The Journal of Supercomputing (2020) 76:5320–5363
https://doi.org/10.1007/s11227-019-02945-z

Distributed denial of service attacks and its defenses in IoT: a survey

Mikail Mohammed Salim[1] · **Shailendra Rathore**[1] · **Jong Hyuk Park**[1]

Published online: 10 July 2019

Abstract
A distributed denial of service (DDoS) attack is an attempt to partially or completely shut down the targeted server with a flood of internet traffic. The primary aim of this attack is to disrupt regular traffic flow to the victim's server or network. DDoS attacks are volumetric attacks, and non-legacy IoT devices with low security such as webcams, baby monitoring devices and printers are compromised to form a botnet. High traffic from compromised IoT devices is rerouted to servers to disrupt their regular services. DDoS attacks are to an extent covered in the research literature. However, existing research do not discuss all DDoS attacks on general servers and botnet attacks on IoT devices and suggest few detection and mitigation solutions which are limited to addressing attacks on the cloud environment. Existing survey focuses either on the cloud layer or the IoT layer. A complete survey of DDoS attacks for both IoT and the cloud environment is not present in the current literature. Our survey is a comprehensive approach which includes general DDoS attack motivations and specific reasons why attackers prefer IoT devices to launch DDoS attacks. Various attack methods to compromise IoT devices and tools used to deploy botnet-infected IoT devices for DDoS attacks on the cloud layer are presented. A detailed attack classification on IoT devices and the cloud environment is presented considering that IoT devices are first compromised and then used by attackers against their primary targets on the cloud layer. Various state-of-the-art defense measures in the current literature for defense against DDoS attacks are present. Suggestions to implement an essential first line of defense for IoT devices are suggested. Our paper, to the best of our knowledge, is first to provide a holistic study of DDoS attacks from IoT devices to the cloud environment.

Keywords Distributed denial of service attacks · Security detection · Security prevention and mitigation · Internet of things · Security and privacy · Cloud computing · Edge computing

✉ Jong Hyuk Park
jhpark1@snut.ac.kr

Extended author information available on the last page of the article

1 Introduction

Distributed denial of service (DDoS) attacks are a constant threat to cybersecurity since the first attack in 1999 against the University of Minnesota [1]. A bandwidth depletion attack using UDP flooding technique was carried out for 2 days. In 2016, a Web site of a security consultant Brian Krebs and a French Webhost were targeted with an attack traffic of 620 Gbps and 1.1 Tbps, respectively. The attack was named Mirai, a Japanese word which translates to "The Future" [2] using a collection of 600,000 infected IoT devices. With the public release of the Mirai source code, more attacks followed including the famous Dyn attack which was of a volume of 1.2 Tbps [3]. The attack brought down hundreds of Web sites including Netflix, Twitter, GitHub and Reddit. The IoT environment is severely vulnerable to DDoS attacks and from being used to launch DDoS attacks on other targets. The growth of DDoS attack volume is steadily increasing. It is observed to be operating at a volume of 100 Gbps in 2013 and 2015. There is an increase in attack volume reported in 2016 and 2017 at 800 Gbps and 1.35 Tbps, respectively [4]. With the introduction of non-legacy IoT devices, DDoS attacks have grown to be more dangerous. Attackers are now able to exploit the poor security implementation in IoT devices which allows them to hijack and use them to attack the intended server or network. Figure 1 highlights the steady growth of spending on security on IoT by organizations. It is observed that as the spending is growing on utilization of more IoT devices, the attack volume has also increased. As per a survey conducted by Gartner in 2018, nearly 20% of organizations observed that they had experienced at least one DDoS attack. The current security spending on IoT in 2018 was at 1.5 billion USD and is forecasted to grow to 3.1 billion USD by 2021 [5]. There is an increasing industry focus on IoT security spending which

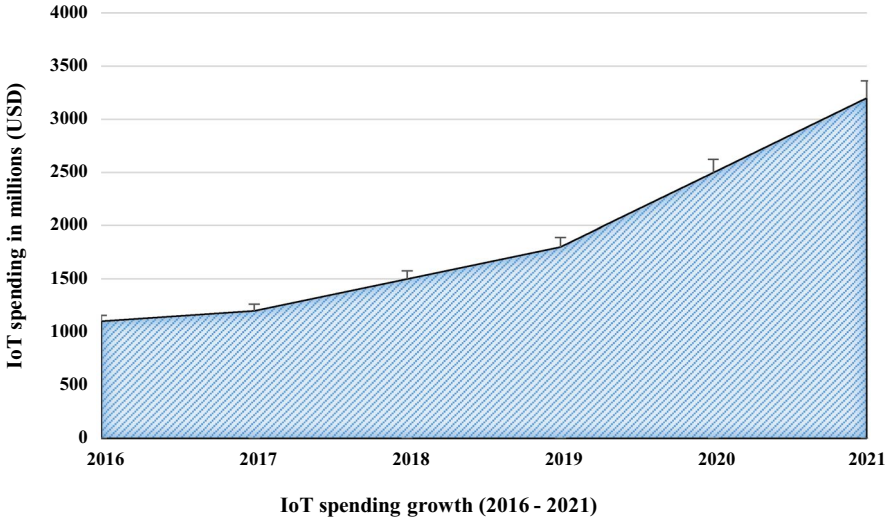

Fig. 1 Industry security spending on security for IoT in millions (US dollars) and future forecast [5]

points to the need for innovative defense mechanisms for IoT environment to be discussed.

The primary target of the DDoS attack is to deprive victims of the network and bandwidth resources resulting in a denial of service to honest users [6]. The attacker initially targets non-legacy IoT devices such as CCTV cameras, webcams, baby monitoring devices and smart watches, which have poor built-in security and suffer from other constraints such as low computational power and battery capacity. Due to their weak security, an attacker injects malware using tools such as the Mirai code or the LizardStresser tool and seizes control of the device. An infected device is called a bot, and multiple bots grouped together are termed as a botnet. Unlike traditional botnets which are used to steal bank credentials, spam the user with advertisements and cause click frauds, IoT botnet is used for launching DDoS attacks on servers. These compromised devices are then used collectively to form large volumetric attacks such as TCP, ICMP and UDP flooding attacks and HTTP GET/POST attacks using open-sourced tools such as GoldenEye, LOIC, Pyloris and DDoSim.

Several DDoS attack surveys as shown in Table 1 have been presented in the research literature which does not cover all aspects related to DDoS attacks. Many among the existing studies are focused on attack taxonomies and defense mechanisms based only on the cloud layer. Alzahrani et al. [7] and Chaudhary et al.'s research [8] focused on identifying DDoS attacks on cloud environments. They do not provide an attack taxonomy for types of attacks on the cloud layer. They discuss defensive techniques to defend against DDoS attacks on the cloud layer. Many of the DDoS defense survey researches are severely limited in the discussion of detection, prevention and mitigation defense techniques against DDoS attacks. Wani et al.'s [9] research is focused primarily on defense techniques which implement machine learning methods to secure the cloud layer from DDoS attacks. They compared various machine learning algorithms to determine the efficiency of each model in detecting DDoS attacks. Malik et al. [10] presented a survey which mentions a basic IoT attack taxonomy based on the IETF standard protocol layers which include the application, transport, network, adaptation and physical layer. It does not mention any attack tools used to infect IoT devices, nor does it suggest any defensive measures for IoT devices against DDoS attacks. Patgiri et al. [11] presented a survey for defending against DDoS attacks; however, it is limited to a single type of defensive measure using Bloomfilter technique and does not present details about attack types on the IoT or cloud layer. To summarize, existing recent literature do not provide a complete overview of the threat of DDoS attacks in cybersecurity. They either focus on only one environment, IoT or the cloud. A detailed taxonomy of attacks for both IoT and the cloud environment is not present. They do not discuss about any recent botnet attacks or the tools that attackers use to infect IoT devices and attack servers and networks with DDoS attacks.

Our survey paper implements a comprehensive approach where we present the motivations for attackers to use IoT devices to launch DDoS attacks. Attack tools are described to infect IoT devices with botnet malware and to launch DDoS attacks on networks and servers. New and evolving attack patterns such as multi-vector attacks are presented along with a detailed taxonomy of attacks on both IoT layer and the cloud layer. Three types of defensive methodologies are included,

detection, prevention and mitigation of DDoS attacks which include several state-of-the-art DDoS defense mechanisms. We cover both IoT and the cloud environment in detail, providing a complete view of DDoS attack and their defensive measures. This survey paper presents a detailed description of DDoS attacks from the formation of a botnet of IoT devices to implement them as sources of DDoS attack traffic.

The motivation of this article is to address the growing and severe threat to cybersecurity which has increased with the exploitation of millions of insecure IoT devices present in the market. The main contribution and significance of this comprehensive paper which covers all important aspects of DDoS attacks on IoT and the cloud environment are listed below:

- This paper presents the motivations for attackers to use non-legacy IoT devices to launch DDoS attacks. General motivations to launch DDoS attacks are also discussed.
- We discuss the evolving attack patterns from a single pattern of DDoS attack to multi-vector DDoS attacks.
- Separate attack tools employed to infect IoT devices and general tools which use infected IoT devices to launch DDoS attacks on servers and networks are examined.
- A highly detailed taxonomy for attacks on IoT devices which details attack methods used to infect non-legacy IoT devices to form a botnet is included Furthermore, the types of attacks that occur on individual layers of IoT devices are presented.
- A taxonomy of attacks on the cloud layer based on the attacker's objective of bandwidth or resource depletion attacks is discussed. The different types of attacks on the cloud are also presented.
- Twenty-one detection, prevention and mitigation techniques are presented in detail for defending against attacks on IoT devices. We discuss their implementation environment and how they improve upon existing systems and related work.
- Basic defensive measure to act as a first line of defense for non-legacy IoT devices is discussed.

This paper is structured as follows, in Sect. 2, we discuss attack targets, motivations for using IoT devices for attacks, the rise of multi-vector attacks using IoT devices and the tools used to infect IoT devices and conduct DDoS attacks. In Sect. 3, we provide the overall classification of DDoS attacks on the cloud layer based on their objectives. In Sect. 4, we discuss the methods used to infect IoT devices to be used as bots and the DDoS attack types on the IoT layer. In Sect. 5, we review the various state-of-the-art literature on detecting, preventing and mitigating DDoS attacks on IoT and present additional mechanisms that can be deployed to protect IoT environment. In Sect. 6, we discuss important findings in the paper and suggest improvements that can be made to provide necessary defenses on IoT devices to serve as an effective first line of defense against attackers. Finally, in Sect. 6, we conclude the paper.

Table 1 Contribution of our study in related with existing surveys

Research work	Year	Attack types (IoT + Cloud)	Attack tools (IoT + Cloud)	Prevention	Detection	Mitigation	Discussion and suggestions
Alzahrani et al. [7]	2018	Cloud	No	No	Yes	No	No
Chaudhary et al. [8]	2018	Cloud	No	No	No	No	No
Wani et al. [9]	2019	No	No	No	Yes	No	No
Malik et al. [10]	2018	Limited but only in the IoT layer	No	Yes	Yes	No	No
Patgiri et al. [11]	2018	Limited and only in the cloud layer	No	No	Yes	No	No
Our survey	2018	IoT and Cloud layers	Yes	Yes	Yes	Yes	Yes

2 DDoS attack targets and motivations for IoT

DDoS attacks have been targeting many industries over the years, and among them, the biggest brunt of them is felt by the Gaming industry. With over 79% of all DDoS attacks targeted at them, the gaming industry has been a constant target and as such had to face financial implications [6]. Many of the gaming companies such as Sony and Microsoft offer online multiplayer and downtime for even a few hours can deliver heavy financial losses. The Internet and Telecom industry which provides many services including Voice over Internet Protocol (VoIP) faces a substantial threat from DDoS. One such incident involved Skype by Microsoft, which was attacked and many of its users were unable to use the service due to poor connectivity [12]. The Financial Industry which is now more inclined toward online transactions has been under constant threat. In 2013, many American banks were targeted, and in 2016, the HSBC bank faced a DDoS attack making it difficult for its customers to access its services [13]. Cryptocurrency such as Bitcoin has also been under constant DDoS attacks.

2.1 Attack motivation

We discuss underlying reasons why attackers choose IoT devices to launch DDoS attacks. There has been a growing trend of DDoS attack method used that exhibits an increase in attacker's preference in using IoT devices to launch DDoS attacks. These devices have proven to be lacking in necessary security protocols which make them easy to take control. An attacker can infect an IoT device and spread the infection to other devices until it forms a collection of devices referred to as a botnet. Below we mention the deciding reasons that make IoT devices an attacker's choice to launch DDoS attacks.

- *Continuously connected* The Internet of Things, as the name suggests, are devices which are always connected continuously to the internet. They are available to be infected throughout the day, throughout the year and are never shut down.
- *Lack of basic security protocols* Many of these devices come with no necessary security protocols set in place from the manufacturers. They are often found to contain backdoors that allow them to be easily exploited.
- *Easily exploitable passwords* Owners of these devices rarely ever change the default password set by the manufacturer. It is common to find IoT devices sharing the same username "root" and "password" as the default set password. Attackers gain swift access to such devices.
- *Inability to reset authorization* Infected IoT devices once taken control over, leave the attacker to change the security credentials of the device unopposed. Should the infected device be ever traced during an attack, the owner of the device or the manufacturer cannot reset the security credentials to take control

back from the attacker. The attacker will use the device to cause as much damage to the victim for as long as possible.
- *No security firmware updates* Manufacturers do not monitor the device's security credentials once they get shipped to the market. There are much security loopholes found in the code which are exploited by the attackers. The manufacturers do not release security updates for these devices addressing the faulty software.
- *Cost-effective* IoT devices are not only easy to hack through but are also cost-effective. Instead of investing and maintaining expensive servers to launch powerful DDoS attacks, attackers can seize control of insecure IoT devices at zero or a fraction of the cost of maintaining a server.

