these devices.

DDoS attacks can be categorized into two main categories: bandwidth and/or resource depletion. These classifications are determined by the different impacts that they have on a target system. Each category type aims to completely consume all bandwidth and/or resources of the target’s server or device. The server gets overwhelmed by the attack traffic being sent, and as a result cannot process legitimate user traffic and runs out of bandwidth/resources [1].

This paper will focus on the types of DDoS attacks that IoT botnets may potentially launch. The DDoS attacks that IoT networks face are similar in nature to that of large-scale systems; therefore, this paper will focus on attacks carried out IoT botnets rather than one that they may experience. Over the last several years, there has been a growing concern for the security of IoT devices because of various organizations pushing insecure devices to the marketplace. Thus, making various organizations more susceptible to wide-scale DDoS attacks. **Figure 2** outlines the classification of the two main categories of DDoS attacks.

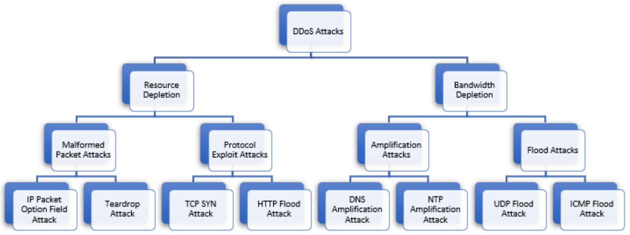

Figure 2: DDoS Attack Classification

A. *Bandwidth Depletion*

The goal of a bandwidth depletion attack is to drain all of the target system’s network bandwidth using some sort of botnet army, such as one consisting of IoT devices. This type of attack prevents legitimate users from being able to access the services of the server being attacked for an extended period of time until workers are able to mitigate the attack [4]. As demonstrated in **Figure 2**, bandwidth depletion attacks can be broken into amplification attack and flooding attacks.

Amplification Attacks. This type of attack method aims to increase the intensity of the packets being requested. The way malicious users carry out these attacks are by generating very large responses to the victim for extremely small requests. The large responses for the small requests are then redirected to a victim (target). Attackers will use bots, IoT devices in this case, to send those spoofed request packets with the source IP of the target in order to carry out the attacks. As a result, the bandwidth of the target systems is depleted and rendered unavailable to legitimate users [1][4][6]. Below, this paper will discuss two very common and powerful amplification attacks.

i. Domain Name Service (DNS) Amplification Attack. A DNS amplification attack, a very powerful attack based

Identify applicable funding agency here. If none, delete this text box.

on the exploit of the DNS protocol and infrastructure, aims to use the services of DNS Name Servers (NS) to deplete the resources of a victim. Attackers will send DNS queries, which tend to be small, to the name server and spoof the source IP with the address of the victim. Then, the NS will respond with large amounts of data to the victim A large number of small requests to the NS, with an IoT botnet, will subsequently cause in a large influx of network traffic to the victim and consume the majority of the victim's bandwidth. Consequently, the victim will no longer be able to process legitimate user requests and disrupt availability [6]. Another powerful amplification attack that is carried out in a similar manner to DNS amplification attack is a Network Time Protocol (NTP) amplification attack.

ii. Network Time Protocol (NTP) Amplification Attack. Similar to a DNS amplification attack, a NTP amplification attack takes advantage of the small size of queries and large responses to carry out the attacks. NTP utilizes a feature, MONLIST, which returns the last 600 IP addresses that connected to the NTP server. The MONLIST command/feature is only a mere 64 bytes which is much smaller than the significantly larger response that the NTP server will send to the attacker's victim. An attacker will use a botnet to request a large number of request queries, with a spoofed source IP address, to attack the victim. The victim's network bandwidth with be depleted by the extremely large responses of UDP traffic from the NTP servers [3]. Thus, creating a high volume of traffic and restricting any legitimate user requests to the victim server.

