International Journal on Semantic Web and Information Systems
Volume 18 • Issue 1

Distributed Denial-of-Service (DDoS) Attacks and Defense Mechanisms in Various Web-Enabled Computing Platforms:
Issues, Challenges, and Future Research Directions

Anshuman Singh, Infosys Ltd., India*

Brij B. Gupta, Department of Computer Science and Information Engineering, Asia University, Taichung, Taiwan & Research and Innovation Department, Skyline University College, Sharjah, UAE & Staffordshire University, Stoke-on-Trent, UK

ABSTRACT

The demand for internet security has escalated in the last two decades because the rapid proliferation in the number of internet users has presented attackers with new detrimental opportunities. One of the simple yet powerful attacks lurking around the internet today is the distributed denial-of-service (DDoS) attack. The expeditious surge in the collaborative environments, like IoT, cloud computing, and SDN, have provided attackers with countless new avenues to benefit from the distributed nature of DDoS attacks. The attackers protect their anonymity by infecting distributed devices and utilizing them to create a bot army to constitute a large-scale attack. Thus, the development of an effective as well as efficient DDoS defense mechanism becomes an immediate goal. In this exposition, the authors present a DDoS threat analysis along with a few novel ground-breaking defense mechanisms proposed by various researchers for numerous domains. Further, they talk about popular performance metrics that evaluate the defense schemes. In the end, they list prevalent DDoS attack tools and open challenges.

KEYWORDS

Blockchain, Botnet, Cloud Computing, Deep Learning, Distributed Denial-of-Service Attacks, IoT, Machine Learning, Web-Enabled Computing Platforms

1. INTRODUCTION

One of the most notorious attacks, raging around the Internet for more than 30 years, are the Denial-of-Service (DoS) attacks. The DoS attacks intend to paralyze the target by disrupting the connectivity between the target and its intended users and preventing users from network access. It exhausts sever resources like bandwidth, memory capacity, CPU processing power, etc. and brings down the entire

DOI: 10.4018/IJSWIS.297143 *Corresponding Author

network at last. This activity forces the target to shut down and reboot. The emergence of Distributed DoS attack was witnessed in summer 1999 (Criscuolo, 2000). Afterwards, the majority of DoS attacks occurring on the Internet are distributed in nature. The foremost purpose of these attacks is to crash the victim server and make it unavailable. It results in the revenue losses as well as economic overhead due to high cost of alleviating the attack and restoring the services.

The advent of new technologies has bestowed significantly greater number of resources, which has imparted attackers with novel ways to carry out cyber-attacks that cause more damage with less effort. A number of DDoS attacks are now carried out using well-organized and remotely controlled botnets. These botnets consist of thousands of malware-infected zombie machines that simultaneously send huge volumes of data to the target continuously, slowing down and eventually crashing the target system. Employing a bot army to execute an attack protects the anonymity of attacker by eliminating the chances of source IP address trace back. It also magnifies the severity of DDoS attack drastically.

In the recent years, a new landscape for DDoS attacks has emerged strikingly, called "DDoS as a Service". These are easily affordable and accessible DDoS-for-hire websites that have altogether remodeled the extent and impact of DDoS attacks around the Internet. Nowadays, the hackers carry out DDoS attacks for others for as little as $5 per hour. Since the release of Mirai botnet source code, powering 100,000 bots, for executing DDoS attacks on dark web in October 2016 (Bing, 2016), the demand as well as the supply of these attack services have rocketed radically. According to Corero COO Dave Larson, "as many as 40% of all network layer attacks are believed to be caused by such DDoS-for-hire botnets." These attack services are advertised as "Stresser" or "Booster" services that provide troubleshooting and testing services in order to identify the vulnerabilities in the user's network.

The recent studies have revealed that nowadays, not just numbers, the harshness of DDoS attacks has also aggravated. Cisco has predicted that the DDoS attacks are going to be even more frequent in the coming years, rising from 7.9 million in 2018 to a colossal figure of over 15 million in 2023. According to the Annual DDoS Threat Report for the year 2020 released by Nexusguard (2020), the frequency of DDoS attacks took an enormous jump from Q1 2019 to Q1 2020 with a year-over-year increase of 341.21\%. One of the largest ever recorded DDoS attack was carried out against the Amazon Web Services (AWS) in February 2020 with an attack volume that culminated to a breath-taking 2.3 Tbps (AWS, 2020). According to the Information Technology Intelligence Consulting (ITIC), an hour of IT services downtime can cost the companies anywhere between $300,000 to $1,000,000 (ITIC, 2019). Given this figure, the amount of financial damage incurred is unimaginable when a DDoS attack was brought down on thousands of Google's IP addresses in October 2020. The attack was perpetrated by three Chinese ISPs and lasted for six months, peaking at an astounding rate of 2.5 Tbps (Huntley, 2020).

The paper proposes a taxonomy of DDoS attacks followed by the detailed description of well-known and frequently occurring attacks along with a detailed analysis of how DDoS attacks impact various areas of the Internet. Despite being exhaustive, we do not claim that our proposed taxonomy is absolute. There are many new attacks and defense mechanisms emerging every day. We have also covered some published approaches available in research literature utilizing diverse technologies for the detection and mitigation of DDoS attacks. The goal of this paper is to provide researchers a deep insight that would trigger them to explore further and come up with diverse and innovative solutions to address DDoS attacks. Apart from all the areas detailing multiple aspects of DDoS attacks, this paper also encompasses DDoS attacks in multiple modern-day computing platforms, like IoT, Peer-to-peer network, Blockchains, smart cities, etc. Table 1 presents a comparison of multiple related works with our work in terms of various fields related to DDoS attacks that are surveyed and included in the literature.

Following the Introduction, the rest of the paper is organized as follows: Section 2 lays out some insights on the issue of DDoS attacks, why and how they are carried out along with a brief analysis of the structure of botnets. Section 3 includes a taxonomy of various DDoS attacks prevalent on the Internet today. Section 4 consists of a brief overview of multiple new web-enabled computing

Table 1. Comparison of related works in terms of domains incorporated

Contributions	Motivation	Botnet	Attack Taxonomy	Platform-specific Attacks	Traditional Defense	Integrated Defense	Performance Metrics	Attack Tools
Bhardwaj et al. (2016)			✓				✓	
Kamboj et al. (2017)			✓		✓			
Aamir et al. (2013)	✓		✓		✓	✓		
Kumar et al. (2009)	✓				✓			
Nagpal et al. (2015)			✓					✓
Zargar et al. (2013)	✓	✓	✓		✓		✓	
Peng et al. (2007)			✓		✓	✓		
Our work	✓	✓	✓	✓	✓	✓	✓	✓

platforms of the Internet that are affected by DDoS attacks. Section 5 presents a detailed analysis of some published methods based on cutting-edge technologies such as machine learning, big data, and blockchain, etc. Section 6 indexes major performance evaluation metrics that are utilized to assess the quality, strength, and dependability of the solutions for DDoS. Section 7 investigates multiple tools that are being employed to carry out DDoS attacks. Section 8 talks about some open research opportunities in developing a distributed defense mechanism against DDoS attacks. Finally, section 9 concludes the paper.

2. BACKGROUND

2.1 Internet Attacks Enhancement

In the early days of the Internet, one could provide scalable and flexible network conveniently, as the security issues were not a major concern. With the rapid increase in the Internet users over the past two decades, the victims of cyber-attacks have also grown significantly. According to the Internet Crime Report 2020 published by the Internet Crime Complaint Center (IC3) of Federal Bureau of Investigation (FBI), an enormous total of \$13.3 billion of victim losses have occurred due to cyber-attacks in the past five years out of which \$7.7 billion took place in the past two years only (IC3, 2020).

One of the early noticeable Internet security incidents that took place was the occurrence of the Morris Worm (Rochlis & Eichin, 1989) in 1988. Since then, the number as well as power of cyber-attacks have been increasing by the day. In the last decade, the relocation of financial and economic sectors from offline to online have also shifted the focus of attackers. In February 2020, the entire network infrastructure of UK cryptocurrency exchange EXMO was paralyzed by a high-scale DDoS attack with attack volume of 30 gbps (Haworth, 2021a). Another such attack knocked the New Zealand stock exchange offline for two days in a row (Haworth, 2020). Apart from the financial services, another sector that has presented itself as a major platform for DDoS attacks is Telecommunications sector. According to a study, the telecom sector rose from sixth most frequent DDoS target in Q4 2020 to the primary attack focus in Q1 2021 (Haworth, 2021b).