There are various possible motivations to launch DDoS attacks. The following motivations are classified under five types [14, 15] and explain why an attacker would attempt to bring down a server or network:

- *Political beliefs* Some attackers with strong ideological beliefs or a sense of patriotism feel the need to challenge an opposing view of a rival organization or nation. The 2008 Olympics incident where a CNN reporter questioned the Olympics preparation which led to what is believed to be an attack conducted by Chinese hackers to disrupt services provided by CNN. In 1997 during the Russian elections, Gary Kasparov, the chess Grandmaster, his Web site was attacked by a robust DDoS attack which resulted in his political party's Web site being inaccessible for many days.
- *Cyberwarfare* Armed groups or government forces often to humiliate their rivals, attack their leader's or their government's Web site and deface it. Many such attacks occur on nations such as South Korea, Russia and Georgia have often been victims of Cyberwarfare. The Syrian Electronic Army emerged in 2011 in support of their president attacking many western news outlets and human rights organizations that were critical of him.
- *Business rivalry* Often businesses to overpower their rivals launch DDoS attacks to steal customers. Attacking their opponents ensures that their customers would be unable to use their services and instead flock to those of the attacker's. Customers, when faced with days of the denial of service, lose trust in the organization's ability to serve them. Such practices are not uncommon among gambling Web sites.
- *Financial benefits* Attackers often when attacking organizations leave a demand of ransom that if they wish to be relieved of the attack, then they must pay up or face a constant threat from them. While paying them would alleviate the organization, it also encourages attackers to attack again in the future. The 2018 DDoS attack on GitHub received a ransom demand of $15,000 embedded in a line of python code which they chose not to pay and resisted the attack.
- *Intellectual challenge* Attackers who wish to show their abilities among their communities of hackers commit a DDoS attack against others. There are many readily available online hackings tools and botnets that allow them to launch attacks. These are usually young enthusiasts wishing to make themselves famous.

2.2 Attacks on IoT devices

Attackers use non-legacy IoT devices as a means of launching DDoS attacks on servers and networks. These devices are equipped with low battery, and computational power allows an attacker to easily infect them. Non-legacy IoT devices include routers, internet-connected audio speakers, CCTV and webcam cameras, office equipment such as printers and home appliances such as internet connectivity-enabled thermostats, refrigerators, televisions and home security systems. Botnet attacks are possible on non-legacy IoT devices due to their weak inbuilt security. Legacy devices implemented in industrial domains are supported by high computational power systems. Though there are threats to the industrial domains against DDoS attacks, they are, however, from external non-legacy IoT devices and not significantly from their own legacy-based IoT devices. To launch a DDoS attack as a substantial volumetric attack, an attacker first needs to form a botnet. A botnet is a collection of compromised non-legacy IoT devices known as bots. Their security has been compromised by an attacker using a brute-force method where an attacker is able to hack the authentication protocols and gain access. Often IoT device manufacturers design their products using a similar password for all devices. The password is set as 'password', and user ID is 'admin'. An attacker who is aware of a single device's security credentials can gain access to a multitude of unsecured devices. The owner of the IoT device, the host, is unaware that their device's security has been compromised. The device, on the other hand, is in control of the attacker and they launch a DDoS attack by broadcasting packets from hundreds of thousands of infected devices toward their target. The attacker need not spoof the address of the packets sent as the source of the broadcast originates from other unaware device owners. There have been many botnet attacks, and among them BashLite, Mirai and Reaper are recent and popular:

- *BashLite* A popular malware whose primary target was Linux-based non-legacy IoT devices such as cameras and digital video recorders (DVR). The malware is responsible for infecting over a million IoT devices. The attacker can bypass the security protocols by applying brute-force method on the telnet access. The attacker is aware of the username and password for authentication can take control of the device. This botnet can launch DDoS attacks such as UDP and TCP flooding attacks and HTTP attacks with a volume capacity of 400 Gbps. The source code for this malware was leaked in 2015 and since then it has evolved and infected other IoT devices.
- *Mirai* Mirai botnet attack resulted in a large DDoS attack with a volume of 1.1 Tbps using 148,000 infected IoT devices. The botnet infected non-legacy IoT devices such as CCTV cameras, DVR and routers. It is far more dangerous than the BashLite attack and was able to infect 4000 IoT device every hour. The range of infected devices covered 164 countries including Brazil, China, USA and Vietnam. This botnet can generate DDoS attacks such as SYN and ACK, UDP flooding, HTTP traffic and DNS attacks. The Mirai source code was published on Github and since then it has evolved and infected more IoT devices. While IoT devices have been available from much earlier, it was the 2016 Mirai botnet-

based DDoS attack which brought to attention that IoT devices pose a significant threat to cybersecurity.

- *Reaper* The Reaper botnet is a variant of the Mirai code and is even more dangerous. Unlike the Mirai botnet which infected IoT devices using their default credentials, Reaper is known to exploit other security vulnerabilities which are present in the code of the IoT devices. Infected devices include CCTV cameras and routers. The botnet implements a lightweight programming language known as LUA to launch DDoS attacks on these computationally weak IoT devices.

IoT devices are known to be used as a foundation to launch large DDoS attacks such as flooding and HTTP attacks. However, the attack pattern has now evolved, and it includes a new pattern known as Multi-vector attacks. Multi-vector attacks using IoT devices have grown in frequency over the years. Attackers would earlier initiate a single type of attack such as UDP flood attack, ICMP flood attack HTTP GET attack. However, over the years DDoS attacks have grown in complexity and are now a combination of multiple attacks are used to evade the defenses deployed by their victims. The attacker aims to disrupt services on a periodic basis. For example, the attacker may launch a single form of attack and stop leaving the victim to recover. As soon as the victim recovers, another attack is launched forcing the victim to deny service to its users. The cycle of attack will repeat. The attacker begins with a flooding attack initially during the first stage leaving the victim to initiate application layer defenses. The second attack, however, will be an amplification attack which will bypass the defenses set up and hit the victim's servers. Among all the DDoS attacks launched in the second quarter of 2018, single vector attacks and multi-vector attacks are known to comprise 52.03% and 47.97%, respectively. Figure 2 displays the current forms of multi-vector DDoS attack combinations [16].

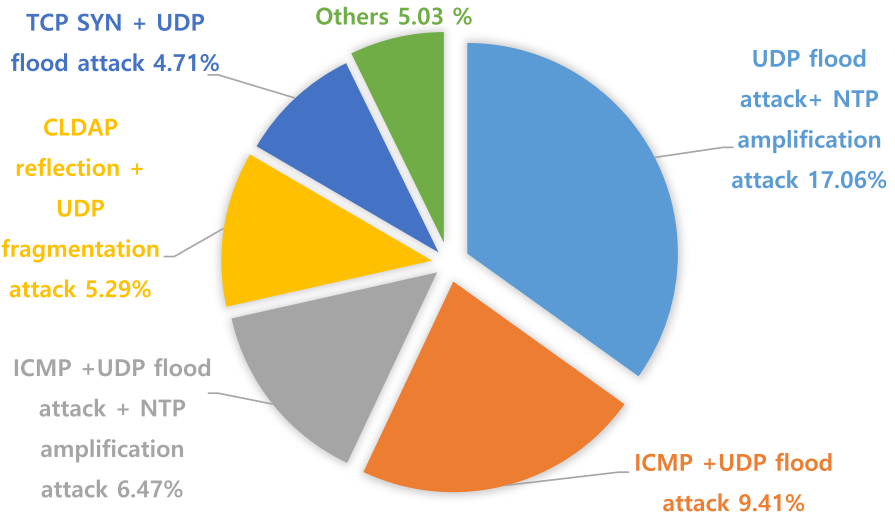

Fig. 2 Current form of the multi-vector DDoS attacks [16]

2.3 Botnet and DDoS attack tools

Several tools exist that are used to launch botnet attacks. These tools allow an attacker to form a collection of infected devices. These tools also enable to detect other botnets present in the IoT devices. If a device is pre-infected with another botnet, it will remove it and seize control over to disallow any other attacker from using it when attempting to launch a DDoS attack. We discuss some of them as follows:

- *LizardStresser* This botnet is a simple tool written in the C language and is capable of operating on IoT devices. It implements a brute-force method to login to different IP addresses using default manufacturer set user credentials. Devices with authentication protocols which cannot be altered by the user and are hard-coded by the manufacturer are likely to be infected using this tool.
- *Nitol* This tool allows malware to infect an IoT device and communicates with the Command and Control server via a TCP socket. It transmits the device's information to the attacker, such as computational power.
- *Mr Black* This tool is known to infect routers by contacting remote servers, which acts as a Command and Control. The device sends its performance information to the server. The server keeps information about all infected device. This tool also allows the device to download an executable file via a control command to launch a DDoS attack and then terminates the connection.
- *Mirai code* This tool is written in the Go language, and it locates unsecured IoT devices by scanning different IP addresses. Once it locates an IP address of an unsecured IoT device, it attempts to access it via a brute-force method by guessing the password and ID using known manufacturer's default set user credentials. It can launch DDoS attacks on servers and networks such as SYN-ACK, UDP, DNS and HTTP flood attacks.

There are several readily available online DDoS hacking tools as shown in Table 2. These tools are often implemented once a large number of IoT devices are infected and can be used to cause large volumetric attacks. We discuss few of them as follows,

- *Golden eye* A multi-threaded attack tool used to launch DDoS attacks using HTTP GET and POST requests. It doesn't provide features such as IP spoofing but works on all popular operating systems such as Windows, Linux and MAC [17, 18].
- *LOIC* An open-source tool used to launch UDP, HTTP and TCP attacks. It is a GUI tool which can deplete user's CPU and memory resources. The Church of Scientology, Visa and Sony were attacked using LOIC tool [19, 20].
- *Slowloris* The attack is launched by opening multiple connections with the target and keeps them open for as long as possible. It uses Perl language and has both a GUI and Command-line interface [17, 21].
- *XOIC* A Graphical Interface tool used to launch TCP, HTTP, UDP and ICMP attacks using manually specified IP address, user-selected port and user-selected

Table 2 Summary of DDoS attack tools, attack generated, methodology, and related studies

Attack tool	Type of attack	Methodology	References
Golden eye	HTTP GET/POST	Persistently tries to keep a socket connection open by exploiting Keep-Alive and No Cache	[17, 18]
LOIC	UDP, HTTP and TCP	Utilizes user's junk traffic to create a botnet attack	[19, 20]
Slowloris	HTTP	Opens multiple connections to the target and keeps them open for as long as possible	[17, 21]
XOIC	TCP, HTTP, UDP and ICMP	Performs an attack using IP address, User-selected port and protocol	[17, 22]
Pyloris	HTTP, FTP, SMTP and Telnet	Utilizes SOCKS proxies and SSL connection	[22, 23]
DDoSim	TCP, HTTP, SMTP and UDP	Uses random IP addresses targeting zombies to attack. Generates HTTP GET requests at random IP addresses	[17, 24]
Tor's Hammer	HTTP POST	HTML POST requests are sent continuously making it difficult to detect	[25–27]

protocol. Implemented using C# and is considered more powerful than the LOIC tool [17, 22].

- *Pyloris* A script-based tool used to launch HTTP, FTP, SMTP and Telnet attacks. Launches an attack using a Slowloris operating system and is used to test server's ability to DDoS attacks [22, 23].
- *DDoSim* Launches attack such as TCP, HTTP, SMTP and UDP. Random IP addresses are used to stimulate several zombies with full TCP connection. A command-line interface implemented using C++ is used to deprive the victim of its resources [17, 24].
- *Tor's Hammer* Python-based attack tool used to launch HTTP attacks. It runs through TOR networks and can exhaust the victim's server resources [25–27].

3 DDoS attack classification

An attack on the IoT device requires authentication details such as username and password. An attacker can infect the device using a brute-force method and take command of it. However, an attack on the cloud layer requires a large volumetric attack to ensure that the target server is partially or completely shut down. An attack from a single computer or a few hundred devices would result in an insignificant impact on the server or network. Modern intrusion detection systems can withstand attacks from few devices. An attacker requires to launch a large volumetric attack like the Mirai attack used more than 600,000 infected IoT devices to attack servers and networks with strong inbuilt security systems. A successful DDoS attack on the cloud layer needs to flood the server or network with requests beyond its capability to handle it. The 2008 GitHub attack with a volume of 1.35 Tbps shut down the server completely for 10 min. Attacks on IoT devices are simpler to execute and require little effort. An attack on the cloud layer requires thousands of infected IoT devices to successfully shutdown a server or network.

The attack classification literature presented in Mirkovic et al.'s research [6] and Specht et al.'s research [28], described the basic DDoS attack structure. We, however, chose to present a detailed and separate classification of types of attacks possible that cause DDoS as shown in Fig. 3. The DDoS attacks classification present the impact DDoS have on the victim's network and bandwidth resources. In such attacks, the attacker aims to consume the victim's limited available resources. The ideal situation would be to drop the malicious packets and allow the legitimate traffic to go through. When packets are dropped, an honest user will cease their attempts to connect, however, an attacker would see this as an opportunity to increase their attempts to intensify the attack. The victim would find their CPU resources depleted and will deny service to all users. Another scenario involves in depletion of network bandwidth wherein an attack does not only affect the victim's resources but also all system-dependent upon the victim's server. Our classification addresses these two types of attacks on bandwidth and network resources. It is possible for an attack to impact both bandwidth and network resources at the same time. A summary of DDoS attack classification is presented in Table 3.