Amplification attacks are a very common and powerful DDoS attack type that are frequently used to render target servers unavailable for an extended period of time. With the rise of vulnerable IoT devices, the potential for large-scale DDoS attacks using amplification attacks has drastically increased as well. However, attackers utilize other various attack vectors that are available to them through their IoT botnet armies, such as Flooding attacks.

Flooding Attacks. This type of attack aims to use the weaknesses of how target systems have implemented particular protocols. Flooding attacks are focused on transport layer protocols like UDP, Internet Control Message Protocol (ICMP), etc. The overall goal of flooding attacks is the same as amplification attacks: render a target system unavailable. Attackers also use IoT botnets to carry out these attacks. Two very common attacks are to the UDP and ICMP protocols.

i. *UDP Flood Attack*. Like all of the aforementioned attacks, the ultimate goal of the UDP flood attack is to consume most of the bandwidth of the victim so that it will be unavailable to the legitimate users. Attackers will use their botnet armies to send a significantly large stream of UDP packets to the victim. The UDP packets may be targeted directly at specific UDP ports or may even be set for any randomly unsecured UDP ports on the system. The UDP packets have spoofed source IP addresses, so as the victim is processing the incoming packets, it will take time to respond with the respective ICMP packets for the spoofed IP addresses. Since the addresses are spoofed, the victim is spending a significant amount of time processing these packets to unreachable destinations. The attacker will send these UDP packets until the victim consumes all of its bandwidth, thus rendering it unavailable. The ICMP flood attack uses a similar approach in overwhelming the victim's bandwidth [1][3].

ii. *ICMP Flood Attack*. The ICMP flood attack, also known as the ping flood attack or smurf attack, aims to deliberately use up the bandwidth of a victim. Attackers will use the botnet army that is at their disposal to send a vast number of ICMP echo requests, using spoofed source addresses, to a victim's unprotected broadcast station. The attack repeatedly sends ICMP echo requests without waiting for an ICMP response message. As a result, the processing of the requests and the lack of time to respond, the bandwidth of the victim is quickly consumed and deemed unavailable [1][3].

Flooding attacks have the same end goal of amplification attacks: consume the victim's bandwidth. Flooding attacks are also very common type of DDoS attack used by botnets. Research challenges, discussed in Section V, still exist in completely preventing amplification and flooding attacks; however, attackers utilize other various attack vectors such as various resource depletion techniques.

B. Resource Depletion

Unlike bandwidth depletion attacks, the goal of resource depletion attacks is to consume the hardware resources of the victim rather than the network bandwidth resources. However, both resource and bandwidth depletion aim to take away the availability of the victim from legitimate users for malicious purposes [1]. As seen in **Figure 2**, resource depletion attacks can be categorized into malformed packet attacks and protocol exploit attacks.

Malformed Packet Attacks. This type of attack is relatively simple and is executed by sending malformed packets to a victim. The attack forces them to extraneously use resources in order to deal with the packet properly. An attacker that sends a consecutively large stream of packets causes the victim to waste hardware resources and eventually crash the system [6]. Two common malformed packet attacks are discussed below.

i. *IP Packet Option Field Attack*. In this form of attack method, the attacker will change the option fields of IP packets. When the victim receives the packet, it will spend additional time analyzing the bits that have been changed by the attacker and consumes a significant amount of time. With a large influx of these modified packets from a botnet, the victim's resources will be greatly consumed and potentially crash the system [1][4]. Another form of malformed packet attack, a teardrop attack, modifies packets in another manner to consume the victim's resources.

ii. *Teardrop Attack*. Attackers that utilize a teardrop attack are generating malformed (fragmented) packets and sends them to the victim. However, these malformed packets will have overlapping offset values. So, when the victim is reassembling the fragmented packets, the packets are invalid and cause the victim's system to crash. A stream of these malformed packet greatly wastes the victim's resources and result in availability of services to legitimate traffic [1][4]. Attackers also use another attack method known as protocol exploit attacks to carry out DDoS attacks.