Year 2019 witnessed the rise of a global pandemic and consequently the businesses shifted their workforce to a full-time work from home model where majority users depend on relatively less secure infrastructure, paving the way for increased cyber-attacks (Bannister, 2020). According to PurpleSec 2021 threat report, cybercrime has escalated by 600\% due to Covid-19 pandemic (Firch, 2021). Along with the previously targeted sectors, like education and government departments, the coronavirus information websites also pose as potential targets to the attackers (Osborne, 2020). In a report published by the American technology and security company, Neustar Inc., the number of attacks mitigated by them have doubled from 2019 Q1 to 2020 Q1 (Leyden, 2020).

2.2 DDoS Attacks

The principle behind a Distributed Denial-of-Service (DDoS) attack is to render the victim unavailable for its intended users by disrupting the services provisionally or perpetually. The DDoS attacks achieve efficacy by taking advantage of multiple affected computer systems as source of attack traffic, ranging from a dozen to a 100,000. One commonly used approach for executing a DDoS attack is to exhaust all the resources of a network or web server by sending exorbitant volumes of data packets at an excessive rate, leaving the server inoperable. The attacker generates several requests via multiple attack sources and the victim drains all its network resources like CPU capacity, memory space, etc. in fulfilling those requests, disallowing access to legitimate clients. Another common approach for DDoS attack is to generate deformed packets to baffle an application or a protocol, rendering the victim machine frozen.

2.2.1 Why are DDoS Attacks Possible?

There are numerous reasons that make a DDoS attack possible. The data packets arriving from various sources makes it tremendously difficult to identify attack source IP address. In case of slow attacks, it is hard to identify the distinction between legitimate and illegitimate traffic, which leads to bypass the majority defense mechanisms allowing attack traffic through (Peng et al., 2007). Apart from these, the current internet design utilizes packet-switching architecture that allows all the users to share network resources and hence bandwidth attacks cause destruction in the network. This end-to-end architecture also leads to high IP Spoofing incidents, as there exists no way to authenticate a packet once it reaches the victim (Mirkovic & Reiher, 2004). Finally, the distributed nature of Internet provides the attackers with unenforceable accountability and, at the same time, making the deployment of cooperative defenses extremely difficult.

2.2.2 How to Perpetrate a DDoS Attack?

In order to carry out a DDoS attack successfully, the first thing that the attacker needs are multiple sources to send traffic to the victim. The attacker recruits the devices into the botnet by scanning for vulnerabilities to penetrate the security protocols of the remote hosts. This process is automatic, and the discovered vulnerabilities are exploited by the attacker to control the machine without being noticed by the owner. Once a device is infected with the help of an attack code, it is ready to obey the commands, issued by the botmaster, like scanning for further potential bots. Another way to recruit bots is by sending spoofed spam emails with malicious content disguised as a useful application. The victim misjudges the email to be benign and once opened, the content corrupts the victim machine by spreading malware like Trojans and enlisting the machine into the botnet.

2.2.3 Motivation Behind DDoS Attacks

Mirkovic & Reiher (2004) proposed four major reasons behind committing DDoS attack and inflicting damage onto the victim. Usually, the motivation is a personal reason. It's intended to be either fun or vengeful (or both) while at the same time demonstrating the power to disrupt a website or network, like cyberbullying and trolling. Another reason could be Hacktivism which gains respect for the hacker community by showing support or opposition regarding a certain topic like Olympic Games

(Kaspersky, 2021) or due to ethical concerns like the attack on WikiLeaks (Schonfeld, 2010). The material gains like financial or economic benefits are also a growing motivation behind DDoS attacks against corporations. The business establishments too get tempted to launch DDoS attacks against their market competitors (Ashford, 2017). Other significant motives behind DDoS attacks could be political or strategic. Cyberwarfare is Usually conducted by a well-trained and organized group like the military of a nation or a terrorist group, to adversely affect the enemy's resources that may inflict economic or physical loss on them (Hanna, 2021).

2.2.4 Botnets

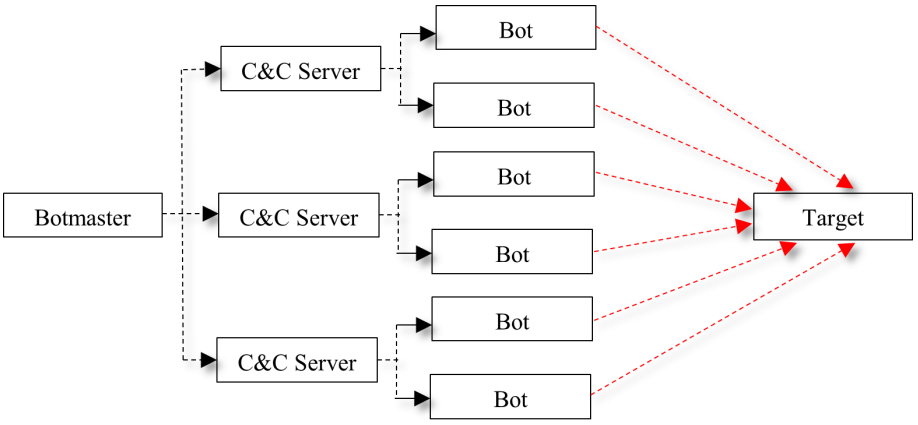
One common way to execute DDoS attacks is by taking advantage of many compromised machines called bots or zombies. These bots are connected to one another through the Internet, forming a group called a botnet. Every botnet has a Botmaster that communicates with all the bots, commanding them through a C&C server (Li et al., 2009) to carry out malicious activities. Figure 1 illustrates a DDoS attack carried out by an attacker that utilize a botnet. These bots take orders from the botmaster and perform specific tasks, may be repeatedly, to destroy the target network, system or web server.

2.2.4.1 Botnet Communications

The botnet communications are carried out by a Command and Control (C&C) server. The C&C server is a computer that is controlled by the attacker to send commands to zombie systems to carry out an attack. Several types of C&C mechanisms are proposed in the existing literature (Eslahi et al., 2012; Hoque et al., 2015; Khattak et al., 2013) and the C&C architectures used for communication are either centralized or decentralized:

- **Centralized C&C servers:** In this approach, the botmaster is connected to the C&C server to command the bots and the bots are also connected to the server to receive commands and updates. The centralized C&C servers are simple to manage on account of their single point of failure, making the response fast.
- **IRC botnet (Zargar et al., 2013):** The IRC (Internet Relay Chat) is a text-based chat system that allows computer users to communicate with multiple participants in a so-called conversation channel. In a botnet, the bots connect to a specific channel in the IRC server and wait for instructions. The IRC networks are relatively easy to construct. They use simple, low bandwidth communication methods, making them widely used to host botnets. Also, they are able to continually switch channels to avoid being taken down, making them an ideal

Figure 1. Botnet Employed DDoS Attack



choice for coordinating DDoS attacks and spam campaigns (Eslahi et al. 2012). When an IRC bot connects to a specific channel, it stays in the connected state, also known as the PUSH approach (Gu et al., 2008a).

- **HTTP botnet (Zargar et al., 2013):** HTTP botnets use HTTP protocol for C&C communication and to control the bots (Koo et al., 2011). HTTP botnet C&C server works just like a normal web server and the bot works just like a normal web client. In a web-based botnet, bots connect to a specific URL or IP address described by the botmaster that plays the role of the C&C server (Hsu et al., 2017). Instead of the PUSH approach employed by the IRC botnet, the bots in the HTTP based botnet make use of a PULL approach. The bots need not stay in the connected state after connecting to the C&C server for the first time. Instead, the commands are posted on the specific web server and the bots regularly update themselves by visiting those web servers to get new commands at regular intervals, predefined by the botmaster (Eslahi et al. 2012; Gu et al., 2008b).
- **Decentralized C&C servers:** In this decentralized approach, the C&C botnet architecture is based on the peer-to-peer (P2P) communication protocol. A P2P botnet offers high flexibility to the network because every P2P node act as a bot as well as the C&C server. The P2P botnet is relatively complex to manage as there is no central server to propagate the commands defined by the botmaster. Each bot spreads the commands to its neighboring nodes until all the nodes receive the commands issued by the botmaster (Eslahi et al. 2012). There is no single specific channel or port for the bots to connect and thus P2P botnets are more difficult to detect, making them highly resistant to termination. The PULL approach renders a special advantage that even though some botnets are detected and taken down, the communication among the botmaster and other P2P nodes could continue (Su et al., 2018).