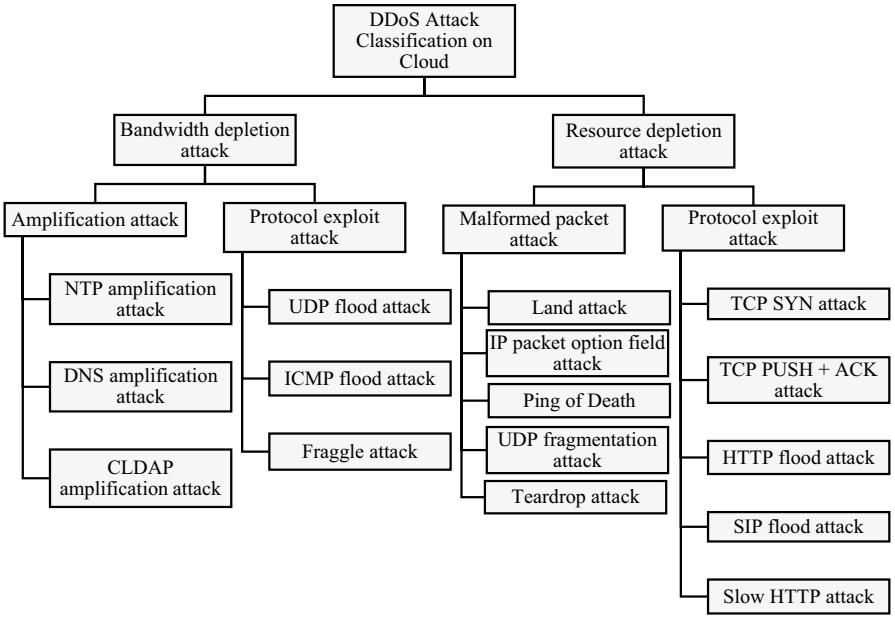

Fig. 3 DDoS attack classification

3.1 Bandwidth depletion attack

The aim of the attack is to consume all the network's bandwidths using an attacking army. The attack can be done by amplifying or broadcasting the attack packets to increase the attack. The legitimate users suffer a denial of service until the attack is detected and mitigated.

Protocol exploit attack The aim of the attack is to consume the resources of the victim by exploiting a feature as a weakness in their system. The attack is done by using a transport layers such as a User Datagram Protocol (UDP) or a network layer protocol such as Internet Control Message Protocol. The attacks using Protocol exploit method are explained below,

- *UDP flood attack* The attacker sends instructions such as the victim's address, the duration of the attack to take place and the method deployed to perform the attack on different infected devices known as the Masters [1, 29, 30]. It is possible that the attacker may first send it to a Master Control program or communicate directly with the Masters. The Master Control program will communicate the attack instructions to the Masters which will cause them to send multiple UDP packets at the random destination port on the victim's computer with a spoofed Internet Protocol (IP) as the source. The victim will, in turn, send the Internet Control Message Protocol (ICMP) packets as the appropriate response to the spoofed address from which it never receives a

Table 3 Summary of DDoS attack, subtype, methodology, impact, and related studies

Topic	Attacks	Subtype	Methodology	Impact	References
Bandwidth depletion	Protocol exploit	UDP flood attack	Floods ports with UDP attack packets	Consumes victim's bandwidth	[1, 29–31]
		ICMP flood attack	Floods victim with ICMP_ECHO_REPLY packets	Consumes victim's bandwidth	[30, 32, 33]
		Fraggle attack	Floods victims with ICMP_ECHO packets	Consumes victim's bandwidth	[34–36]
	Amplification Attack	DNS attack	Amplifies queries to targeted DNS server	Depletes victim's bandwidth	[30, 40, 41]
		NTP attack	Uses MON_GETLIST command to exploit NTP	Depletes victim's bandwidth	[37–39]
		CLDAP attack	Spoofed traffic is sent to the CLDAP protocol server	Floods victim's servers causing system crash	[42–44]
Resource depletion	Protocol exploit attack	TCP SYN attack	Exploits 3-way handshake by flooding the server	Server is unable to process requests, becomes unresponsive	[45–47]
		TCP PUSH + ACK attack	Sends TCP packets with PUSH and ACK set to 1	Consumes memory and CPU resources resulting in system crash	[1, 28, 36, 47]
		HTTP flood attack	Exploits HTTP POST and GET requests	Consumes server's resources	[2, 45–49]
		SIP flood attack	Floods with messages to the REGISTRAR server	Consumes server's resources	[50–52]
		Slow HTP flood attack	Keeps a HTTP connection open for a long time	Consumes server's resources	[53–55]
	Malformed packet attack	Land attack	Sets IP source and destination to the victim's IP	Infinite loop causes the victim's system to crash	[56–58]
		IP packet option field attack	Sets random value to optional fields of IP packets	Overwhelms processing capability of the victim	[59]
		Ping of Death	Sends packet exceeding the maximum value	System will freeze or crash	[60, 61]
		UDP Fragmentation attack	Fraudulent packets are sent larger than the network maximum transmission unit	Fraudulent packets cannot be reassembled and consume server's resources	[44, 62, 63]
		Teardrop attack	Sends fragmented packets	Unable to assemble packets, causes system buffer overflow	[62, 64, 65]

response [31]. Due to receiving a large number of packets and the lack of response, the victim's computer will begin to slow down and ultimately crash.

- *ICMP Flood attack* The attacker sends multiple ICMP echo messages which contain the spoofed source address of the victim's computer to an unprotected broadcast station [30]. The broadcast station will help augment the echo messages (DDoS attack) when attacking the victim's computer with multiple echo reply messages. The more significant number of broadcast stations involved, the higher number of augmented messages the victim will receive. Smurf attack causes the victim's computer to be flooded with high traffic of echo reply messages resulting in the slowing of the victim's computer and eventually rendering it impossible to work with anymore [32, 33].
- *Fraggle attack* Fraggle attacks, also known as amplification attacks, flood the victim's bandwidth with UDP_ECHO_PACKETS [34]. These attacks implement reflectors as their launching mechanism. A reflector that will further broadcast the message to the victim is known as the reflector such as a router or a DNS server. The reflectors will send the attack packet with the spoofed IP address similar to that of the victim [35, 36]. The attacker is hard to detect because of the spoofed IP; however, the reflectors are easily detectable as they are not using spoofed IP addresses.

Amplification attack These attacks generate a significant response as the attacker sends small packet sizes of smaller bytes, but by amplifying them, it transmits a vast number of packets to the victim who consumes all of its bandwidth resources. Two common types of such attacks are the Network Time Protocol (NTP) and the DNS amplification attack.

- *NTP amplification attack* The prime purpose of Network Time Protocol (NTP) is to synchronize the system clock with the server to set the time. The attacker exploits the NTP UDP protocol to send amplified data packets to the victim using a spoofed IP. NTP server attack is initiated by using the "MONLIST" command on the NTP server [37, 38]. Since the MONLIST reply packet is amplified, the MONLIST request packet itself sent by the attacker is much smaller at 64 bytes. The MONLIST command or MON_GETLIST can be issued to the NTP server which replies with a list of 600 systems that have been interacted with [39] which shows that NTP is ideal to be used as an amplification attack.
- *DNS amplification attack* The attacker sends a DNS lookup request to the DNS servers using a spoofed IP address which is the address of the victim. The DNS server responds with the record and sends it to the victim [30]. The attacker will state for any "ANY" request which will carry as much information to the victim. Since the size of the request exceeds the capacity of the response, the DNS attack is an amplification attack. The response being valid replies from the server, it is difficult to determine whether an attacker sends the packets or they are from legitimate users [40, 41].
- *CLDAP amplification attack* Connectionless lightweight directory access protocol amplification attack uses UDP ports to send spoofed packets to the CLDAP server [42, 43]. UDP is often used in DDoS attacks as it does not authenticate

the sender's address. The server sends the reply back to the spoofed address. The response can vary in between 46–55 times the size of the original packet and this it is one of the possible amplification attacks [44].

3.2 Resource depletion attacks

Resource depletion attack aims to deprive the user of their memory, CPU, and socket. This attack is possible by either sending malformed packets such as the Ping of Death attack or by exploiting the weakness in the victim's networks, application or transport layer protocols such as the HTTP Flood.

Protocol Exploit attack The weaknesses in the network layer protocols are exploited by the attacker resulting in the victim exhausting all their CPU and memory resources. Many protocols are exploited such as the Hypertext Transfer Protocol (HTTP), Session Initiation Protocol (SIP) or the Transmission Control Protocol (TCP).

- *TCP SYN attack* This attack involves exploiting weakness in the TCP protocol. The TCP handshake protocol requires a series of acknowledgements from both the parties to execute a successful handshake and establish connection [45]. As part of completing the handshake, the server sends the client a SYN ACK packet. The attacker exploits this protocol by sending spoofed SYN packets to the Server. Upon sending the SYN ACK request, the server keeps waiting for a reply which never arrives [46, 47]. The server stores the connection state in its memory stack until wither a timeout takes place or the connection is successfully established. The attacker floods the server's memory and forces the server to drop SYN requests sent by honest users.
- *TCP PUSH+ACK attack* The PUSH and ACK bits of the header are set to '1' [28, 36]. The botnet of attacking machines will send multiple of these TCP packets resulting in the victim server trying to clear its memory and send an acknowledgement to the client [1, 47]. This forces the server to drop packets sent by honest users.
- *HTTP flood attack* The attacker arranges a serious of infected systems known as bots as part of a botnet. The attacker will use bots to send large volumes of requests which expand the scope of the attack [47]. There are two means to execute an HTTP Flood attack [2]. The HTTP GET attack occurs when the attacker uses the botnet to place GET requests at the same time for files, images, etc. from the targeted server [45]. The server will be kept busy on replying to these requests to all infected systems on the botnet while preventing legitimate requests to be dropped. A GET request is much simpler to execute as any unsuspecting user may partake in the attack by simply visiting a Web site. An attacker may add an inline image in the body of a web-page. Anyone who visits the web-page may unknowingly send a GET request to the target server [48]. The HTTP POST attack involves the attacker using the botnet to submit forms on a Web site [46]. The Web site will be kept busy in this computationally and bandwidth-intensive task. Coupled with the fact that the requests are coming from many infected sys-

tems, the server attaches additional resources [49]. The server is eventually overwhelmed, and a denial of service event occurs. POST attacks are more dangerous for the server as they include parameters which trigger complicated processing and heavy operations on the server. As a result, POST request attacks are far more dangerous than GET request flooding.

- *SIP Flood attack* This attack aims to flood the SIP REGISTRAR or the SIP registration server and consumes all of its resources including CPU, network bandwidth and memory [50, 51]. This attack will overwhelm the server, and legitimate users will be unable to connect and suffer an outage. SIP attacks are made on services offering voice over IP (VOIP). An attack is possible by sending either SIP REQUEST, SIP INVITE, SIP INFO, SIP NOTIFY or SIP RE-INVITE [52].
- *Slow HTTP attack* This type of attack is a slow attack aims to slowly consume all the resources of the victim. Attacks such as Slowlories begin with sending information slowly [53, 54]. It will send a partial HTTP request, and then it will send the header request at intervals ensuring that the sockets remain open. The server then drops all legitimate requests, and a denial of service occurs. It can be mitigated by imposing a limit of transfer rate from a client. Another attack is R.U.D.Y. or R.U. Dead Yet [55]. This attack uses the form submission fields on the Web sites to launch its attack. Using multiple HTTP POST connections, the attacker will submit information in small-sized packets slowly. This behavior forces the server to keep the link open for as long as possible and ultimately exhausting all its connections in the server table. The server will malfunction and deny service to honest user's packets.

Malformed packet attack The basic structure of this attack is to send a deformed packet to the victim which will crash their system. These include Land attack, IP packet option field, Ping of Death and the Teardrop attack.

- *Land attack* This attack is possible by forming an infinite loop. The attacker sets the victim's IP address as the source address of the packet sent [56]. When the victim or system replies to the packet, it essentially replies to itself resulting in an infinite loop. The system crashes eventually [57, 58].
- *IP Packet option field attack* The attacker randomizes the values in the IP packet optional fields. The optional field such as the quality of service is set to 1 which forces the system to spend additional time to analyze it. If an attacker sends a stream of such packets, the system exhausts its processing ability [59].
- *Ping of Death* The attack results in crashing the server by sending ICMP echo requests greater than the IP standard packet size limit. The maximum limit of an IP packet is 65,535 bytes. Larger packet sizes are broken into smaller fragments and then send as multiple packets. The attacker sends multiple oversized packets to the victim who when reassembles the packets, crosses well over the 65,536 bytes limit [60, 61]. Passing the threshold leads to a buffer overflow which results in system crashing. Once the system crashes, it is now more susceptible to other attacks such as the Trojan horse attack.
- *UDP fragmentation attack* The attacker transmits across packets that are fraudulent and are larger than the network's maximum transmission unit. The server is

unable to reassemble the packets as it exceeds the size limit and thus the server's resources are all consumed. This attack finally results in a denial of service to its users [44, 62, 63].

- *Teardrop attack* The attack occurs when the attacker sends fragmented packets to the system [62]. The system due to an error in the TCP/IP fragmentation assembly sends fragmented packets with coinciding offset numbers. The packets upon overlapping upon one another result in crashing the target system [64, 65].

4 DDoS attacks on IoT platform

Merging of smart devices with the internet and their constant communication has been a significant factor for the growth of IoT devices across multiple domains such as logistics, manufacturing industries, smart cities and homes. Industries benefit from data generated from these devices and optimized performance metrics that help in conditions such as to conserve energy, improve agricultural output, monitor medical conditions of patients, improve industrial devices output, etc. IoT devices are open to intrusion attacks and can use further as platforms to launch large volumetric DDoS attacks on other platforms such as the cloud. Most of these devices sold have standard passwords which are shared among all devices. The 2016 Mirai attack used a brute-force method to take control of these vulnerable devices using a list of known passwords which allowed to infect thousands of devices with ease. There are different layers from where DDoS attacks are launched from in the IoT platform as shown in Fig. 4; however, we first discuss the inherent weaknesses found in IoT devices which makes them open for formation of a botnet.