Protocol Exploit Attack. This type of attack is similar to a flooding attack. This attack method uses weaknesses found in various network protocols in order to exhaust a victim's resources. A couple common attacks include TCP SYN and HTTP flood attacks [1].

i. TCP SYN attack. The nature of the TCP protocol leaves itself to vulnerability due to the need to complete handshaking to establish a proper connection. When an attacker uses bots to send TCP SYN packets with spoofed source IP addresses, the victim responds with TCP SYN/ACK packets to the spoofed address; meaning that the victim is waiting for ACK packets that will never be returned. The victim has a limited sized buffer for TCP connections. As a result of all these processed TCP SYN packets, the victim's resources are overwhelmed and rendered unavailable to any legitimate TCP connection requests [4]. An HTTP flood attack is conducted in a similar manner to a TCP SYN attack.

ii. HTTP Flood Attack. Attackers will deplete a victim's resources with an HTTP flood attack by either using a GET or POST request. In each type of request, the attacker will use the botnet to generate many legitimate requests to the server to consume its resources on a large scale. POST attacks are computationally intensive on system resources compared to GET; however, a large volume of either will drain the resources of the victim and restrict the processing of legitimate user requests [4].

Resource depletion attacks, like all other DDoS attacks, are devised with the goal of preventing legitimate users from accessing the services of the victim. The only difference is the means in which it is accomplished, and which object is being attacked directly such as network or hardware. The next section will discuss potential defensive measures for various DDoS attack methods.

IV. DEFENSIVE MEASURES

As depicted in the previous sections, defensive measures to detect, prevent, and mitigate IoT-based DDoS attacks are very desirable due to the fact that there is a myriad of attacks that malicious users can use. DDoS attacks are a massive threat to the resources, bandwidth, and infrastructure of the objects that are being targeted. With the great increase in vulnerable IoT devices in the marketplace, attackers are able to create much larger botnet armies to carry out their attacks. Therefore, it is critical for researchers and developers to devise possible defensive measures to protect our critical systems. The following subsections will discuss some promising and up-and-coming technologies that are being developed and researched to detect, prevent, and mitigate IoT-based DDoS attacks.

A. *Software-Defines Networks (SDN)*

Software-defined networks (SDN) have been a growing and developing technology that has taken root in many organizations. SDNs allow networks to be able to connect to switches via software. This technology decouples the control plane from the forwarding plane. SDNs are accessed through a centralized controller which allows for more flexibility, granular security, traffic programmability, etc. for the network as a whole [11]. Researchers are in the process of developing frameworks to aid in the detection, prevention, and mitigation of DDoS attacks generated by IoT-based botnets along with others. SDNs are capable of mitigating DDoS attacks to an extent based upon how engineers configure the network. A few promising SDN implementations and techniques are discussed in the following subsections.

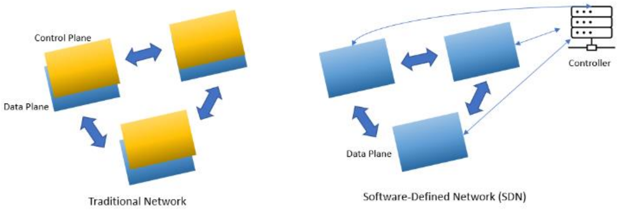

Figure 3: Traditional Network vs. SDN [6]

Honeypots. This type of technology is an extremely powerful detection, prevention, and mitigation tool that organizations can implement within their networks. Honeypots are used in conjunction with other network security features, such as intrusion detection systems (IDS), to properly handle any potential malicious Internet traffic. The purpose of a honeypot is to convince attackers into believing that the honeypot is a legitimate network so that the believes that he is carrying out an attack on the actual system itself. Then, the honeypot is able to analyze the contents of the network traffic being sent and take the appropriate measures to deal with it. A large number of packets containing the same source IP address can be dropped and prevent forwarding to the actual system [4]. **Figure 4** provides a visual representation of the implementation of a honeypot. Another up-and-coming implementation used to detect, prevent, and mitigate DDoS attacks is a defense mechanism called Learning Automata.