2.2.4.2 Botnet Functions

The basic functions that a botnet usually perform are infection and propagation, command and control and attack eventually. Apart from phishing emails and 'water-holing' techniques used in the early days of the Internet, a new infection and propagation method is being employed these days. It utilizes the steganographic techniques, which 'embed' the botnet code into a picture or a PDF document attached to an email that often mimics a colleague or friend. A new and unique form of botnets coming into play today are the Fast Flux Networks (FFNs) which promise high flexibility and availability (Al-Nawasrah et al., 2020). It keeps on changing the IP addresses of the domain names in order to avoid detection and prospective shutdown by intrusion detection systems. As noted above, the purpose of the botnet command and control is to enable the communication between the bots and botmaster. The fourth and supreme function of botnet is to attack. There are many potential malicious botnets, like BuleHero and Mirai, which perform various activities that include DDoS attacks, unsolicited spamming, stealing personal and financial information, click fraud and adware (Eslahi et al. 2012).

3. TAXONOMY OF DDOS ATTACKS

With a diversified range of DDoS attacks present around the Internet, it becomes extremely difficult to trace or detect these attacks owing to their distributed nature. The attackers usually spoof the IP addresses i.e., create a false IP address that can be either the IP address of the target device or a fake address, to disguise themselves and keep their identities hidden. This makes the identification of a DDoS attack source even difficult. In order to deploy a successful mitigation scheme for DDoS detection, it is extremely important to understand the characteristics and possible effects of DDoS attacks. Numerous domain-specific classifications of DDoS attacks have been proposed by myriads of researchers in past (Chen et al., 2004; Douligeris & Mitrokotsa, 2004; Li et al., 2009; Mirkovic & Reiher, 2004; Peng et al., 2007; Riorey, 2012; Sharafaldin et al., 2019; Specht & Lee, 2003; Tariq et al., 2006; Yue et al., 2009; Zargar et al., 2013; Zhijun et al., 2020). However, in this paper, we

present a classification of DDoS attacks based on the ways the attack is perpetrated and that covers multiple domains. Figure 2 presents the proposed taxonomy of DDoS attacks. In the remainder of the section, we analyze some of the most notorious DDoS attacks that are prominent in present-time.

3.1 Application-Layer Attacks

These are the attacks that exploit weaknesses in the application layer by opening connections and initiating new processes. These processes involve transaction requests that consume server resources like disk space and memory; subsequently, leaving the server unavailable to process the legitimate user requests. The application layer attacks (Dantas Silva et al., 2020; Yu et al., 2007) cause more damage however using very small bandwidth. Nevertheless, they are harder to detect since they are indistinguishable from legitimate traffic and cause similar impact to the services.

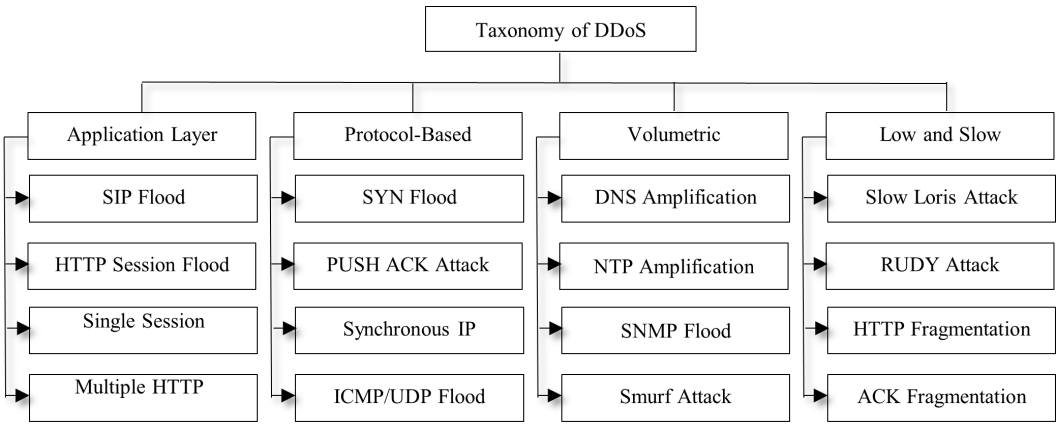
3.2 VoIP Flooding

The VoIP Flooding (Riorey, 2012; Zargar et al., 2013) or SIP Flooding (Peng et al., 2007) attack is slightly different from application specific UDP packet flooding. The VoIP flood is carried out by attacking the victim server by heavily populating fake VoIP requests through an SIP (Session Initiation Protocol) connection from a wide range of IP addresses. Generally, the victim does not possess enough resources to process these valid as well as invalid requests together that results in server overloading. It is difficult to differentiate an attack from legitimate traffic due to the use of connection-less UDP protocol for transporting the packets. If the flood is launched from a botnet using non-spoofed IP addresses, the traffic appears to be originated from the legitimate VoIP servers. The SIP proxy servers and the call receivers are victims in this attack.

3.3 HTTP GET/POST Flooding

The HTTP GET/POST Flooding (Zargar et al., 2013) (a.k.a. Excessive VERB (Riorey, 2012)) attack damages the victim web server by continuously swamping it with valid HTTP GET/POST requests. Here, instead of IP spoofing, the attacker evades suspicion by utilizing a botnet with valid IP addresses to carry out the attack. Since the bots use their non-spoofed IP addresses, the defense mechanisms find these HTTP requests legitimate and do not flag them. A single bot can send a large number of HTTP GET/POST requests to execute an attack and hence the attacker can use significantly fewer number of bots to achieve a large-scale attack. Consequently, the target server can be completely crippled by a handful of bots in case of an HTTP DDoS attack.

Figure 2. Taxonomy of DDoS Attack



3.4 Single-Session HTTP GET/POST Flooding

The Single-session HTTP GET/POST Flooding (Zargar et al., 2013) or Excessive VERB Single Session (Riorey, 2012) attack is a request flooding attack that exploits a loophole in HTTP 1.1. It sends multiple requests, using a single HTTP session, which culminates in DDoS flooding attack. It allows attackers to generate plenty of requests using few sessions only. In other words, the attackers can evade the bound imposed by DDoS defense mechanisms on maximum sessions that a single user can initiate. The Single Session HTTP Flood aims the server's assets to compromise the performance or targets complete system breakdown.

3.5 Multiple HTTP GET/POST Flooding

The Multiple HTTP GET/POST Flooding (Zargar et al., 2013) or Multiple VERB Single Request (Riorey, 2012) attacks were devised with workarounds to sidestep the defense mechanisms that block many incoming packets. Similar to Single Session HTTP flooding attack, this transformation of HTTP flood also exploits a loophole in the HTTP technology. The loophole enables a single HTTP session to make multiple HTTP requests by concealing them within a single HTTP packet instead of issuing them one after the other during an HTTP session. This trick allows an attack to stay invisible to net flow anomaly detection systems and to consume the server's resources by keeping packet rates within permitted bounds.

3.6 Protocol-Based Attacks

The Protocol-based attacks or the state-exhaustion attacks are designed to exhaust the processing capacity of network resources as well as of intermediate communication equipment, like server and firewall, by targeting Layer 3 and Layer 4, with malicious connection requests.

3.6.1 SYN Flooding

The SYN Flooding (Patel & Borisagar, 2012; Riorey, 2012) attack circumvents the three-way handshake process required to establish TCP connections between clients and servers. In three-way handshake, a SYN request packet is sent to initiate a TCP connection with a server, which returns a SYN-ACK packet, and in the end the client confirms the receipt of SYN-ACK by sending an ACK packet. In a SYN flood scenario, the attacker sends multiple SYN requests; however, either it sends the SYN requests from a spoofed IP address, or does not respond to the host's SYN-ACK response. The target system, in either case, binds its resources and waits for the acknowledgement for every request. The resources are drained out to a point where no fresh connections could be initiated. The enervated server fails that leads to the denial of service eventually.