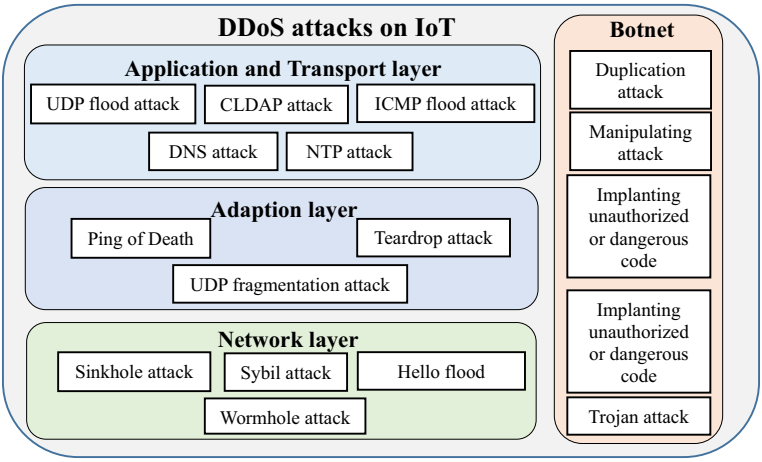

Fig. 4 Layered architecture of DDoS attacks on IoT

4.1 Formation of botnet on IoT devices

The computational constraints and low-power supply of these devices make them easy targets for attackers. Establishing complex security defense mechanisms on them appears to be very difficult with the current technology in place. These devices can not only be tampered physically but remotely which makes especially dangerous. The result of tampering these devices leads to abnormal traffic flow to pass which results in DDoS attacks. Each infected IoT device is called a bot, and the methods mentioned below are deployed to form a botnet of infected IoT devices:

- *Duplication attack* An attacker can attack the entire IoT ecosystem by establishing a new node which copies the information from an honest node and use it to flow abnormal traffic across the IoT ecosystem. Another harm produced from this fake node is that it acts as an unauthorized device and passes on inaccurate information to other connected devices which results in wrong data being generated overall. Incorrect data will severely impede industries that use IoT devices to optimize their services and result in causing damage to organizations.
- *Manipulating attacks* IoT devices do not require onsite human intervention and are accessed remotely which makes them particularly susceptible to manipulation attacks. The attacker can take over the security of the device and use it to infect other devices with the aim to form a collection of infected devices which are also known as a botnet.
- *Implanting unauthorized and dangerous code* The attacker may upon successfully gaining access to an IoT node, inject malicious code to the device/node into its memory and gain access to the entire IoT ecosystem. This code can be used to further spy on the IoT network and seek new devices to infect to launch DDoS attacks.
- *Depletion of device's battery resource* IoT devices contain low-power supply batteries, and as such for efficient usage they deploy a sleep mode to help manage the battery consumption of these devices. The device will power on when it is being required by the ecosystem and will enter in a low-powered state mode to conserve limited resources. An attacker may take advantage of this and upon successfully implanting a malicious code into the IoT device may force it to forego its sleep mode to deplete its battery resources as quickly as possible. This attack is extremely harmful in circumstances where a weather management system is unable to gather information from its sensor to accurately predict weather conditions or a medical system in a hospital which constantly requires an update on the patient's condition and fails to report on any severe situation the patient may be suffering.
- *Trojan attack* This attack requires an attacker to install a triggering device on an IoT device when being manufactured. The attacker can exploit the device and gain information contained in the device itself or seek to gain access to the entire IoT ecosystem via the infected Trojan code.

4.2 DDoS attack classification on IoT

There are many forms of attacks possible using IoT devices on their respective IoT-oriented ecosystems such as the cloud which is an integral part in providing relevant services to users. Cloud is also used to offer computational resources especially in the case of security. The low computational power of IoT devices requires them to rely on other platforms. The various types of possible attacks include Eavesdropping attack, Man in the Middle attacks, etc. However, the most dangerous of them is a DDoS attack. We discuss the layered architecture of DDoS attacks in IoT as shown in Fig. 4.

- *Application and Transport Layer* Internet Engineering Task Force (IETF) introduced the Constrained Application Protocol (CoAP) as the standard protocol at the application layer to facilitate smooth transport and provision of Web applications. CoAP is designed for IoT devices keeping in mind the specifications of low-power and low lossy networks such as simplicity, low overhead, multicast support, and low energy expenditure. User Datagram protocol which is an unreliable and connectionless protocol is used to carry CoAP information from one node to another. The implementation of UDP makes it easily susceptible to DDoS attacks. There are various types of DDoS attacks carried out on the application and transport layer such as the UDP flood attack, ICMP flood attack, DNS attack, NTP attack and CLDAP attack. These attacks have been discussed above in the DDoS attack classification.
- *Adaptation Layer* Adaptation layer is responsible for defining the method to carry IP packets over the link layer. The maximum transmission unit (MTU) of a packet in IPv6 is of 1280 bytes making it unable to fit into an IEEE 802.15.4 network whose size is limited to 127 bytes. Therefore, there is a need for an adaptation layer that enables IPv6 communications in IoT. The layer tools allow optimization of headers which include reassembly, fragmentation, and compression of packets. Packet fragmentation attacks are possible via malformed packets as the layer tools make it difficult on low resource devices to process fragmented attacks. There are various types of attacks on the adaptation layer using malformed packets such as Ping of Death, Teardrop attack and UDP fragmentation. These attacks have been mentioned above in the DDoS attack classification.
- *Network Layer* The attacker attacks the IoT device by exploiting the weaknesses found in this layer. RPL protocol is aimed at IoT devices to reduce energy consumption via traffic-flow methods such as point-to-point, point-to-multipoint, and multipoint to point. However, RPL protocol in IoT devices is vulnerable to DDoS attacks such as Sinkhole attack, Hello flood attack, Wormhole attack, and the Sybil attack. These attacks are discussed below,

  (a) *Sinkhole attack* The attacker compromises the node inside the IoT network to launch a DDoS attack [66]. The attacked node attempts to attract neighboring nodes on the routing metric implemented in the routing protocol. The attacker makes the sinkhole node to route the traffic through it making

it possible for to access the data and perform preferential/selective forwarding [67, 68].

(b) *Hello flood attack* The Hello packet is used by networks to announce that it is within range to another IoT node. An attacker using a computationally powerful device like a desktop or laptop will broadcast information via a powerful transmission that can convince every node that it is within range [69]. The attacker may confuse the network giving the impression that it is the preferred or parent route and have them all transmit their data via it. Since the attacker is not within the range of these nodes, the packets sent will be lost. Packets can also be spoofed via routing updates to allow two nodes to keep sending packets to one another [70, 71].

(c) *Wormhole attack* The attacker performs a replay attack among two non-neighboring nodes in the network. This attack is considered severe in the IoT network [72]. A route is formed between two malicious nodes where they forward packet to one another. The neighboring nodes assume that these packets are close to one another and transfer the packets through them. This attack is considered harmful as the attack has complete control of the two nodes and the data passing through it [73, 74].

(d) *Sybil attack* The attacker will fraudulently steal or promote its identity as multiple distant nodes in a peer-to-peer network. The attacker affects the routing protocols and overall management of the network [75, 76]. The attacker can achieve control over the network to decrease its effectiveness. Network performance and integrity deteriorates under a Sybil attack [77].

5 DDoS attack defense on IoT

Detection of DDoS attack traffic is only one element in defense against DDoS attacks, and while it is an important aspect, prevention and mitigation are two key criteria which help protect the IoT environment. Prevention helps in ensuring the detected DDoS attack does not result in disabling or takeover of the system. Mitigation helps in reducing the severity of the ongoing DDoS attack on the IoT platform [78]. In this section, the different prevention, detection and mitigation mechanisms are discussed. The defense mechanisms are shown in Fig. 5, and their summary is present in Table 4.

5.1 DDoS prevention on IoT

To prevent Denial of service attacks on the healthcare IoT ecosystem, Rajagopalan et al.'s research [79] proposed a mutual authentication scheme between the gateway and the client based on Datagram Transport Layer Security (DTLS) handshake. The gateway will include a list/table of nodes that are preapproved to communicate with other nodes within the medical IoT environment. The gateway manages an active session column to ensure if the node is corresponding with another node, the same node is denied communication with other nodes in the IoT ecosystem. The gateway

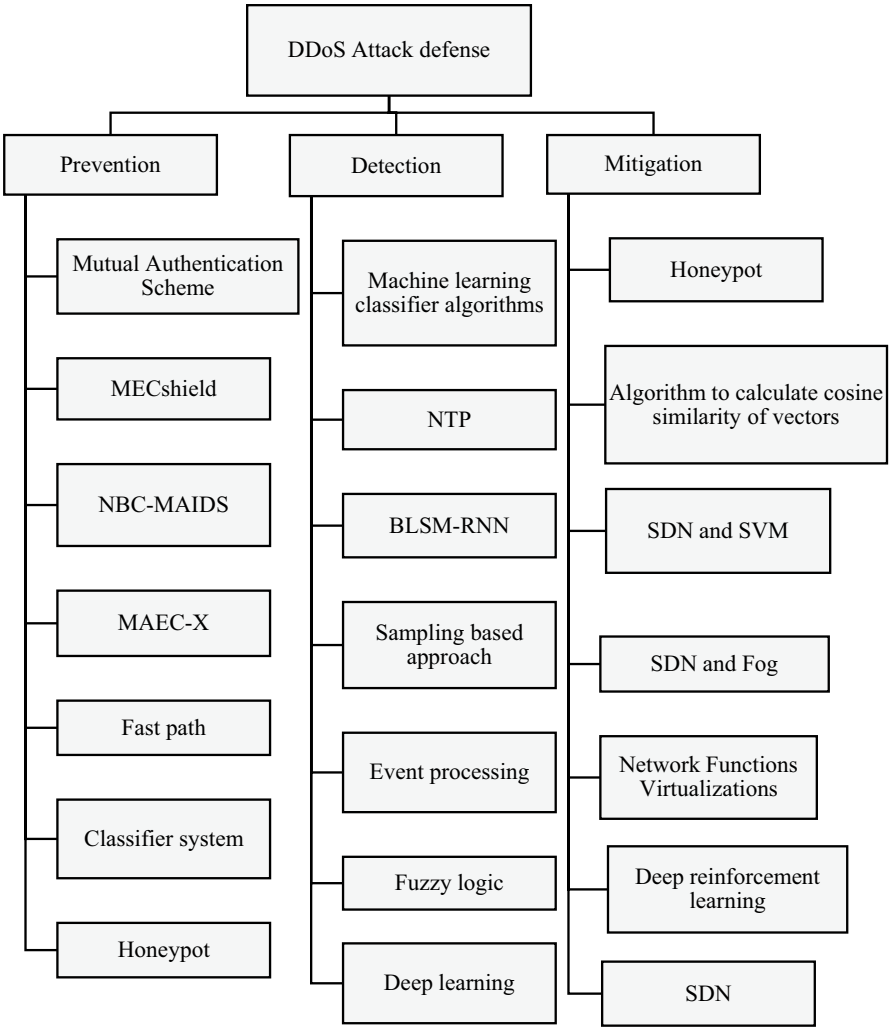

Fig. 5 DDoS attack defense on IoT

will increase the count each time the node requested a connection with other nodes, and if it crosses a limit, it will finally block it. The proposed method prevents both DDoS and Replay attacks. The prevention method is theoretically examined, and no practical implementation is mentioned. The proposed model strengthens the disadvantage found in related works of lacking computational power in IoT layer by ensuring all computational work is done on the Smart e-Health gateway. This takes the load off from the medical sensors which cannot handle heavy processing.

Dao et al.'s research [80] proposed a mobile edge computing-based MECshield mechanism to safeguard Heterogenous IoT environment. This mechanism uses mobile edge computing to deploy several smart filters at the edge of the network. A

Table 4 Summary of DDoS attack defense, technique, platform, description, and related studies

Topic	Technique	Platform	Description	References
Prevention	Mutual authentication scheme	IoT	Gateway manages an active session column to ensure if the IoT node is corresponding with another node, the same node is denied communication with other nodes in the IoT ecosystem. Strengthens the disadvantage of lacking computational power found in the IoT devices	[79]
	MECshield	IoT	Deploys several smart filters at the edge of the network. A central controller monitors the communication between intelligent filters and shares identifying features of the attacking traffic to smart filters. It improves the detection rate and accuracy over other distributed and centralized self-organizing map (SOM) filters by separately training the SOM maps in the MECshield agents by separate local IoT traffic	[80]
	NBC-MAIDS	IoT	IDS using Naïve Bayes algorithm is deployed on nodes using agents which are deployed to audit the network traffic, classify if the incoming packet is an attack packet or not. NBC-MAIDS outperformed existing BRTM and DRPT models in attack detection rate	[81]
	Multi-access edge computing	IoT	Deploys filters at source and destination of attack packets to prevent flooding attacks and mark connections as suspicious emitting from sensors to prevent HTTP GET/POST, UDP, ICMP, SYN attacks, etc. MAEC-X improves MAEC in 5G networks against DDoS attacks by utilizing the computational power present at the edge of the network to generate local rules	[82]
	Fast path	IoT	Edge functions at edge nodes will transmit data profile sketch to a web service on cloud to assess if the packet is malicious or legitimate using a fast path. Improves upon existing systems by implementing a priority packet scheduling policy which sends smaller packets containing small keywords for self-authentication. They travel at a faster rate than other normal packets	[83]
	Classifier system	Cloud	Detection phase collects traffic and compares source of packet with prior DDoS attack list. Classifier as part of prevention phase will analyze the packet if it is malicious or not. The proposed CS_DDoS improves the existing system by offering a faster and accurate detection rate while maintaining the network scalability	[84]
	Honeypot	Cloud	Uses a virtual system to lure attackers. Attack traffic is transferred to another virtual device. The Honeypot-based network addresses the disadvantage of existing system where the network fails to respond when assaulted with several packets. It shifts all attack packets to a virtual computer and blocks attacking IP addresses	[85]

Table 4 (continued)