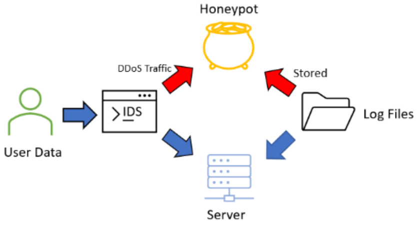

Figure 4: Honeypot Architecture [6]

Learning Automaton (LA). This type of implementation is used within a SDN environment to aid in the prevention and mitigation of DDoS attacks. LA uses a three-step process: detection, identification, and defense. The detection step will set a threshold value for resource utilization which determines if there is a chance for a DDoS attack to occur. If the threshold is surpassed, then packets being collected will be analyzed for various information such as device id, IP address, etc. In the defense step, sampling of the incoming packets will begin, and LA will discard any illegitimate packets [12]. Thus, preventing and/or mitigating the potential DDoS attack.

The detection, prevention, and mitigation SDN implementations mentioned in this section show strong promise through the use cases and simulated environments that they have been used in. These techniques are not a one-size-fits all solution for every network infrastructure. Another promising technology that aids in the defense of DDoS attacks is the use of blockchain.

B. *Blockchain*

Blockchain, a distributed ledger, is based on a decentralized architecture that consists of peers that are able to perform various actions and store information of a large number of transactions. Blockchain offers a transparency between the transactions as they are stored across a significantly large network. Researchers are using Ethereum blockchain and its use of smart contracts to aid in the mitigation of DDoS attacks [13].

A particular architecture proposed by Uzair Javaid et al. focuses specifically on IoT-based DDoS attacks. The proposal uses the Ethereum blockchain and smart contracts to register IoT devices with the server which is responsible for generating the smart contract. Before devices are able to send and receive any messages with other IoT devices, they must be registered. When the IoT devices are registered, they receive a gas limit, refers to number of resources that can expire, which only allows them to work up to that limit. The gas limit prevents the servers from getting overloaded with traffic. The server is capable of unregistering, removing, or revoking contracts for devices once the limit is reached or if malicious activity is suspected. The designed implementation is appealing because of the limit that is placed on each individual device and the idea that the devices are only capable of communication if they are on the smart contract's trusted list. If nodes that are not on the trusted list are attempting communication, then the communication is automatically discarded by the smart contract. Thus, preventing the chances of a DDoS attack from occurring [13].

A publication by Rajeev Singh et al. provides an extensive list of prototypes and proof of concepts (PoC) of blockchain in mitigating DDoS attacks [13]. The list is not exhaustive; however, it does provide a plentiful number of PoCs that can assist in further research and development. Blockchain itself is still gaining traction in the IoT world and the world of cybersecurity. It also provides potential security solutions for these concepts. Blockchain use in DDoS mitigation and prevention is still in very early developmental stages but has been deemed promising. Another growing concept in the mitigation of DDoS attack is through the use of machine learning.

C. *Machine Learning (ML)*

Machine Learning (ML) are utilized to detect anomalies within a network and are typically applied to an SDN. The techniques used to detect anomalies are through various ML algorithms, statistics, models, etc. This technique is also gaining traction amongst researchers due to the fact that it is able to detect malicious packets quickly with a high accuracy. ML implementation techniques vary depending on the type of algorithms/classifiers that are chosen and the data that is used to detect anomalies within the network [3].

Roshan Doshi et al. provides a framework that detects possible IoT-based network traffic that may contain DDoS traffic. The anomaly detection pipeline is composed of four steps: traffic capture, grouping of packets by device and time, feature extraction, and binary classification. The initial step captures traffic and records packet information such as source IP address, source port, etc. The next step, grouping of packets by device and time, separates and categorizes the packets by the information that was captured in the first step. The feature extraction step generates stateless and stateful features for every packet. These features are dependent on the domain information that the IoT devices contain. The final step, binary classification, will use various ML classifiers, such as deep neural networks, decisions trees, etc. in order to differentiate regular traffic from DDoS traffic [14]. ML techniques are seen as a detection and/or preventative technique.