3.6.2 PUSH ACK Flooding

The PUSH ACK Flooding (Patel & Borisagar, 2012; Riorey, 2012) attack is carried out by overwhelming the victim with spoofed ACK packets that do not belong to any active session within the server's connection list. Though these packets have no active session linked to them, the server spends plenty of resources on matching them to their alleged session and consequently, making the server inoperable to process any legitimate request. It may result in performance degradation or complete server crackdown until the attack lasts.

3.6.3 Synonymous IP Attack

The Synonymous IP Attack (Riorey, 2012) intends to take down a server by sending large number of spoofed TCP-SYN packets. The spoofed packets carry the victim server information as source address as well as destination address. The target server starts using additional resources to address this anomaly and due to this live lock condition, it fails to process any legitimate request.

3.6.4 ICMP/UDP Flooding

The ICMP/UDP Flooding (Riorey, 2012) attacks flood the victim server with UDP or ICMP packets. Being a connection-less protocol, it is harder for any defense mechanism to identify a UDP/ICMP flood attack. These attacks generally tend to preoccupy the entire bandwidth available in a network. Myriads of spoofed UDP/ICMP packets are sent to a target server from a massive set of source IP to exhaust network resources especially bandwidth.

3.7 Volumetric Attacks

The volumetric attacks are a classic type of DDoS that employ some amplification method, like requests from a botnet, to generate massive volume of traffic that saturate the bandwidth and paralyze the targeted site (Dantas Silva et al., 2020). One way to achieve the goals is by utilizing a botnet and multiple third-party machines that unwittingly participate in a DDoS attack on the target. The bots are instructed by the attacker to send spoofed traffic with victim's IP address as the source to the third parties. The high volume of response generated by the third parties is then sent to the victim, constituting a Distributed Reflector Denial-of-Service (DRDoS) attack. The DRDoS attacks are asymmetric in nature because the response is much larger in volume than the original request sent.

3.7.1 DNS Amplification

The DNS Amplification (Patel & Borisagar, 2012; Peng et al., 2007; Riorey, 2012) attacks are executed by sending large amount of spoofed DNS request packets that appear identical to valid requests from a huge set of source IP. The target server is unable to differentiate between legitimate and illegitimate DNS requests. Eventually, the server exhausts its resources in the attempt to serve all the requests. The attack cripples the network by consuming the entire available bandwidth. The spoofed DNS requests that output large amount of data can also be generated with a victim's IP address as source. It causes the DNS server to send huge response packets to the victim causing a DDoS attack that appears to be originated at the DNS server.

3.7.2 NTP Amplification

The NTP Amplification attack exploits NTP (Network Time Protocol) servers to overwhelm a targeted server with UDP traffic. The query-to-response ratio in such scenarios could range within 1:20 to 1:200 or higher. The small queries are sent by the attacker via botnet with victim's IP address spoofed as the source address. Once the victim receives the response flood, it ends up exhausting its resources in handling the flood and may reboot or go offline.

3.7.3 SNMP Amplification

The SNMP Amplification attack works similar to NTP Amplification attack. Small packets carrying a spoofed IP of the target are sent to the internet enabled devices running Simple Network Management Protocol (SNMP). These devices generate UDP flood as response to the spoofed requests and send them to the target causing a DDoS attack. As the number of respondent devices increases, the attack volume grows until the target network is crashed due to the collective SNMP responses.

3.7.4 Smurf Attack

The Smurf Attack (Patel & Borisagar, 2012) is a network layer distributed denial of service (DDoS) attack, named after the 'DDoS.Smurf' malware that enables it execution. In fact, it is an amplification attack vector that magnifies its damage potential by exploiting the characteristics of broadcast networks. In an IP broadcast network, bots send a spoofed ping request with victim's IP address as the source address to every host. Each host sends an ICMP response to the spoofed source address. Once enough ICMP responses are forwarded, the target server goes down.

3.8 Low and Slow Attacks

In this type of DDoS attack, partially formed packets are generated and sent over the network as slow as possible in order to defy the session time out. The victim waits for the remainder of a packet for long periods and eventually becomes unavailable for legitimate users once the number of concurrent connections is maximized. These attacks are hard to detect as the common defense mechanisms rely on high packet rate and thus fail to identify the partially formed packets and let them through.

3.8.1 SlowLoris Attack

The SlowLoris Attack \cite{zargar2013survey} is carried out by repeatedly sending partially formed HTTP requests for opening new connections to the victim server. These requests overwhelm the server by keeping the newly formed connections open for longest possible duration and eventually disallows the formation of new connection. The attacker periodically sends a fragment of the request to avoid the connection being timed out. This type of DDoS attack requires minimal bandwidth to launch and can be executed using a single machine.

3.8.2 R-U-Dead-Yet (RUDY) Attack

The R-U-Dead-Yet (RUDY) Attack (Zargar et al., 2013) is executed by sending an HTTP POST request and slowly crashing the web server by submitting long form-fields in low volumes that appear as legitimate traffic. The attacker sends a properly formed HTTP header which contains an abnormally long header-field. Subsequently, it proceeds by injecting single byte of information followed by long wait and thus puts the application threads in a wait loop to perform processing. The multiple simultaneous connections eventually exhaust the server's connection table constituting a denial-of-service condition.

3.8.3 HTTP Fragmentation

The HTTP Fragmentation (Riorey, 2012; Zargar et al., 2013) attack is similar to Slowloris attack and is accomplished by opening valid HTTP connections and keeping them alive for longest possible duration without raising any alarm. The bots with legitimate IP addresses establish a genuine HTTP connection with a web server. Subsequently, these bots split the HTTP packets into tiny fragments and send them to the target as slowly as it allows before the time out. Thus, it allows the attackers to keep a connection open for as long as possible in order to bypass the defense mechanism. An attacker can bring down a web server with a handful of bots since one bot can initiate multiple undetected, extended, and resource consuming sessions.

3.8.4 ACK Fragmentation

The ACK Fragmentation attack utilizes 1500-byte packets with the motive of consuming the target network's bandwidth without generating a high packet rate. Therefore, if the application-level filters are embedded in the network equipment, they need to apply packet reassembling, which consumes much of their resources. If no filters are applied, these attack packets will be able to pass undetected through many network devices such as routers, ACLs, and firewalls. The fragmented packets containing junk data consume entire bandwidth perpetrating a DDoS attack.

3.8.5 DDoS Attacks in Different Web-Enabled Computing Platforms

The world came to know about DDoS attacks in the summer of 1999 (Criscuolo, 2000) and since then it has become one of the most dynamically advancing vectors of cybercrime. The growing attack power, along with cheap availability of attack tools, makes DDoS the number-one choice for attackers on the globe. In this section, we will look at how DDoS attacks are constituted in some of the major fields around the Internet.

3.9 DDoS Attacks in Traditional Environments

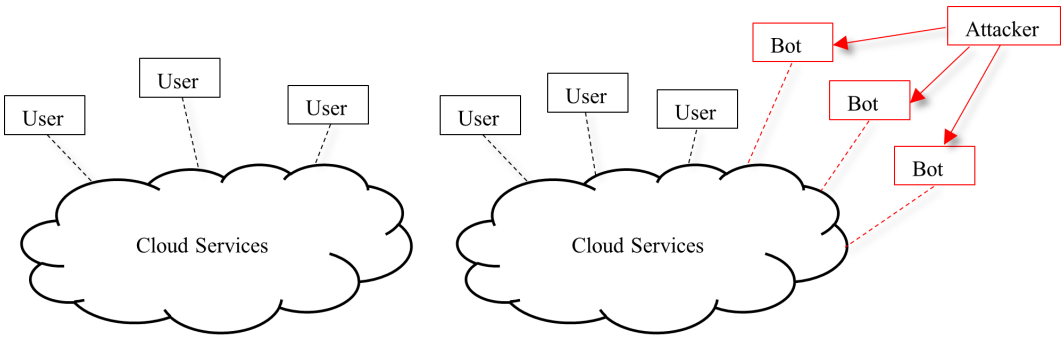
In the previous section, we have seen how DDoS attacks are perpetrated in traditional networked environments. The attacker first scans for potential remote machines with open ports to exploit. Once such vulnerable machines are located, the attacker infects them using an attack code and exploits them in order to create a botnet without the knowledge of their owners. The botnet is then utilized to perform DDoS attacks on the intended target. The target is flooded with myriads of requests either by a handful of machines or a single machine with spoofed IP address or by a bot army with non-spoofed valid IP addresses. The spoofing makes attack source identification near to impossible preventing the anonymity of the attacker. These requests overwhelm the target by consuming all its resources since the target puts its assets namely CPU computing time, network bandwidth, stack memory etc. in handling these requests. As the attack power escalates, the target pools additional resources trying to compute these requests and when the attack power is peaked, the target gets exhausted. This causes the target to crash and reboot, which blocks it from serving legitimate clients and falling prey to a successful DDoS attack.