Topic	Technique	Platform	Description	References
Detection	Machine learning classifier algorithms	IoT	Monitors network traffic flow using a computationally low-cost algorithm. IoT devices suffer from low computational power. The proposed detection mechanism implements a packet-level detection policy with low feature set to reduce computational overhead	[86]
	NTP	IoT	NTP client communicates the NTP server to assess the difference in the device clock time from the primary server clock. Existing solutions are expensive and require periodic maintenance with technical knowledge. The NTP-based solution requires no periodic maintenance and any expensive equipment for implementation	[38]
	BLSM-RNN	IoT	Analyses the packet flow and implements text identification in features. The proposed solution offers better attack vector detection accuracy compared to other models with different selection of features. It performs detection at the packet level focused on the text recognition in the selected features	[87]
	Sampling-based approach	IoT	Place sample collector with network policy at edge gateway to block malicious traffic. Existing sampling-based methods are unable to detect suspicious traffic in small volume. The proposed method sets the rule with sampling rate at the edge gateway which enables it to detect small volume of attack traffic	[88]
	Event processing	Edge	Implements Packet analyzer module to analyze traffic properties and Attack detection module to determine type of DDoS attack. The proposed CEP mechanism using event processing consumes less CPU and RAM resources compared to SNORT and BRO existing solutions. This enables it to be implemented on computationally low powered IoT devices	[89]
	Fuzzy logic	Cloud	Fuzzy Inference system will report any high traffic volume to cloud. Existing defense systems are expensive and difficult to establish. The proposed implementation of the Fuzzy logic system is cost-effective, reliable and simpler to implement	[90]
	Deep learning	IoT	Uses Recurrent Neural Network to identify attack packets. Existing machine learning-based IDS are limited by shallow representation models and provide an error rate of 7.57%. The proposed model based on RNN offers a lower error rate of 2.1%	[91]

Table 4 (continued)

Topic	Technique	Platform	Description	References
Mitigation	Honeypot	IoT	Routes all traffic toward the honeypot to analyze and gather information about attack. Existing honeypot-based solutions are theoretical-based and cannot be proven to be effective in real time. The proposed system is practically implemented on the IoT environment and results show a 55–60% increased efficiency in attack detection	[92]
	Algorithm to calculate cosine similarity of vectors	IoT	Determines the cosine similarity of the vectors of each incoming packet rate at the SD-IoT. Existing solutions use DPCC algorithm which is unable to address large amounts of traffic and fails to detect DDoS attacks on the SD-IoT switches. The proposed algorithm uses the threshold value to detect a DDoS attack and results show faster detection and mitigation of DDoS attacks on IoT devices	[93]
	SDN and SVM	IoT	SVM is implemented at the SDN controller to identify the malicious packet flow. Controller prevents the attacking node from communicating with other IoT nodes. The proposed framework addresses the disadvantage of existing frameworks in detecting DDoS attacks at the higher level of the network and focus on device security. The framework detects attack traffic in the entire network by focusing on packets flowing from the edge of the network	[94]
	ECESID	IoT	Scans to identify attack packets and implements rules in switches to blacklist and drop attack packets. Existing solutions do not focus on Mirai attacks which are common on IoT devices. The proposed solution is simulated in a real-time environment which shows it is an effective defensive approach against Mirai-based attacks	[95]
	Network functions virtualization	Edge and Cloud	Traffic screener sends malicious packets to NFV which will auto scale based on the size of the attack traffic. Existing ISP-based solutions do not defend against all forms of DDoS attacks and violate a user's privacy. The proposed scheme is implemented at the organization's datacenters which reduces latency and maintains user's privacy	[96]
	Deep reinforcement learning	SDN	Uses a reward function on the Mitigation agent to protect victim's server. The proposed solution's algorithm is placed at the SDN controller compared to existing solutions which execute the algorithm at the switches. This enables to view the entire network's traffic flow information from the SDN controller	[97]
	SDN	IoT	Controller access if the packet from the flowtable is malicious and if so, it will drop it. Existing sampling-based approaches are a lossy process and as such provide inaccurate profiling of network traffic. The proposed model collects traffic flow statistics from SDN-enabled switches to achieve higher detection accuracy	[98]

central controller monitors the communication between intelligent filters and shares identifying features of the attacking traffic to smart filters. The dilemma of the formation of a bottleneck in traffic flow is resolved by placing the filters at various mobile edge points. Based on the attack method deployed, the smart filter is trained using its local attack traffic features obtained from the different IoT devices where the smart filters are applied. The experimental setup for practical implementation used three different datasets, the CAIDA-attack traffic, NSL-KDD and DARPA datasets. Attack traffic is based on three types, sensor traffic, monitor traffic and alarm traffic. The SOM filters are trained using the datasets. At the destination site, protocol, port number and flow number are applied for SOM training at the MECShield agent. At the source site, protocol, port number, packet/flow, transmission contiguity and flow number are applied to SOM training at the MECShield. Results show better detection accuracy and rate due to optimized SOM filters. The proposed MECshield framework improves the detection rate and accuracy over other distributed and centralized self-organizing map (SOM) filters by separately training the SOM maps in the MECshield agents by separate local IoT traffic.

Mehmood et al.'s research [81] proposed implementing a machine learning-based classification algorithm to prevent DDoS attacks on IoT platform. The mechanism deployed is termed as Naive Bayes Classifier-Multi-agent Intrusion Detection System (NBC-MAIDS). The Intrusion Detection System implements a Naive Bayes algorithm, and it is used across the nodes in the IoT ecosystem in the form of multi-agents. These agents monitor the nature of the traffic and the management of nodes. Multi-agents are deployed to audit the network traffic, classify if the incoming packet is an attack packet or not, drop the attack packet and manage the traffic database, and communicate with other agents to monitor the detection results and inquire about further information. To implement the proposed algorithm, NS 2.35 was used to simulate and evaluate the results. Source and destination nodes were randomly selected after being infected. This helps to determine the threat effectiveness. NBC_MAIDS was compared with the performance with the Bio-inspired Reputation and Trust Model (BRTM) WSN and Distributed Reputation-based Beacon Trust System (DRBT) model. Results showed that NBC-MAIDS outperformed other models in attack detection rate. The NBC-MAIDS strengthens other intrusion detection systems in detection accuracy using machine learning-based solutions by implementing multiple agents for accurate DDoS attack detection.

Dao et al.'s research [82] proposed a prevention method via a multi-access level edge computing. The architecture involves a controller located at the cloud while the MAEC-X clients which include the attack detection and prevention module are present at the edge nodes and the data center. The clients will monitor the traffic in real time and when a DDoS attack occurs, they will report to the controller the IP address, section port, attack method used, the source of the attack and security level of the attack. The controller will warn the clients of the security status and assign prevention policies for effective prevention. To counter attacks such as UDP and ICMP flood attacks, the MAEC-X deploys filters at both the source and destination of the attack packets. To prevent attacks such as Ping of Death and SYN Flood, the clients will mark packets as suspicious which exist from sensory nodes or thermal sensors requesting connections within short intervals. HTTP GET/POST attacks

can be prevented by having the edge nodes report abnormal traffic to the controller which will block the suspicious packets. The proposed prevention method is theoretically examined, and no practical implementation is mentioned. To strengthen multi-access edge computing (MAEC) in 5G networks against DDoS attacks, MAEC-X utilizes the computational power present at the edge of the network to generate local rules for controlling malicious traffic generated from terminals.

To secure IoT against DDoS attacks, Bhardwaj et al.'s research [83] proposed a proactive defensive measure while making edge the first line of defense against DDoS attacks. The ShadowNet Web Service will be stored in the cloud with ShadowNet edge functions running at the edge nodes. These edge functions will transmit a sketch of the data profiles collected from the incoming packet traffic known as Shadow packets to the ShadowNet Web service via a path known as ShadowNet Fast path. The Webs service will assess if the incoming traffic is a legitimate packet or a DDoS attack. It is, however, not without its demerits. Firstly, this approach foregoes accuracy over speed because of which there is no way of identifying if it's a genuine attack or a Flash crowd. Secondly, the concept is based on its application on a singular network, i.e., it is insufficient to handle data from different geographical positions. If it were to be deployed, the Fast Path utilized to transfer data to the ShadowNet Web service may not be as quick as required. They tested it against UDP and HTTP GET flooding attacks. To evaluate the proposed system, GENI platform was used to create a testbed where there are four virtual machines, the attacker, ShadowNet edge function, ShadowNet service and the victim. HTTP GET via sensors and UDP flooding attacks via cameras were simulated. The attack pattern was intended to follow the Mirai attack pattern of short bursts and also experimented attack sessions for extended periods of time. Based on the experimental analysis, the proposed system detects UDP flood attacks 10.6 times faster than other existing models. The proposed model addresses the disadvantage of other existing systems in detecting malicious traffic in real time. It implements a priority packet scheduling policy which sends smaller packets containing small keywords for self-authentication which travel at a faster rate than other normal packets.

The above discussed methods provide solutions to prevent DDoS attacks on IoT networks. The mechanisms mentioned below prevent DDoS attacks and can be applied successfully on the IoT environment.

According to Sahi et al.'s research [84] proposed CS-DDoS system, a new detection and prevention method to prevent a DDoS TCP attack on a cloud environment. The Least Squares Support Vector Machine (LS-SVM) classifier system is used to analyze the incoming packets and judge if the cloud is under attack or normal traffic is flowing. The proposed system is built upon two foundations, Detection phase, and Prevention phase. During the first phase, it will collect all incoming traffic and check their sources with a prior formed blacklist if their sources are known attackers of the cloud. If not blacklisted, it will send the packet to the classifier to measure whether the packet is part of an attack or not. If it results in normal, it will be allowed to connect with the cloud and if not, it will be sent to the second phase, the prevention phase. During this phase, the administrator is informed of the attack, the packet is added into the blacklist and finally dropped. Four different classification algorithms were used to experiment the CS_DDoS system using LS-SVM, Naïve

Bayes, k-nearest and multi-layer perceptron algorithms. LS-SVM demonstrated an accuracy rate of 94% from multiple source and 97% from single-source TCP flood attacks. Existing solutions offer low accuracy in DDoS detection and suffer from scalability issues as the network grows. The proposed CS_DDoS offers a faster and accurate detection while maintaining its scalability.

Manoja et al.'s research [85] proposed implementing a Honeypot tool to prevent DDoS attacks on the Cloud environment. The intention is to lure the attacker into a false notion that they have successfully attacked the Cloud server. Honeypot acts as a shield for the cloud by collecting data regarding the attacker and the attack type. Based on the attack type, security measures can be implemented on the cloud to safeguard it. The honeypot will now block access to all the attacking IP addresses collected by it. All attack traffic with irregular patterns from victimized devices, i.e., attack packets with spoofed IP addresses are transferred to a different virtual computing device. The proposed Honeypot tool is theoretically examined, and no practical implementation is mentioned. The proposed study addresses the disadvantage of existing system where the network fails to respond when assaulted with several packets. Honeypot-based network strengthens traditional networks by shifting all attack packets to a virtual computer and blocks all attack IP addresses.

5.2 DDoS detection on IoT

Doshi et al.'s research [86] proposed classifier-based machine learning algorithm to monitor network traffic to prevent the DDoS attack on the IoT ecosystem. Their proposed idea was able to detect DDoS attack traffic on local IoT nodes implementing using a computationally low-cost algorithm with a 0.99 accuracy. For simulation, different algorithm models are tested and compared including K-nearest algorithm, support vector machine, decision tree, Random forest and neural network. The choice of features selected was a limited feature set to avoid high computational overhead. IoT devices have the disadvantage of low computational power and so they cannot run heavy machine learning solutions. The proposed detection mechanism implements a packet-level detection policy with low feature set to reduce computational overhead. The detection algorithm was able to perform detection in real time on IoT devices and the accuracy was recorded at 0.99.

Kawamura et al.'s research [38] proposed implementing an NTP-based DDoS attack detection mechanism for the IoT environment. During a DDoS attack, there is an overall delay in the system processing and as such the device clock deviates from the server clock. The NTP client will communicate the NTP server to assess the difference in the device clock time from the primary server clock. Based on the clock difference, DDoS attack events can be detected if the clock is unable to synchronize continuously with the server. To simulate the mechanism, background processes were made to run on the node. Chrony is used as the time synchronization module. The interval between the IoT device and the NTP server is set at 2 s. Polling interval between the NTP server and the local NTP server is set at 1024 s. Apache Bench is used to generate DDoS attacks. Results show that the system is ready for real-time event detection on IoT devices as it demonstrates a precision value of 0.92 with

a recall value of 1.0. Existing DDoS detection systems use Honeypot, firewall and packet capture solutions which are expensive and require periodic maintenance with high technical knowledge. The proposed model supplements the existing solutions with the NTP-based solution which requires no periodic maintenance and expensive equipment for implementation.

Detection of botnet formation on the IoT devices and network, McDermott et al.'s research [87] proposed implementing a machine learning-based solution using Bidirectional Long Short-term Memory Recurrent Neural Network. Detection of DDoS attack is done by analyzing the packet flow. Compared to other flow detection methods, the proposed method focused on text identification within features. Prediction of attack vectors is made using Word embedding for text identification. The proposed methodology is simulated using UDP flooding and DNS attacks which displayed 98% accuracy in attack detection. The attack type was split between train and validate. Each model was trained over 20 iterations. Existing methods of detecting botnets based on signature or flow-based anomaly intrusion detection are unable to prevent IoT-based botnet attacks. The proposed solution offers better attack vector detection accuracy by performing at the packet level focused on the text recognition in the features selected.

The above discussed methods provide solutions in detection of DDoS attacks on IoT networks. The mechanisms mentioned below also detect DDoS attacks and can be applied successfully on the IoT environment as well.