ML needs to be implemented with additional security features in order to successfully mitigate potential DDoS attacks. This technique is capable of differentiating normal traffic from malicious traffic and offers high accuracy. ML has been successful within simulated environments; however, it still has yet to be used in a live environment. The ML technique mentioned above is just one of dozens that have been published and researched.

This section discussed various defensive measures that can be used to detect, prevent, and mitigate DDoS attacks. SDNs have made their way into live environments within organizations, but some techniques mentioned above have only been executed in simulated environments. SDNs are a promising technology that have helped organizations successfully set up defensive measures against DDoS attacks. Blockchain as a whole is still a new and evolving technology, particularly in its use with defending against DDoS attacks. There are no live implementations of blockchain but does provide effective use cases that utilize Ethereum blockchain to limits communication with the use of smart contracts. ML uses algorithms and classifiers to separate malicious traffic and regular traffic. ML implementations have been used in live environments; however, they have not directly been used to differentiate DDoS traffic and normal traffic. Many defensive measures to detect, prevent, and mitigate DDoS traffic is still in early stages of development and research. The next section will discuss some of the research challenges associated with the development of DDoS defensive techniques.

V. Research Challenges

Researchers have been able to closely analyze the various types of DDoS attacks launched on systems throughout the years. As a result, they have been able to devise some detection, prevention, and mitigation techniques for certain attacks. The issues that come to light are that no one network is configured in the same exact manner. Different network configurations are vulnerable to different types of DDoS attacks and attackers are able to determine these vulnerabilities through reconnaissance.

The defense mechanism presented in Section IV offer some possible solutions, but each comes with their own research challenges. Honeypots and LA that are implemented in an SDN, blockchain, and machine language can all possibly misinterpret malicious packets as legitimate packets (false positive) which can potentially lead to a DDoS attack. Another research challenge presented is that many of these defensive measures are still new technologies. They require further development and research before they can effectively be deployed into a live environment. Furthermore, many of the defensive measures mentioned above are PoCs and have been simulated in small-scale networks. The time to test and deploy these measures are years away which leaves many systems vulnerable to attack. Defensive measures for DDoS attacks are a work-in-progress and will need to evolve with attackers as time progresses.

VI. Conclusion

In this work, we discussed botnets in IoT, classification of DDoS attacks, and defensive measures for DDoS attacks. Mirai, Mozi, and Hajime are very powerful IoT botnets that have the capability to launch large-scale DDoS attacks on a specified target. Each botnet can be classified into a particular architectural model type such as agent-handler, reflector, IRC-based, web-based, and P2P-based. Next, we discussed the classification of DDoS attacks: resource depletion and bandwidth depletion. Each classification can be broken down into further sub-classes. Resource depletion can be further subcategorized into protocol exploit and malformed packet attacks, and bandwidth depletion can be subcategorized into flooding and amplification attacks. The more common attacks for each subcategory are discussed. Lastly, we discuss potential defensive measures that can be used to detect, prevent, and mitigate DDoS attacks. Some promising technologies include SDNs, blockchain, and machine learning. More research and testing need to be devoted to the defensive measures in order to better protect our systems.

References

[1] Mikail M. Salim, Shailendra Rathore, and Jong H. Park, "Distributed Denial of Service Attacks and Its Defenses in IoT: A Survey", The Journal of Supercomputing, Volume: 76, Issue: 7, pp:5320-5363, July 2020. https://doi.org/10.1007/s11227-019-02945-z.J.