3.10 DDoS Attacks in Cloud-Based Environments

The cloud computing has emerged as one of the most prominent technologies due to the on-demand availability of resources like storage and computing power. In addition to being cost-efficient, it provides multiple benefits to its users like flexibility, disaster recovery, increased collaboration etc. According to a study by the International Data Group (IDG, 2020), 81% of businesses are already using cloud technology in one capacity or another. With more and more data shifted over cloud, the cloud security is a new concern for the users. The DoS and DDoS attacks are the dominant barriers that impacts the availability of cloud.

The on-demand and self-service are the characteristics of cloud technology that assist attackers to create a powerful botnet almost instantly by infecting a large number of devices in short time (Yan & Yu, 2015). Another reason behind increased DDoS attacks in cloud is the virtualization technology that lets attackers create multiple virtual machines using little disk space and thus launch more attacks at low cost. Due to the multi-tenant infrastructure of the cloud, the attack against a single cloud user result in the attack against all users of that cloud. In addition, a new breed of DDoS attacks, called the Economic Denial-of-Sustainability (EDoS) (Shawahna et al., 2018; Xiao & Xiao, 2012) are becoming prominent because the pay-per-use policy of cloud eliminates the key requirement for DDoS attacks i.e., resource bottleneck. The EDoS attack exploits the cloud elasticity and auto-scaling features to charge a cloud adopter bill an excessive cost resulting in large-scale service withdrawal or bankruptcy. Figure 3 illustrates the state of the cloud before and after an EDoS attack where attacker blocks the

Figure 3. EDoS Attack Scenario



cloud resources using a botnet. In such case, the cloud continuously allocates additional resources as required and consequently, an exorbitant amount of bill is charged to the user.

3.11 DDoS Attacks in P2P Environments

Unlike the traditional centralized client-server network models, a peer-to-peer (P2P) network contains multiple connected computers with no central control to pass the data through. Figure 4 demonstrates the basic architectural difference in a server-based and a P2P network. All users that join the network are peers and hence allowed to receive and send files to other machines in the network. All the peers share resources like storage, computational power, and bandwidth. Thus, P2P file sharing can be exploited to spread malware and gather personal and financial information. Since each node also acts as router, the malware spreads faster aiding severe DDoS attacks in a P2P network (Naoumov & Ross, 2006; Yue et al., 2009). A study demonstrating the impact of a DDoS attack on a P2P network by simulating the same using Gia network is available in (Qwasmi et al., 2011).

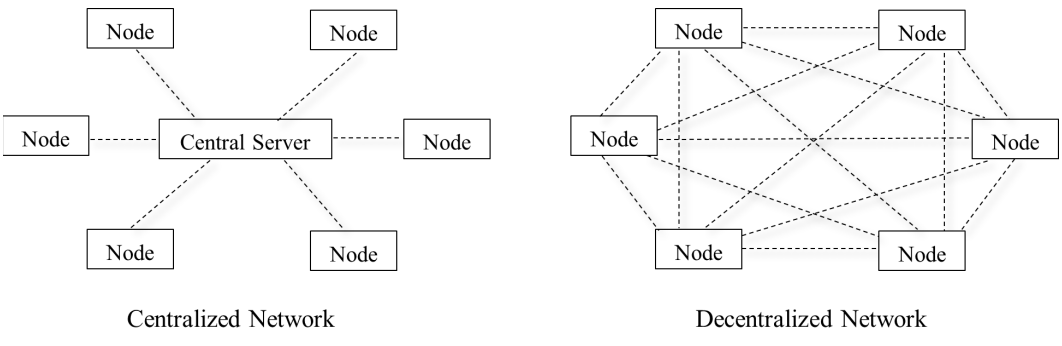
A query flood can be initiated in a P2P network by a malicious peer by broadcasting massive queries in the network, eventually paralyzing the network. A P2P network could also be used as an agent to carry out DDoS attack on the intended target. The peers can overwhelm the target by issuing huge query requests to exhaust the target's resources like CPU processing, bandwidth, etc. Another way to execute DDoS attack in a decentralized network is by injecting junk data, which increases query time and turns over invalid results. This could be achieved either by index poisoning, where the attacker inserts fake records of target IP address and port number, or by route table poisoning, where the attacker tricks the peers into adding false neighbors in the route table (Qi, 2009).

3.12 DDoS Attacks in Blockchain

The blockchain technology records information in a manner that makes it impossible to alter, hack, or evade the system. A blockchain can be visualized as a digital ledger of transactions that is duplicated and dispersed across the entire network of the blockchain. Within each block of the chain, there are multiple transactions. Whenever a new transaction takes place on the blockchain, every participant's ledger appends a record of that transaction with a fixed cryptographic signature called a hash. This eliminates the chances of tampering with any of the blocks in a chain. In order to corrupt a blockchain, every block in the chain has to be changed, across all the distributed versions of the chain.

Prima facie blockchain appears unshakable however it is susceptible to several Internet attacks, like Sybil attack, 51% attack, Phishing attack, BGP Routing attack, and DDoS attacks (Wen et al., 2021). Various papers in the past have demonstrated many different ways to execute DDoS attacks in a blockchain environment (Saad et al., 2018; Wang & Li, 2019). Mirkin et al. (2020) explained a new form of DDoS attack called Blockchain Denial-of-Service (BDoS) that could seize the functioning of a blockchain with remarkably little resources. The attackers exploit the incentive mechanism of the

Figure 4. Server-based Network and Peer-to-Peer Network architecture



system by utilizing their resources in discouraging the rational miners to stop mining. The attacker only publishes the header of a generated block, implying a decrease in the expected profitability of the rational miner. This causes the miner to stop mining and if a significant profitability decrease is achieved, all the miners stop mining. At this point, the attacker can also stop mining with an advantage of one block originally generated. It causes the blockchain to halt eventually.

Another type of DDoS attack is presented by Wu et al. (2020) that targets the mining pools. The mining pools are a collection of miners that come together and pool their respective mining powers to successfully mine a block and earn steady rewards. These miners are unable to find a block individually owing to their limited resources. The miners in a mining pool share the earned reward proportional to their respective mining power. Saad et al. (2019) discussed another form of DDoS attack on the memory pool where an attacker can consume entire memory by issuing invalid transactions and preventing further mining. Higher the size of mempool, higher is the mining fee paid by the user. By flooding the mempool with invalidated transactions, the attacker tricks the user into paying more.