Nguyen et al.'s research [88] proposed an approach to detect DDoS traffic at the edge gateway in a Software-Defined Network (SDN). There are multiple sample collectors placed at different switches. The incoming traffic is forwarded to the Intrusion Detection System for analysis. Assuming an attack scenario where there are a large number of packets, but their volume is small, they suggested placing the sample collector at the Edge gateway to help block any suspicious traffic from heading to its destination. The collector collects samples of traffic heading to each domain which makes it possible to detect the small volume of traffic. The controller will apply a new network policy wherein it will send new policy instructions to the collectors to block any suspicious traffic from heading toward its destination. The detection mechanism was tested on UDP, ICMP and TCP attacks. To evaluate the attack, a testbed was setup consisting of a network with 6 switches connected using OpenSwitch. The user devices were connected at the edge gateway. The IDS with the rule and the sampling rate is calculated at the edge gateway. The sample traffic is forwarded to the IDS by the network for analysis and the result of the detection is updated in the database. Other sampling-based methods are unable to detect suspicious traffic in small volume; however, proposed method sets the rule with sampling rate at the edge gateway which enables it to detect small volume of attack traffic.

Cardoso et al.'s research [89] suggested a Complex Event Processing Intrusion Detection System (CEPIDS) which would allow a real-time analysis of traffic. The Intrusion Detection System (IDS) has been placed at the network edge. CEPIDS via an Event filter will monitor and collect network traffic. The Event Processor which includes two modules, Packet analyzer and Attack Detection, will analyze the traffic packets based on their properties and determine which attack the network is under. The CEP will send rules to the Action Engine which will further send an alert on

the malicious activity taking place and block access to the relevant services. The proposed model was tested on a Raspberry Pi device to ensure that it performs well on devices with low computational power such as IoT devices. The attack simulation was conducted on a Raspberry Pi 3 Model B device where the CPU usage, the Ram usage and the packet arrival rate and the count of attacks are determined. Existing open-sourced IDS such as SNORT and BRO consume high RAM and CPU resources whereas the proposed CEP mechanism using event processing consumes less resources. This enables it to be implemented on IoT devices.

Mondal et al.'s research [90] proposed using Fuzzy Logic to secure cloud environments from DDoS attacks. It is implemented on the Cloud layer where all incoming traffic is first filtered through the Fuzzy system and then allowed entry to the cloud. Using the IF–THEN logic, the Fuzzy Inference system will report any high amount of data traffic and report to the cloud. Should the traffic seem very large for the cloud, it may activate the defense systems or discard the data entirely. The Fuzzy IF–THEN rule is simulated using the MATLAB software. An increase in packet arrival rate, the attack status becomes high whereas an increase in entropy rate, the attack status decreases. Other existing defense systems are expensive and difficult to establish. The proposed implementation of the Fuzzy logic system is cost-effective, reliable and simpler to implement.

Yuan et al.'s research [91] proposed using a DeepDefense method implementing a Deep Learning to help improve DDoS detection. The Recurrent Neural Network (RNN) was used to identify attack packets. They formulated the DDoS detection as a sequence classification problem and transformed the packet-based DDoS detection to the window-based detection. In simulation, RNN noticeably outperformed other models in identifying attacking packets as it was able to learn features better. Tracing the history from previous packets using RNN; Long Short-term memory Neural Network (LSTM) and Gated Recurrent Unit Neural Network (GRU) were used to eliminate all scaling issues. RNN showed better performance in generalization than Random Forest. Existing machine learning method-based IDS are limited by the shallow representation models and provide an error rate of 7.57%, whereas the proposed model based on RNN offers a lower error rate of 2.1%. RNN benefits from its design which enables it to learn traffic patterns from sequences of network traffic.

5.3 DDoS mitigation on IoT

To successfully prevent DDoS attacks on IoT devices, Anirudh et al.'s research [92] proposed using a honeypot to secure the IoT ecosystem where all traffic goes through the IDS. If the traffic is malicious, it will route all traffic toward the honeypot to analyze and gather information about the nature of the attack. Information about the attacker is collected such as the IP address and MAC address, and the data are logged in a database. If the attacker repeatedly sends malicious packets matching with the same IP address, the IDS will block the attacker from any further communication with the network. If the traffic does not match with the log, it will be allowed to pass through. To practically implement the proposed solution, a socket server client model is used with a central server and nodes connected to it which emulates as

an IoT model. The simulation consists of bots introduced in 10 steps ranging from 0 to 100. Existing honeypot-based DDoS mitigation solutions are theoretically based and cannot be proven to be effective. The proposed system is practically implemented on the IoT environment and results show a 55–60% increased efficiency in attack detection over systems without honeypot implemented.

Yin et al.'s research [93] proposed an algorithm which aims to detect whether the IoT platform is under a DDoS attack, identify the intruder and terminate the attack at its source. The algorithm ascertains the cosine similarity of the vectors of each incoming packet rate at the Software-Defined IoT (SD-IoT) switches. When the incoming malicious packet has no identical match with the flow table, it will forward the packet to the controller. The controller will determine from which port the packet was sent. The algorithm is applied to determine the infected IoT node from which the attacking packet was forwarded. The algorithm serves to instantly discover which IoT node is being used to launch DDoS attacks and can be mitigated promptly. To test the proposed algorithm, the algorithm calculates the cosine vectors of each packet at the SD-IoT switches. Based on the value of the cosine similarity determined, the DDoS attack is detected. Existing solutions use DPCC algorithm which is unable to address large amounts of traffic and as such fails to detect DDoS attacks on the SD-IoT switches. The proposed algorithm uses the threshold value to detect a DDoS attack and results show faster detection and mitigation of DDoS attacks on IoT devices.

Identifying the sub-par level of inbuilt security of IoT devices, Bhunia et al.'s research [94] proposed a combined SDN and machine learning-based detection and mitigation mechanism termed as SoftThings. SDN is applied to continuously monitor the packet traffic flow and report of any anomalies in the network using the SDN controller. A classification-based Support Vector Machine (SVM) is implemented at the controller to identify the malicious packet flow. Once it detects the source of the incoming malicious traffic from the IoT node, it is partially or fully blocked. The controller is updated with the newly identified source of the attack and prevents the attacking node from communicating with other IoT nodes. The proposed method is tested with TCP flooding attacks. The proposed scheme uses Mininet to simulate IoT devices and is tested using different attack scenarios. In the first scenario, a single IoT device is accessible only via a SDN-enabled switch is under an attack. In the second scenario, two IoT devices are connected to the switch and one of the devices is compromised. In the third scenario, three devices are present where two of them are part of the botnet. A non-linear SVM machine learning algorithm using kernel trick gave accurate detection results. The proposed framework addresses the problem in existing frameworks in detecting DDoS attacks at the higher level of the network and focus on device security. The framework focuses on early detection of attacks on IoT layer by detecting attack traffic in the network at the edge of the network.

According to Ozcelik et al.'s research [95], to combat IoT DDoS traffic, they used Mirai botnet attack as the case study and suggested using Edge-oriented detection scheme while using Software-Defined Networking (SDN) and Fog approaches. The security function was placed on the endpoints of the edge node to be close to the IoT devices in a fog computing environment. There are two phases in DDoS

mitigation used here with first involving a scanning phase to first identify the attack packets and second, implementation of rules to blacklist and drop all flows from the malicious host. A typical DDoS attack involves the malicious host attempting to connect with as many other hosts as possible to infect them. In the proposed architecture, ECESID utilizes four modules to detect the attacking packet. It will determine whether the TCP packet is an attack packet. It will now check the failure count of the flagged packet and update the TCP-SYN queues based on the timeout and Reset occurrences. If the unsuccessful attempts exceed the limit, the host is added to the blocked list. SDN makes it possible to easily dictate the packet flow rules and assign rules as and when necessary. The ECESID scheme is implemented using the Mininet WiFi to emulate IoT nodes and Floodlight is used to implement the security model. Benign traffic was generated using iperf and attack traffic was demonstrated the Mirai botnet firmware. The proposed solutions focus is on Mirai-based attacks and their mitigation which are commonly used to launch DDoS attacks. The author has provided simulated results which show an effective defensive approach against Mirai-based attacks. Existing solutions focus on other botnet attacks which makes it difficult to know with certainty if they can provide adequate defense from Mirai attacks.

The above discussed methods provide solutions to successfully mitigate DDoS attacks on IoT networks. The mechanisms mentioned below also help in mitigation DDoS attacks and can be applied successfully on the IoT environment as well.

Alharbi et al.'s research [96] proposed using Network Functions Virtualization (NFV) and Edge computing to mitigate DDoS attacks with the aim to protect from all DDoS attacks. The proposed model is based on two processes, Traffic Screening on edge networks and taking advantage of Autonomy and Scalability by implementing NFV. The Traffic Screener module will inspect the flood of packets based on traffic and packet features. If the packets are malicious, it will perform Virtual Security Functions (VSF) based on the type of attack, i.e., network layer or application layer. If the VSF demands more resources to handle the attacking traffic, more resources will be allocated to it via the Resource Allocation module. The Traffic Screener module interacts with the Resource Allocation module to manage scalability and demands of the VSFs. The proposed scheme is theoretically examined, and no practical implementation is mentioned. Existing solutions do not defend against all forms of DDoS attacks and some violate a user's privacy as they are based on ISP. The proposed scheme is implemented at the organization's datacenters which reduces latency but also maintains a user's privacy.

Liu et al. 's research [97] proposed a mitigation framework based on deep reinforcement learning which can learn the DDoS attack situation it is facing and defend against it in real time. Their method can increase or slow down the attack traffic flow. Using an SDN controller where the network traffic is collected, the deep reinforcement learning method can target the attacking traffic and leave the legitimate traffic to have target access. The framework can defend against flooding attacks such as ICMP, UDP and TCP SYN. The framework architecture includes two modules, the Information Collection module, and the DDoS mitigation module. The first module is implemented to collect the network traffic so that the reinforcement learning may take appropriate actions and the second module contains two parts, Mitigation

server, and an agent. The mitigation server functions on the SDN controller while the agent is equipped with the deep reinforcement learning algorithm and these two work together to mitigate the DDoS attack in real time. The mitigation agent uses a reward function by which it is trained to protect the victim's server from being overloaded and allows legitimate traffic to pass through successfully. Should the agent fail, it receives a negative reward of -1. The framework shows high performance in mitigating malicious packets as it uses OpenFlow protocol to have a global view of the entire network and collects information on a regular interval. The model's mitigation solution is implemented in TensorFlow and deployed in a different virtual host. The deep reinforcement agent is trained using benign and attack traffic. The model is evaluated on five different attack patterns such as constant-rate, increasing, pulse and group attacks to launch DDoS attacks on the victim's server. The proposed solution's algorithm is placed at the SDN controller compared to existing solutions which execute the algorithm at the switches. This enables to view the entire network's traffic flow information from the SDN controller.

Ahmed et al.'s research [98] discussed using Software-Defined Networking (SDN) architecture to detect and mitigate DDoS attacks on IoT systems. To achieve accuracy in DDoS detection and mitigation, they used SDN infrastructure to overcome the sampling-based anomaly detection constraints to collect traffic flow statistics maintained at switches. Open flow protocol allows switches to be managed by an external controller. The controller will assess if the received packet from the flow table is malicious and in which case it will drop the packet at the immediate routers. The proposed mitigation solution is a theoretically presented and hence no practical implementation is demonstrated. Existing sampling-based approaches are a lossy process and as such provide inaccurate profiling of network traffic. **T**he proposed model collects traffic flow statistics from SDN-enabled switches to achieve higher detection accuracy.

6 Discussion and suggestions

6.1 Discussion

There is an extensive study done while preparing this survey to propose the types of DDoS attacks possible and their defense mechanisms as shown in Table 5. The paper focuses on the following key areas:

- Motivation to why attackers prefer IoT devices to commit DDoS attacks on servers and networks.
- The tools required to capture non-legacy IoT devices to form botnets. These compromised devices are used to launch large volumetric attacks using readily available tools to commit DDoS attacks.
- Growing trend for attackers to launch multi-vector DDoS attack combinations deployed using IoT devices.
- The different kinds of DDoS attacks that take place based on bandwidth and resource depletion objectives.

Table 5 Summary of DDoS attack by type and their respective techniques to prevent, detect and mitigate them

Solution	Prevention							Detection						
Attack	Mutual authentication	MECshield	NBC-MAIDS	Multi-access	Fast path	Classifiers system	Honeypot	Machine learning classifier algorithms	NTP	BLSM-RNN	Sampling-based	Event processing	Fuzzy logic	Deep learning
UDP attack		✓		✓			✓	✓		✓	✓	✓	✓	✓
ICMP attack			✓	✓			✓	✓			✓	✓	✓	✓
Fraggle				✓			✓						✓	
DNS attack		✓		✓			✓			✓			✓	✓
NTP attack		✓		✓			✓		✓				✓	✓
CLDAP attack							✓							
TCP SYN attack				✓		✓	✓	✓		✓	✓	✓		
TCP PUSH + ACK attack						✓	✓			✓	✓	✓		
HTTP flood attack				✓	✓		✓	✓					✓	✓
SIP flood attack							✓						✓	
Slow HTTP flood attack							✓							
Land attack							✓							
IP packet option field attack							✓							
Ping of death				✓			✓							
UDP fragmentation attack							✓							
Teardrop attack							✓							

Table 5 (continued)

Solution	Prevention							Detection						
Attack	Mutual authentication	MECshield	NBC-MAIDS	Multi-access	Fast path	Classifiers system	Honeypot	Machine learning classifier algorithms	NTP	BLSM-RNN	Sampling-based	Event processing	Fuzzy logic	Deep learning
Sinkhole attack							✓							
Wormhole attack							✓							
Hello flood attack			✓				✓							
Sybil attack	✓						✓							

Solution	Mitigation						
Attack	Honeypot	Algorithm to calculate cosine similarity of vectors	SDN and SVM	SDN and Fog	NPV	Deep reinforcement learning	SDN
UDP attack	✓	✓	✓	✓	✓	✓	✓
ICMP attack	✓	✓	✓	✓	✓	✓	✓
Fraggle	✓			✓	✓		✓
DNS attack	✓			✓	✓		✓
NTP attack	✓			✓	✓		✓
CLDAP attack	✓						
TCP SYN attack	✓		✓	✓	✓	✓	✓
TCP PUSH + ACK attack	✓			✓	✓		✓
HTTP flood attack	✓		✓	✓	✓		✓
SIP flood attack	✓		✓		✓		

Table 5 (continued)

Solution	Mitigation						
Attack	Honeypot	Algorithm to calculate cosine similarity of vectors	SDN and SVM	SDN and Fog	NPV	Deep reinforcement learning	SDN
Slow HTTP flood attack	✓		✓		✓		
Land attack	✓				✓		✓
IP packet option field attack	✓				✓		✓
Ping of Death	✓				✓		✓
UDP fragmentation attack	✓						
Teardrop attack	✓				✓		✓
Sinkhole attack	✓						
Wormhole attack	✓						
Hello flood attack	✓		✓				
Sybil attack	✓						

- Formation of botnet using IoT devices to use to commit DDoS attacks on other platforms.
- DDoS attacks on IoT devices.
- The defense mechanisms in place based on three criteria, prevention, detection and mitigation against DDoS attacks.