[2] Georgios Kambourakis, Constantinos Kolias, and Angelos Savrou, "The Mirai Botnet and the IoT Zombie Armies", IEEE Military Communications Conference (MILCOM), pp:267-272, October 2017. https://ieeexplore.ieee.org/abstract/document/8170867.

[3] Ruchi Vishwakarma and Ankit K. Jain, "A Survey of DDoS Attacking Techniques and Defence Mechanisms in the IoT Network", Telecommunications System, Volume: 73, Issue: 1, pp:3-25, January 2020. https://doi.org/10.1007/s11235-019-00599-z.

[4] Tasnuva Mahjabin, Yang Xiao, Guang Sun, and Wangdong Jiang, "A Survey of Distributed Denial-of-Service Attack, Prevention, and Mitigation Techniques", International Journal of Distributed Sensor Networks, Volume: 13, Issue: 12, December 2017. https://doi.org/10.1177/1550147717741463.

[5] Artur Marzano, David Alexander, Osvaldo Fonseca, et al., "The Evolution of Bashlite and Mirai IoT Botnets", 2018 IEEE Symposium on Computers and Communications (ISCC), pp:00813-00818, June 2018. 10.1109/ISCC.2018.8538636.

[6] Michele De Donno, Nicola Dragoni, Alberto Giaretta, and Angelo Spognardi, "DDoS-Capable IoT Malwares: Comparative Analyis and Mirai Investigation", Security Communications and Networks, Volume: 2018, February 2018. https://doi.org/10.1155/2018/7178164.

[7] Tara Seals. "Mozi Botnet Accounts for Majority of IoT Traffic." threatpost.com. https://threatpost.com/mozi-botnet-majority-iot-traffic/159337/#:~:text=Once%20it%20cracks%20a%20device,join%20the%20botnet%27s%20P2P%20network. (accessed Nov. 2, 2020).

[8] Alex Turing and Hui Wang. "Mozi, Another Botnet Using DHT." Netlab.360.com. https://blog.netlab.360.com/mozi-another-botnet-using-dht/ (accessed Nov. 2, 2020).

[9] Dave McMillen. Wei Gao, and Charles DeBeck. "A New Botnet Just Into Town." SecurityIntelligence.com, https://securityintelligence.com/posts/botnet-attack-mozi-mozied-into-town/ (accessed Nov. 2, 2020).

[10] "Hajime Worm Battles Mirai for Control of the Internet of Things". Broadcom.com, https://community.broadcom.com/symantecenterprise/communities/community-home/librarydocuments/viewdocument?DocumentKey=d7cadaf3-4bd7-440c-a6e7-a2ea386b0670&CommunityKey=1ecf5f55-9545-44d6-b0f4-4e4a7f5f5e68&tab=librarydocuments (accessed Dec. 11, 2020).

[11] IBM Services, "What is Software-Defined Networking (SDN)?", IBM.com, https://www.ibm.com/services/network/sdn-versus-traditional-networking (accessed Dec. 11, 2020).

[12] Kshira Sagar Sahoo, Mayank Tiwary, Sampa Sahoo, et al., "A Learning Automata-based DDoS Attack Defense Mechanism in Software Defined Networks", Proceedings of the 24th Annual Internation Conference on Mobile Computing and Networking (MobiCom'18), pp. 795-797, October 2018. https://doi.org/10.1145/3241539.3267764.

[13] Uzair Javaid, Ang Kiang Siang, Muhammad Naveed Aman, and Biplab Sikdar, "Mitigating IoT Device based DDoS Attacks using Blockchain", Proceedings of the 1st Workshop on Cryptocurrencies and Blockchains for Distributed Systems (CryBlock'18), pp. 71-76, June 2018. https://doi.org/10.1145/3211933.3211946.

[14] Rohan Doshi, Noah Apthorpe, and Nick Feamster, "Machine Learning DDoS Detection for Consumer Internet of Things Devices", 2018 IEEE Security and Privacy Workshops (SPW), pp. 29-35, May 2018. 10.1109/SPW.2018.00013.