3.13 DDoS Attacks in Internet of Things (IoT)

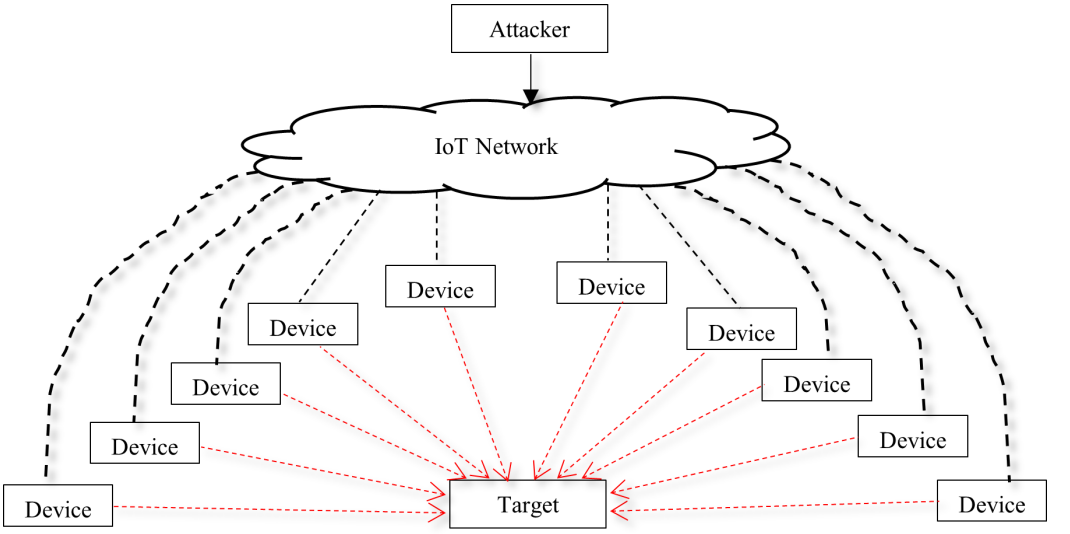
The Internet of Things (IoT) is a revolutionary technology that connects a device with an ON/OFF switch to billions of other such devices and to the Internet. It is an enormous network of wide array of devices, ranging from smart vehicles that can detect objects in their way to healthcare devices that include highly popular fitness trackers counting heart rate. The sensors embedded in these devices continuously collect and share data among themselves and the cloud for further computations. This gigantic network is getting bigger by the day, merging digital and physical universes, with more and more devices connecting to it. According to a study by Juniper Research (Juniper, 2020), an increase of 130% in the number of IoT connections is expected over the next three years, rising from 35 billion in 2020 to 83 billion in 2024.

The data collected by sensors may be extremely sensitive in some cases, making security a prime concern in IoT networks. A wide range of attacks on IoT devices, including DDoS attacks, have been discussed by Munshi et al. (2020). The botnets, like Mirai (Kolias et al., 2017), pose a serious threat of DDoS attacks to IoT networks. Recently, the A10 networks have tracked down approximately 12.5 million unique source addresses of exploited hosts in their DDoS threat report (A10, 2020). Apart from DDoS attacks on itself, IoT network can also be utilized by an attacker to infect other devices. An attacker can infect single IoT device with minimal effort owing to the modicum security standards, turning it into a zombie. Subsequently, attacker can utilize the zombie to infect the entire cloud, to which the zombie IoT device sends the recorded data. Consequently, all the unsecured connected IoT devices are affected. Therefore, turning IoT network into a powerful and well-connected botnet that can carry out DDoS attack of high magnitude. Figure 5 illustrates an IoT network being utilized by an attacker to execute a DDoS attack against a target. In such cases, the attacker's goal is not to interfere with a user's daily tasks, but to harness hundreds and thousands of devices creating a robust zombie army that is capable of bringing an otherwise secure corporate network to its knees in a matter of seconds.

3.14 DDoS Attacks in SDN Environment

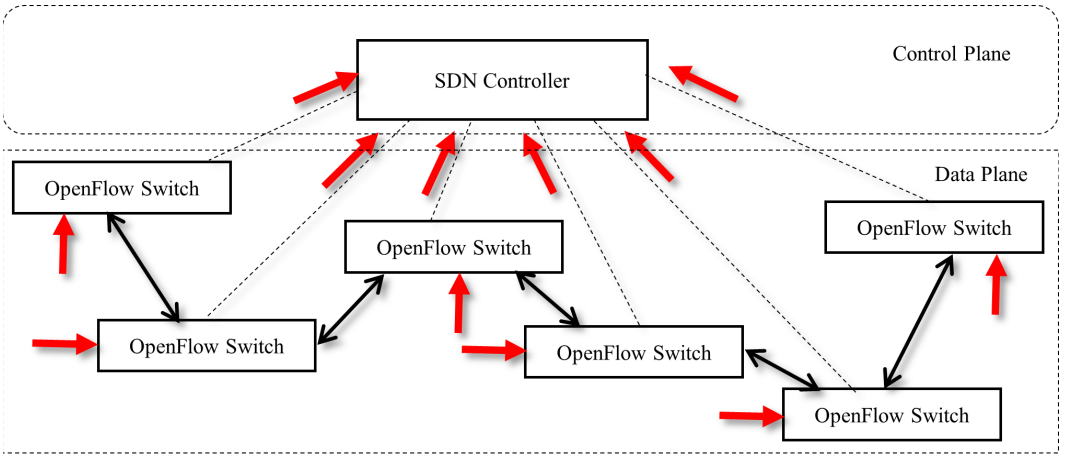
The rapid increase in number of consumers and their growing requirement for modern networks have advocated the development Software-defined Networking (SDN) in the past decade. The SDN architecture has addressed various drawbacks of traditional networks, like manual configuration, less agility, and complex infrastructure, by centralized software monitoring, flexible operational procedures, infrastructure abstraction, etc. This increases the efficiency and consequently the performance of entire network. The SDN provides these advantages by decoupling the data plane from the control plane and a standardized OpenFlow protocol is implemented on interface between the layers.

Figure 5. IoT Network acting as a botnet for a DDoS attack



The inherent architecture of SDN provides it with the capability to detect and mitigate DDoS attacks effectively in a cloud-based network, but this basic structure also renders SDN vulnerable to DDoS attacks on itself. The separation of data plane and control presents attackers with new attack planes (Dong et al., 2019). Figure 6 illustrates a DDoS attack scenario on an SDN architecture. In one such attack scenario, the attacker floods the SDN network with fake requests. The SDN switches hand over these requests, containing no valid return address or packet source IP, to the controller for computation. These requests overwhelm the controller and renders the network frozen. The controller waits for return address keeping the connection active for long time which consequently leaving the network unreachable for legitimate users. The aim of attacker in such attack cases is not to gather any confidential information but to deplete the resources of controller by wasting time (Parashar et al., 2019).

Figure 6. DDoS attack on SDN controller



3.15 DDoS Attacks in Cellular Networks

The upcoming 5G network technology is expected to meet various critical requirements like extremely low latency, increased reliability, efficiency, availability, superior performance, and enormous network capacity along with breakneck data rates. As a result, many different fields, like augmented reality (AR), virtual reality (VR), healthcare etc., are awaiting the onset of 5G technology. With the upsurge of 5G technology the Cyber-Physical Systems (CPS) (Humayed et al., 2017), employed across various domains involving critical infrastructures, will depend heavily on cellular networks, exposing to a wide array of Internet attacks (He et al., 2018; Mavoungou et al., 2016).

The DDoS attacks in a cellular network can be particularly destructive since the extent of attack is two folds: First, DDoS attack of a significant scale can excessively infect and influence a network and all its legitimate users. Secondly, CPSs are an intricate integration of digital and physical systems, a DDoS attack in cellular networks can be more disastrous and lasting, depending on the application domain. The attacker can exploit various vulnerabilities, to carry out multiple attacks like silent call (He et al., 2018; Tu et al., 2015), SMS flooding (He et al., 2018; Murynets & Jover, 2013), and signaling (Gupta et al., 2013; He et al., 2018) and DDoS attacks. These attacks utilize a vast botnet, containing plenty of malware/spyware infected mobile devices, which can perpetrate an attack or gather personal/financial information on botmaster's commands.

3.16 DDoS Attacks in Smart Cities

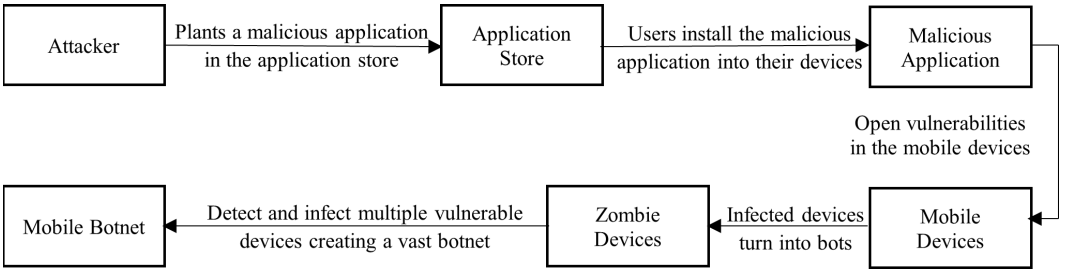
A smart city (Chen et al., 2020) is a centralized framework primarily composed to tackle growing urbanization challenges. A smart city consists of billions of IoT devices that utilize technology in daily tasks like smart cooking, smart lighting, smart vehicles, traffic management, water treatment and supply, trash management, power grid management, etc. The ultimate goal of developing a smart city is to develop a sustainable and technology-enabled infrastructure. The IoT plays a considerably major role in achieving this goal by enabling communication and data sharing among the devices. The IoT devices collect data using the built-in sensors and store it on the cloud for further usage.