We presented multiple defense solutions against DDoS attacks on IoT. Studying Table 5, we learn that mechanisms such as Mutual authentication scheme, MECshield, Fast Path, Event processing among others, utilize the advantage that Edge and Fog layer provides in monitoring the network level traffic. Some of the mechanisms such as ECESID, machine learning algorithms such as SVM and deep reinforcement learning take advantage of SDN. They implement SDN flowtable which is used to maintain a list of identifying features of the detected malicious packet data. They also implement the SDN controller which monitors the traffic to help detect and prevent DDoS attacks. The honeypot-based tool is implemented at the network level to prevent all types of DDoS attacks. SDN-based solutions which were evaluated and simulated were found to stop all DDoS attacks except SIP flood and slow HTTP attacks.

The most common types of DDoS attacks witnessed are UDP flood attacks. During the first and second quarter of 2018, UDP flood attack has consistently been responsible for half of all DDoS attacks [99, 100]. To summarize the observations found in Table 5, we determine that to prevent UDP flooding-based DDoS attacks, MECshield, multi-access edge computing and fast path are suggested as appropriate prevention mechanisms. Other solutions offer good detection accuracy such as machine learning classifier algorithms, BLSM-RNN, sampling-based approach, fuzzy logic, event processing and deep learning. To mitigate the UDP flooding-based DDoS attacks occurring on devices, servers and networks, Honeypot-based network security, algorithms which study the cosine similarity of attack vectors, Network Function Virtualization and Deep Reinforcement Learning algorithm are effective solutions.

DDoS attacks have increased in volumetric size from 100 Gbps in 2013 to 1.35 Tbps in 2017. A lot of research work is done to detect, prevent and mitigate such attacks and yet it continues to show an upward trend. One of the prime reasons for such explosive growth are insecure non-legacy IoT devices [101–103]. Many of these devices do not have the necessary security protocols in place to defend themselves from malicious intrusion. Thus, it is vital to study and present state-of-the-art defense mechanisms to thwart such attacks. We have presented multiple mechanisms that can be implemented to thwart DDoS attacks on IoT platforms. Many of these mechanisms are tested on IoT environments. However, there are some mechanisms which are implemented on edge and fog layers. Upon close reading, we have observed that these mechanisms can be implemented to prevent DDoS attacks on IoT platform.

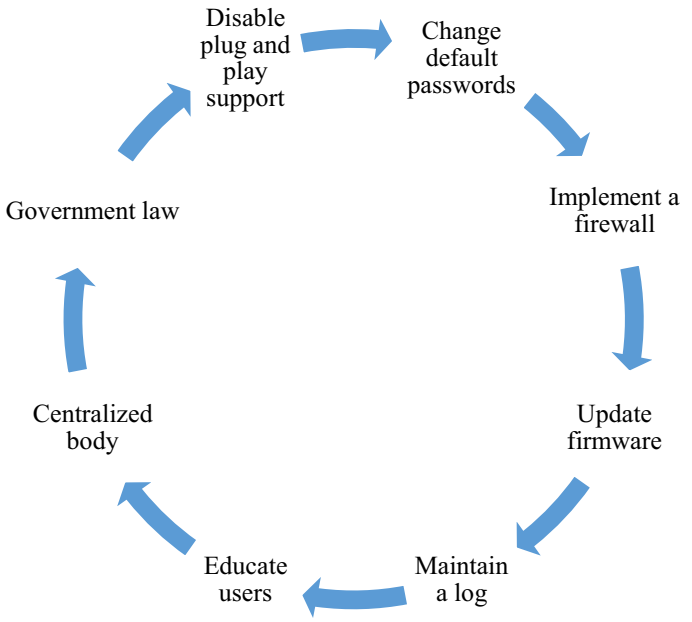

Fig. 6 Suggestions to secure IoT devices from DDoS attacks

6.2 Suggestions

Despite the studied solutions in Sect. 5, there are still some necessary protocols which must be in place to act as the first line of defense for IoT devices as shown in Fig. 6

- *Change passwords* Many of the IoT devices come shipped with a default password set on them. The problem arises when the entire product line has the same password in place. It does not take much time to figure out what is the default password set across the range; an attacker could own one of the devices and use the same password to attack other devices. An ideal solution would be to change the password in the device settings which hardly takes any time.
- *Implement a firewall* The first line of defense against foreign intrusion is a firewall placed on the network. However, many users look over this necessary protection mechanism assuming they are safe as long as their router password is not known to anyone else. An easy to implement solution is to make use of a Virtual Private Network (VPN) service to provide an additional layer of security which would monitor incoming traffic on their networks.
- *Update firmware* One of the critical weak points among the IoT devices is the lack of security updates from their manufacturers. The Mirai botnet attack exploited this weakness using the BrickerBot and operated many of these devices to attack their victims. The attacker will always seek new methods to

infect devices, and thus, it is imperative that device manufacturers periodically issue device firmware updates to stay a step ahead.

- *Centralized body* There is a need for a body that can provide certification to IoT devices based on their level of inbuilt security. Certain security standards must be set in stone that IoT device manufacturers must follow. Devices that can be easily exploited should either be labeled as insecure or refused to be sold at all. However, it is possible that cheap priced insecure devices can flood the market through back-channels and to prevent that from happening, the government can make it a law requiring all Internet Service Providers (ISP) to block internet access to such IoT devices.
- *Government law* A local government body should require by law that all ISP providers by default should implement preventive measures such as Ingress filtering. This filtering process helps in detecting attacking traffic and can be used to halt an ongoing traffic by blocking all packets with spoofed IP addresses. It can be implemented on edge devices such as a router which will deny access to bad traffic and allow legitimate traffic to pass through. For this to be an effective solution, a mandated law passed by the government must be in place.

7 Conclusion

In this survey, we presented motivations and reasons for attackers to select non-legacy IoT devices to launch DDoS attacks. We listed different types of tools that are available for attacking IoT devices to form a botnet and further tools are discussed which allow to use IoT bots to launch DDoS attacks. We presented a detailed and systematic classification of different types of DDoS attacks that take place on the cloud. We discussed how IoT devices are infected to be used as bots and presented a classification of DDoS attacks on the IoT environment. Furthermore, we studied and provided a detailed overview of twenty-one state-of-the-art defense mechanisms in existing and current research literature to prevent, detect and mitigate DDoS attacks. This survey helps in providing a complete overview of the types of attacks possible with specific kind of tools available and how to defend against them based on specific objectives, i.e., to detect an attack, to prevent or to mitigate them. Finally, recognizing the lack of security in IoT devices by default, we have suggested some protocols that must be in place as the first line of defense to protect IoT devices from being misused to launch a DDoS attack.

Acknowledgements This study was supported by the Research Program, which was funded by the Seoul National University of Science and Technology.

References

1. Douligeris C, Mitrokotsa A (2004) DDoS attacks and defense mechanisms: classification and state-of-the-art. Comput Netw 44(5):643–666
2. Kolias C, Kambourakis G, Stavrou A, Voas J (2017) DDoS in the IoT: mirai and other botnets. Computer 50(7):80–84

3. Jerkins JA (2017) Motivating a market or regulatory solution to IoT insecurity with the Mirai botnet code. In: 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), pp 1–5
4. Waterman S (2017) DDoS attacks growing faster in size, complexity—Arbor report. https://edscoop.com/ddos-attacks-growing-faster-in-size-complexity-arbor-report/. Accessed Dec 2018
5. Gartner Says Worldwide IoT Security Spending Will Reach $1.5 Billion in 2018. https://www.gartner.com/newsroom/id/3869181. Accessed Dec 2018
6. Industries most frequently targeted by denial of service (DDoS) attacks worldwide as of 4th quarter 2017 (https://www.statista.com/statistics/440600/ddos-attack-traffic-by-industry/). Accessed 9 Nov 2018
7. Kamboj P, Trivedi MC, Yadav VK, Singh VK (2017) Detection techniques of DDoS attacks: a survey. In: 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON), Mathura, pp 675–679
8. Mallikarjunan KN, Muthupriya K, Shalinie SM (2016) A survey of distributed denial of service attack. In: 2016 10th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, pp 1–6
9. Bhardwaj A, Subrahmanyam GVB, Avasthi V, Sastry H, Goundar S (2016) DDoS attacks, new DDoS taxonomy and mitigation solutions—a survey. In: 2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), Paralakhemundi, pp 793–798
10. Rai A, Challa RK (2016) Survey on recent DDoS mitigation techniques and comparative analysis. In: 2016 Second International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, pp 96–101
11. Divyasree IR, Selvamani K (2017) Defeating the distributed denial of service attack in cloud environment: a survey. In: 2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT), Kollam, pp 1–8
12. Vempati J, Dantu R, Thompson M (2018) Uninterrupted video surveillance in the face of an attack. In: 2018 17th IEEE International Conference On Trust, Security and Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), New York, NY, pp 843–848
13. Praseed A, Thilagam PS, In: IEEE Communications Surveys & Tutorials on DDoS Attacks at the Application Layer: Challenges and Research Perspectives for Safeguarding Web Applications
14. Mansfield-Devine S (2016) DDoS goes mainstream: how headline-grabbing attacks could make this threat an organisation's biggest nightmare. Netw Secur 2016(11):7–13
15. Nazario Jose (2008) DDoS attack evolution. Netw Secur 2008(7):7–10
16. Dramatic Increase of DDoS Attack Sizes Attributed to IoT Devices (https://www.bleepingcomputer.com/news/security/dramatic-increase-of-ddos-attack-sizes-attributed-to-iot-devices/). Accessed 3 Dec 2018
17. Behal S, Saluja K (2017) Characterization and comparison of DDoS attack tools and traffic generators -a review. Int J Netw Secur 19(3):383–393
18. Yadav S, Selvakumar S (2015) Detection of application layer DDoS attack by modeling user behavior using logistic regression. In: 2015 4th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), Noida, pp 1–6
19. Sauter M (2013) "LOIC Will Tear Us Apart": the impact of tool design and media portrayals in the success of activist DDOS attacks. Am Behav Sci 57(7):983–1007
20. Mansfield-Devine S (2011) Anonymous: serious threat or mere annoyance? Netw Secur 2011(1):4–10
21. Dantas YG, Nigam V, Fonseca IE (2014) A selective defense for application layer DDoS attacks. In: 2014 IEEE Joint Intelligence and Security Informatics Conference, The Hague, pp 75–82
22. Osanaiye O, Choo K-KR, Dlodlo M (2016) Distributed denial of service (DDoS) resilience in cloud: Review and conceptual cloud DDoS mitigation framework. J Netw Comput Appl 67:147–165
23. Badve OP, Gupta BB (2016) Taxonomy of recent DDoS attack prevention, detection, and response schemes in cloud environment. In: Proceedings of the International Conference on Recent Cognizance in Wireless Communication & Image Processing. Springer, pp 683–693
24. Apiecionek L, Makowski W (2015) Firewall rule with token bucket as a DDoS protection tool. In: 2015 IEEE 13th International Scientific Conference on Informatics, Poprad, pp 32–35