Utilizing IoT to such an enormous extent exposes its vulnerabilities that could easily be utilized to gain control over the network and thus posing serious risk to residents and authorities. Among a multitude of cyber-attacks, DDoS attacks appear to be most threatening. The smart cities are based on a centralized infrastructure, and a high magnitude DDoS attack to the central command of a smart city can wreak havoc and cause catastrophic damage to devices, residents, and eventually to economy. A smart city offers an overabundance of devices, e.g., traffic lights, that when recruited to a botnet, can increase the severity of attack many folds.

3.17 DDoS Attacks in Mobile Applications

The mobile applications are also highly susceptible to a diverse range of cyber-attacks. The DDoS attacks in mobile applications are major cause of concern for two reasons: first, profiling application users is fairly easy for an attacker through their mobile devices and hence many mobile apps like Facebook, LinkedIn, Ola, Airbnb, Uber, etc. are highly vulnerable to such infections. Second, the mobile applications are quite easy to create and relatively less secure and thus plenty of e-commerce and online payment apps, containing sensitive information, are susceptible to such malware. Unfortunately, there is no way to differentiate between a legitimate app and a malicious one. Once such malevolent application is downloaded in the mobile device, the application may open up some vulnerabilities in the device by installing backdoors that can be used to command and control it. Once the device is infected by the attacker, it could be under attack or be used to carry out attack on other devices or networks as a bot. Figure 7 demonstrates the formation of a mobile botnet using a malicious application. As many as half a million infected mobile devices can be utilized in executing an attack on a target, raising the number of requests generated per second to multi-millions. Therefore, the mobile devices are easy targets to infect, control and propagate malware across other mobile devices via malicious mobile applications.

Figure 7. Botnet formation using mobile applications



4. TAXONOMY OF DDOS DEFENSE MECHANISMS

The development of a strong DDoS defense mechanism is challenging because it should not only detect the attack before it happens, but should also respond to an ongoing attack efficiently. There are multiple challenges that are encountered to develop such defense mechanisms, like lack of detailed information about the attack such as packet rate, packet size, duration of attack, quantum of damages, etc. and the need for strong collaboration among networks. These challenges, if not dealt with, result in a weak defense mechanism and thus allow DDoS attacks to lead disastrous consequences depending on the attack domain and the severity of attack. To deal with such humongous and deleterious effects of DDoS attacks, multiple defense mechanisms based on different research recommendations have been proposed in the literature. Table 2 presents a comparison of multiple related works with our work in terms of defense approaches, based on various modern technologies, included in the literature. In this section, we outline some of the promising techniques and how they could be used for effective DDoS detection and mitigation.

4.1 DDoS Detection and Mitigation, Based on Traditional Methods

In this section, we briefly cover some early DDoS defense approaches that laid the foundation upon which the modern detection and mitigation techniques now stand.

Ingress/Egress packet filtering (Aamir & Zaidi, 2014; Peng et al., 2007; Zargar et al., 2013) is one of the early countermeasures against DDoS attacks, deployed at the edge routers of the networks. The Ingress filtering involves monitoring the packets arriving in the network and the Egress filters examine outbound packets originating within the network. Similarly, the Ingress filtering permits only trusted networks to send traffic inside a network, while the Egress filtering stops the attacks emerging from a compromised device within the network. If the packets are routed using illegitimate address, the Ingress/Egress filters drop them.

Another packet filtering technique is Hop Count Filtering (HCF) (Aamir & Zaidi, 2014; Wang et al., 2007; Zargar et al., 2013). In HCF, incoming packets are scanned for their Time-To-Live (TTL) field in the IP header and an IP-to-Hop-Count (IP2HC) mapping table is maintained by the victim router, that contains the information about the hop counts for multiple IP addresses. The TTL value of a packet gets decremented by 1 as it traverses through each router, and the victim router is only able to see the final TTL value. The router reads from each incoming packet the final TTL value along with the source IP address. An initial TTL value is predicted, and the difference between the initial and final TTL value provides the final hop count. If the final hop count matches the hop count value corresponding to the source IP address in the IP2HC table, the packet is validated, else the packet is dropped.

The IP Traceback is also a well-known approach that has been widely used to determine the actual address of the attacker. Several papers have proposed different mechanisms for IP traceback (Aamir & Zaidi, 2014; Peng et al., 2007; Zargar et al., 2013). Chae et al. (2007) put forward a structure of

Table 2. Comparison of related works in terms of various computing platforms

Contributions	Traditional Environment	Cloud Computing	Internet-of-Things	SDN	Machine Learning	Deep Learning	Big Data Analytics	Blockchain Technology
Agrawal & Tapaswi (2019)		✓						
Džaferović et al. (2019)			✓					
Asosheh & Ramezani (2008)					✓			
Taj & Khalil (2018)	✓							
Saad et al. (2018)								✓
Masdari and Jalali (2016)		✓						
Xu et al. (2019)							✓	
Yan and Yu (2015)		✓		✓				
Eliyan & Di Pietro (2021)				✓	✓			
Munshi et al. (2020)			✓					
Yuan et al. (2017)						✓		
Wen et al. (2021)								✓
Douligeris & Mitrokotsa (2004)	✓							
Aladaileh et al. (2020)				✓				
Specht & Lee (2003)	✓							
Farahmandian et al. (2013)		✓						
Li et al. (2018)				✓		✓		
Cheng et al. (2018)							✓	
Our work	✓	✓	✓	✓	✓	✓	✓	✓

agent systems, containing IDS which detects a malicious packet and alerts the server system about the attack by embedding the packet in an ICMP Traceback (iTrace) message. Xiang & Zhou (2005) proposed a packet marking mechanism that tags the incoming packets with a mark that contains the source IP address at edge ingress routers and thus eliminating the possibility of mark-spoofing. The

mechanism also allocates a segment number to each mark that help in reconstructing the source IP address of the packet.

4.2 DDoS Detection in a Cloud-Based Environment

As explained in Section 4.2, the DDoS attacks in a cloud-based environment could end up having some disastrous consequences. Thus, many proposals exist to handle it. Table 3 presents a comparison of multiple DDoS defense approaches in cloud computing along with their strengths and weaknesses. Joshi et al. (2012) proposed a Cloud Trace Back (CTB) model for attack detection, along with a trained back propagation Neural Network (NN), called Cloud Protector defense system, for attack mitigation. Being located before the Web Server, any service request passed on to the server initially passes through the CTB, positioned at the edge routers, where it is marked with a Cloud Trace Back Mark (CTM) tag and then sent to the actual server. In case of an attack, a reconstructed CTM tag can pinpoint the attack source and the Cloud Protector detects and filters out any malicious packet from that address. The results show a reasonable detection rate based on training and test data sets.

Agrawal and Tapaswi (2017) have presented a lightweight approach in order to detect and filter out spoofed DDoS attack packets. The approach is based on two major assumptions: first, the firewall present at the entry point of the cluster is already configured to detect and filter out the previously blacklisted IP addresses and therefore only the traffic from legitimate and spoofed IP addresses is passed on to the detection system. Second, the behavior of attack traffic differs from that of normal packets in terms of flow count i.e., the number of packets. Based on above two assumptions, filtered traffic is captured by Wireshark and the corresponding packet count is utilized in detecting and mitigating the attack.

Zhao et al. (2009) proposed a novel approach utilizing a Virtual Machine Monitor (VMM) containing a detector, tagger, and a duplicator program. In an attack scenario, as and when the number of available resources reach below threshold, the VMM detects an attack and duplicates the operating system as well as the tagged applications to an isolated environment and hence it successfully breaks out of a DDoS attack without crashing down. One major limitation of this approach is the resource wastage when no attack is present. The isolated environment is created however remains idle until an attack is detected.