25. Iyengar NChSN, Banerjee A, Ganapath G (2014) A fuzzy logic based defense mechanism against distributed denial of services attack in cloud environment. Int J Commun Netw Inf Secur 6(3):233
26. Kumar G (2016) Denial of service attacks—an updated perspective. Syst Sci Control Eng 4(1):285–294
27. Ye K, Liu Y, Xu G, Xu CZ (2018) Fault injection and detection for artificial intelligence applications in container-based clouds. In: Luo M, Zhang LJ (eds) Cloud computing—CLOUD 2018: CLOUD 2018, vol 10967. Lecture notes in computer science. Springer, Cham
28. Specht SM, Lee RB (2004) Distributed denial of service: taxonomies of attacks, tools, and countermeasures. In: Proceedings of the 17th International Conference on Parallel and Distributed Computing Systems, 2004 International Workshop on Security in Parallel and Distributed Systems, pp 543–550
29. Kiruthika Devi BS, Saglani VJ, Gupta AV, Subbulakshmi T (2018) Classifying and predicting DoS and DDoS attacks on cloud services. In: 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, pp 1–5
30. Bhuyan MH, Bhattacharyya DK, Kalita JK (2015) An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection. Pattern Recogn Lett 51:1–7
31. Acharya AA, Arpitha KM, Santhosh Kumar BJ (2016) An intrusion detection system against UDP flood attack and ping of death attack (DDOS) in MANET. Int J Eng Technol (IJET) 8(2):1112–1115
32. Saied A, Overill RE, Radzik T (2016) Detection of known and unknown DDoS attacks using artificial neural networks. Neurocomputing 172:385–393
33. Gupta N, Jain A, Saini P, Gupta V (2016) DDoS attack algorithm using ICMP flood. In: 2016 3rd International Conference on Computing for Sustainable Global Development, pp 4082–4084
34. Hoque N, Bhattacharyya DK, Kalita JK (2015) Botnet in DDoS attacks: trends and challenges. IEEE Commun Surv Tutor 17(4):2242–2270
35. Phan TV, Van Toan T, Van Tuyen D, Huong TT, Thanh NH (2016) OpenFlowSIA: an optimized protection scheme for software-defined networks from flooding attacks. In: 2016 IEEE Sixth International Conference on Communications and Electronics (ICCE), Ha Long, pp 13–18
36. Phan TV, Bao NK, Park M (2016) A novel hybrid flow-based handler with DDoS attacks in software-defined networking. In: 2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress, pp 350–357
37. Czyz J, Kallitsis M, Gharaibeh M, Papadopoulos C, Bailey M, Karir M (2014) Taming the 800 pound gorilla: the rise and decline of NTP DDoS attacks. In: Proceedings of the 2014 Conference on Internet Measurement Conference, pp 435–448
38. Kawamura T, Fukushi M, Hirano Y, Fujita Y, Hamamoto Y (2017) An NTP-based detection module for DDoS attacks on IoT. In: IEEE International Conference on Consumer Electronics, Taiwan, pp 15–16
39. Zand A, Modelo-Howard G, Tongaonkar A, Lee SJ, Kruegel C, Vigna G (2017) Demystifying DDoS as a service. IEEE Commun Mag 55(7):14–21
40. Bawany NZ, Shamsi JA, Salah K (2017) DDoS attack detection and mitigation using SDN: methods, practices, and solutions. Arab J Sci Eng 42(2):425–441
41. Afek Y, Bremler-Barr A, Cohen E, Feibish SL, Shagam M (2016) Efficient distinct heavy hitters for DNS DDoS attack detection. Cryptography and Security
42. Choi S, Kwak J (2017) A study on reduction of DDoS amplification attacks in the UDP-based CLDAP protocol. In: 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), Kuta Bali, pp 1–4
43. Ko I, Chambers D, Barrett E (2018) A lightweight DDoS attack mitigation system within the ISP domain utilising self-organizing map. Proc Future Technol Conf (FTC) 881:173–188
44. Nagy B, Orosz P, Varga P (2017) Low-reaction time FPGA-based DDoS detector. In: NOMS 2018—2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, pp 1–2
45. Shah D, Kumar V (2018) TCP SYN cookie vulnerability. Networking and Internet Architecture
46. Mohammadi R, Javidan R, Conti M (2017) SLICOTS: an SDN-based lightweight countermeasure for TCP SYN flooding attacks. IEEE Trans Netw Serv Manage 14(2):487–497
47. Yan Q, Huang W, Luo X, Gong Q, Yu FR (2018) A multi-level DDoS mitigation framework for the industrial internet of things. IEEE Commun Mag 56(2):30–36
48. Choi J, Choi C, Ko B, Kim P (2014) A method of DDoS attack detection using HTTP packet pattern and rule engine in cloud computing environment. Soft Comput 18(9):1697–1703

49. Singh K, Singh P, Kumar K (2017) Application layer HTTP-GET flood DDoS attacks: research landscape and challenges. Comput Secur 65:344–372
50. Ehlert S, Geneiatakis D, Magedanz T (2010) Survey of network security systems to counter SIP-based denial-of-service attacks. Comput Secur 29(2):225–243
51. Geneiatakis D, Vrakas N, Lambrinoudakis C (2009) Utilizing bloom filters for detecting flooding attacks against SIP based services. Comput Secur 28(7):578–591
52. Rafique MZ, Akbar MA, Farooq M (2009) Evaluating DoS attacks against sip-based VoIP systems. In: IEEE Global Telecommunications Conference, Honolulu, HI, pp 1–6
53. Tripathi N, Hubballi N, Singh Y (2016) How secure are web servers? An empirical study of slow HTTP DoS attacks and detection. In: 2016 11th International Conference on Availability, Reliability and Security (ARES), Salzburg, pp 454–463
54. Cambiaso E, Papaleo G, Aiello M (2012) Taxonomy of slow DoS attacks to web applications. In: Recent Trends in Computer Networks and Distributed Systems Security, (CCIS), pp 195–204
55. Damon E, Dale J, Laron E, Mache J, Land N, Weiss R (2012) Hands-on denial of service lab exercises using SlowLoris and RUDY. In: Proceedings of the 2012 Information Security Curriculum Development Conference, ACM, pp 21–29
56. Yaar A, Perrig A, Song D (2004) SIFF: a stateless Internet flow filter to mitigate DDoS flooding attacks. IEEE Symposium on Security and Privacy, 2004. Proceedings. pp 130–143
57. Chapade SS, Pandey KU, Bhade DS (2013) Securing cloud servers against flooding based DDOS attacks. In: 2013 International Conference on Communication Systems and Network Technologies, Gwalior, pp 524–528
58. Srivastava A, Gupta BB, Tyagi A, Sharma A, Mishra A (2011) A recent survey on DDoS attacks and defense mechanisms. In: International Conference on Parallel Distributed Computing Technologies and Applications, Berlin, pp 570–580
59. Mahjabin T, Xiao Y, Sun G, Jiang W (2017) A survey of distributed denial-of-service attack, prevention, and mitigation techniques. Int J Distrib Sens Netw 13:1550147717741463
60. Stone R (2000) CENTERTRACK: An IP overlay network for tracking DoS floods. In: USENIX Security Symposium
61. Elleithy KM, Blagovic D, Cheng WK, Sideleau P (2005) Denial of service attack techniques: analysis, implementation and comparison. J Syst Cybern Inform 3(1):66–71
62. Nagy B, Orosz P, Tóthfalusi T, Kovács L, Varga P (2018) Detecting DDoS attacks within milliseconds by using FPGA-based hardware acceleration. In: NOMS 2018—2018 IEEE/IFIP Network Operations and Management Symposium, Taipei, pp 1–4
63. Zakaria N, Shamsi BA, Salah K (2017) DDoS attack detection and mitigation using SDN: methods, practices, and solutions. Arab J Sci Eng 42(2):425–441
64. Wankhede SB, Study of network-based DoS attacks. In: Nanoelectronics, Circuits and Communication Systems. Lecture Notes in Electrical Engineering, vol 511. Springer
65. Patel J, Katkar, A multi-classifiers based novel DoS/DDoS attack detection using fuzzy logic. In: Proceedings of International Conference on ICT for Sustainable Development. Advances in Intelligent Systems and Computing, vol 409. Springer
66. Mathew A, Terence JS (2017) A survey on various detection techniques of sinkhole attacks in WSN. In: 2017 International Conference on Communication and Signal Processing (ICCSP), Chennai, pp 1115–1119
67. Sejaphala LC, Velempini M (2017) Detection algorithm of sinkhole attack in software-defined wireless sensor cognitive radio networks. 2017 Global Wireless Summit (GWS), Cape Town, pp 151–154
68. Kaur M, Singh A (2016) Detection and mitigation of sinkhole attack in wireless sensor network. In: 2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE), Ghaziabad, pp 217–221
69. Mosenia A, Jha NK (2017) A comprehensive study of security of internet-of-things. IEEE Trans Emerg Top Comput 5(4):586–602
70. Hussain R, Abdullah I (2018) Review of different encryptionand decryption techniques used for security and privacy of IoT in different applications. In: 2018 IEEE International Conference on Smart Energy Grid Engineering (SEGE), Oshawa, ON, pp 293–297
71. Jain A, Jain S (2018) A survey on miscellaneous attacks and countermeasures for RPL routing protocol in IoT. Emerg Technol Data Min Inf Secur 814:611–620

72. Mehta R, Parmar MM (2018) Trust based mechanism for Securing IoT Routing Protocol RPL against Wormhole &Grayhole Attacks. In: 2018 3rd International Conference for Convergence in Technology (I2CT), Pune, pp 1–6
73. Shukla P (2017) ML-IDS: a machine learning approach to detect wormhole attacks in Internet of Things. In: 2017 Intelligent Systems Conference (IntelliSys), London, pp 234–240
74. Ahsan MS, Bhutta MNM, Maqsood M (2017) Wormhole attack detection in routing protocol for low power lossy networks. In: 2017 International Conference on Information and Communication Technologies (ICICT), Karachi, pp 58–67
75. Rajan A, Jithish J, Sankaran S (2017) Sybil attack in IOT: Modelling and defenses. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, pp 2323–2327
76. Valarmathi ML, Meenakowshalya A, Bharathi A (2016) Robust Sybil attack detection mechanism for Social Networks—a survey. In: 2016 3rd International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, pp 1–5
77. Evangelista D, Mezghani F, Nogueira M, Santos A (2016) Evaluation of Sybil attack detection approaches in the Internet of Things content dissemination. 2016 Wireless Days (WD), Toulouse, pp 1–6
78. Kang WM, Moon SY, Park JH (2017) An enhanced security framework for home appliances in smart home. Hum-Cent Comput Inf Sci 1(6):6
79. Rajagopalan M, Jagga M, Kumari A, Ali ST (2017) A DDoS prevention scheme for session resumption SEA architecture in healthcare IoT. In: 2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT), Ghaziabad, pp 1–5
80. Dao N-N, Phan TV, Ad USa, Kim J, Bauschert T, Cho S (2017) Securing heterogeneous IoT with intelligent DDoS attack behavior learning. Networking and Internet Architecture
81. Mehmood A, Mukherjee M, Ahmed SH, Song H, Malik KM (2018) NBC-MAIDS: naïve Bayesian classification technique in multi-agent system-enriched IDS for securing IoT against DDoS attacks. J Super Comput 74(10):5156–5170
82. Dao N, Vu D, Lee Y, Park M, Cho S (2018) MAEC-X: DDoS prevention leveraging multi-access edge computing. In: 2018 International Conference on Information Networking (ICOIN), Chiang Mai, pp 245–248
83. Bhardwaj K, Miranda JC, Gavrilovska A (2018) Towards IoT-DDoS prevention using edge computing. In: Workshop on Hot Topics in Edge Computing
84. Sahi A, Lai D, Li Y, Diykh M (2017) An efficient DDoS TCP flood attack detection and prevention system in a cloud environment. IEEE Access 5:6036–6048
85. Manoja I, Sk NS, Rani DR (2017) Prevention of DDoS attacks in cloud environment. In: 2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC), Chirala, pp 235–239
86. Doshi R, Apthorpe N, Feamster N (2018) Machine learning DDoS detection for consumer internet of things devices. Cryptography and Security
87. McDermott CD, Majdani F, Petrovski AV (2018) Botnet detection in the internet of things using deep learning approaches. In: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, Rio de Janeiro, pp 1–8
88. Nguyen S, Choi J, Kim K (2017) Suspicious traffic detection based on edge gateway sampling method. In: 2017 19th Asia-Pacific Network Operations and Management Symposium (APNOMS), Seoul, pp 243–246
89. Marques da Silva Cardoso A, Fernandes Lopes R, Soares Teles A, Benedito Veras Magalhães F (2018) Poster abstract: real-time DDoS detection based on complex event processing for IoT. In: 2018 IEEE/ACM Third International Conference on Internet-of-Things Design and Implementation (IoTDI), Orlando, FL, pp 273–274
90. Mondal HS, Hasan MT, Hossain MB, Rahaman ME, Hasan R (2017) Enhancing secure cloud computing environment by Detecting DDoS attack using fuzzy logic. In: 2017 3rd International Conference on Electrical Information and Communication Technology (EICT), Khulna, pp 1–4
91. Yuan X, Li C, Li X (2017) DeepDefense: identifying DDoS attack via deep learning. In: 2017 In: Proceedings of IEEE International Conference on Smart Computing (SMARTCOMP), Hong Kong, pp 1–8
92. Anirudh M, Thileeban SA, Nallathambi DJ (2017) Use of honeypots for mitigating DoS attacks targeted on IoT networks. In: 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP), Chennai, pp 1–4

93. Yin D, Zhang L, Yang K (2018) A DDoS attack detection and mitigation with software-defined internet of things framework. IEEE Access 6:24694–24705
94. Bhunia SS, Gurusamy M (2017) Dynamic attack detection and mitigation in IoT using SDN. In: 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), Melbourne, VIC, pp 1–6
95. Özçelik M, Chalabianloo N, Gür G (2017) Software-defined edge defense against IoT-based DDoS. In: 2017 IEEE International Conference on Computer and Information Technology (CIT), IEEE
96. Alharbi T, Aljuhani A, Liu H (2017) Holistic DDoS mitigation using NFV. In: 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, pp 1–4
97. Liu Y, Dong M, Ota K, Li J, Wu J (2018) Deep reinforcement learning based smart mitigation of DDoS flooding in software-defined networks. In: 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Barcelona, Spain, pp 1–6
98. Ahmed ME, Kim H (2017) DDoS attack mitigation in internet of things using software defined networking. In: 2017 IEEE Third International Conference on Big Data Computing Service and Applications, pp 271–276
99. Q1 2018 DDoS Trends Report: 58 Percent of Attacks Employed Multiple Attack Types. https://blog.verisign.com/security/q1-2018-ddos-trends-report-58-percent-of-attacks-employed-multiple-attack-types/. Accessed Jan 2019
100. Q2 2018 DDoS Trends Report: 52 Percent of Attacks Employed Multiple Attack Types. https://blog.verisign.com/security/ddos-protection/q2-2018-ddos-trends-report-52-percent-of-attacks-employed-multiple-attack-types/. Acessed Jan 2019
101. Suryani V, Sulistyo S, Widyawan W (2017) Internet of Things (IoT) framework for granting trust among objects. J Inf Process Syst 13(6):1613–1627
102. De Donno M, Dragoni N, Giaretta A, Spognardi A (2018) DDoS-capable IoT malwares: comparative analysis and mirai investigation. Security and Communication Networks
103. Vlajic N, Zhou D (2018) IoT as a land of opportunity for DDoS hackers. Computer 51:26–34

Affiliations

Mikail Mohammed Salim[1] · Shailendra Rathore[1] · Jong Hyuk Park[1]

Mikail Mohammed Salim
mikail@seoultech.ac.kr

Shailendra Rathore
rathoreshailendra@seoultech.ac.kr

[1] Department of Computer Science and Engineering, Seoul National University of Science and Technology (SeoulTech), Seoul 01811, Korea