A solution to the shortcomings of the cloud technology is addressed by Edge Computing. It is a distributed architecture that pushes some amount of memory and processing power to the edge of the network, close to the source of data, instead of a cloud-based environment. Edge computing boosts real-time results by minimizing latency and bandwidth consumption since the data has to be sent to the edge of the network, instead of all the way to the cloud. He et al. (2021) proposed a game-theoretical approach to mitigate DDoS attacks in the edge servers by finding sub-optimal solutions to large-scale DDoS mitigation problems, called Edge DDoS Mitigation Game (EDMGame). EDMGame simulates each request generated by the users in an edge network, as individual players, allocating specific edge

Table 3. Comparison of related works in Cloud Computing

Contributions	Proposed Detection Approach	Strength	Weakness
Joshi et al. (2012)	Packet Marking and Neural Networks	High accuracy	Data-intensive
Agrawal & Tapaswi (2017)	Flow Rate Monitoring	Lightweight	Based on multiple assumptions
Zhao et al. (2009)	Available Resource Monitoring	Continuous execution	Resource wastage
He et al. (2021)	Edge Computing and Game-theory	Minimal latency and computation cost	High complexity

servers for computation by finding a mitigation strategy that contains one allocation decision for each request and maximizing the benefit. EDMGame uses an algorithm that, on completing, results in a final mitigation strategy constituting all individual allocation decisions made in parallel.

4.3 DDoS Detection Using Software-Defined Networking

Software-Defined Networking (SDN) is an emerging paradigm that attempts to centralize the currently decentralized system as a means to upgrade network performance. SDN utilizes software applications to program the network with the aim to control the network intelligently, thus making network troubleshooting easier. Several researchers have moved towards employing SDN for DDoS attack mitigation. Table 4 presents a comparison of multiple DDoS defense approaches in SDN-based environment along with their strengths and weaknesses. SDN and DDoS attacks have a contrary interconnection with each other. While separating the data plane from control plane increase the chances of DDoS detection, it also introduces new attack dynamics with SDN being prone to DDoS attacks as well.

Hong et al. (2017) came up with a Slow HTTP DDoS Defense Application (SHDA) that utilizes SDN controllers to subdue a Slowloris or Slow HTTP POST attack. It starts a timer as soon as the first partial packet arrives after a threshold number of concurrent connections are established with the server. In case the timer expires before completing the request, SHDA blocks the packets and terminates the connection, notifying the server about the source IP address. One limitation of SHDA is the false alarms generated in case of slow users that complete the requests after the timer expires, misidentifying them as attackers.

Thomas & James (2017) have presented another approach for detecting DDoS attacks utilizing a third-party traffic monitoring application in SDN, called Iftop. Iftop analyzes the traffic for a specific period of time in order to observe the bandwidth and source addresses of incoming packets. After the evaluation, if the DDoS attack conditions, based on the throughput of the client, are met, the traffic is classified as malicious and sent to the SDN controller for a detailed analysis to reduce false alarms. The attack packets are dropped and the source address is blocked by the firewall to prevent any further transactions.

Sambandam et al. (2018) brought forward an entropy-based detection of DDoS attack in an IoT network using Raspberry Pi. Entropy defines the randomness of traffic in a network. In case of a DDoS attack, entropy drops significantly since majority traffic arrives from a handful of attack sources. Sambandam et al. (2018) monitored the entropy level of the network with every incoming packet and during an attack, a substantial increase in packet rate drops the entropy below threshold value, notifying the occurrence of a possible DDoS attack.

Another method to detect DDoS attacks based on incoming traffic behavior is presented by Abdulkarem & Dawod (2020) utilizing an SDN application developed with a Python script. The

Table 4. Comparison of related works in SDN environment

Contributions	Proposed Detection Approach	Strength	Weakness
Hong et al. (2017)	Request Completion Timer	Dynamic flow update	Expensive SDN switches
Thomas & James (2017)	Bandwidth and Throughput Monitoring	Low performance overhead	Vulnerable SDN controller
Sambandam et al. (2018)	Entropy-based Detection	Timely detection	Vulnerable to slow attacks
Abdulkarem & Dawod (2020)	Ordered Flow Monitoring	Early detection	Vulnerable to slow attacks
Bhushan & Gupta (2018)	Flow Rate Probability	Low computation overhead	Expensive SDN switches

proposed solution utilizes Open vSwitches in an SDN architecture to detect abnormal traffic behavior at the earliest possible stage. SDN controller extracts IP address of the biggest data source, sending huge volumes of data to the server. The switches implement a packet filtering rule in order to drop the malicious packets and let the normal traffic smoothly reach the server.

Bhushan & Gupta (2018) proposed a mechanism based on probability distribution of flow rule hit count in the absence of DDoS attack and maintaining a flow lookup table for packet forwarding. This is achieved using a counter field in the flow entry which gets incremented as the packet traverses through the network. The count of DDoS attack packets is usually higher than normal traffic. Probability distribution of incoming traffic flow rate is calculated when it surpasses the threshold and compared with the probability distribution calculated with no attack. A difference between the values, higher than a specific threshold value, is considered as a notification of potential DDoS attacks and mitigation scheme is activated, dropping all requests arriving from the attacker.

4.4 DDoS Detection Based on Machine Learning

Machine Learning is coming across as one of the major approaches to overcome DDoS attacks in recent years. Several literatures propose DDoS detection mechanisms based on machine learning approaches like Naïve Bayes, K-Nearest Neighbors (KNN), Random Forest, Decision Tree, Fuzzy Logic, etc. Table 5 presents a comparison of multiple DDoS defense approaches that utilize machine learning algorithms along with their strengths and weaknesses. Dong & Sarem (2019) proposed a DDoS Detection Algorithm based on Machine Learning (DDAML) that makes use of an improved K-Nearest Neighbors algorithm to identify the malicious data packets in an SDN environment. Another research by Fouladi et al. (2016) proposes a monitoring system based on packet sampling mechanism. It creates two metrics based on Distant Fourier Transform (DFT) and Distant Wavelet Transform (DWT) and a Naïve Bayes classifier uses them as features to identify an attack.

Vishwakarma & Jain (2019) presented a machine learning centric approach to detect botnet-based DDoS attacks in an IoT environment. The proposed system utilizes IoT honeypot devices to fascinate the attackers into targeting the vulnerabilities and generate a log file of all the extracted information about the attack, like the type of malware, type of application or protocol it targets, C&C server, port number, etc. Based on the log file, a real-time implementable machine learning model is trained and accurately classifies the malware families based on its features. The only drawback to

Table 5. Comparison of DDoS defences utilizing Machine Learning

Contributions	Proposed Detection Algorithm	Strength	Weakness
Dong & Sarem (2019)	K-Nearest Neighbours (NN)	High detection rate	Unknown dataset
Fouladi et al. (2016)	Naïve Bayes	Minimal implementation complexity	Speed-accuracy Trade-off
Vishwakarma & Jain (2019)	IoT Honeypots	Tackle unknown attacks	High implementation cost
Lin et al. (2015)	k-means Clustering and NN	Less computational effort	Low accuracy for R2L attacks
Sudar et al. (2021)	Decision Tree and SVM	High accuracy and less complexity	Expensive SDN switches
Jia et al. (2017)	Multi-classifier Ensemble Model	Heuristic detection	Implementation complexity
Alsirhani et al. (2019)	Fuzzy Logic System	High accuracy and low latency	Inconsistent performance

this approach is the implementation cost incurred. The classifier has to be implemented on each IoT device of the network separately.

Lin et al. (2015) proposed an Intrusion Detection System (IDS) based on a novel approach, called Cluster Center and Nearest Neighbors (CANN), by combining two well-known machine learning algorithms. In CANN, two distances for each data point from training and testing data sets are calculated. One, from data point to the cluster center and the other, from data point to the nearest neighbors in the same cluster. The aggregate value of these two distances results in a new feature value for training and test set which is utilized by the IDS for DDoS detection.

Sudar et al. (2021) presented a DDoS detection method utilizing machine learning algorithms along with SDN-based architecture. The proposed method employed highly accurate and significantly less complex machine learning algorithms, called Decision Tree (DT) and Support Vector Machine (SVM) for the classification of incoming data traffic into normal or attack. After the model training and feature extraction phases, SVM and DT classify the dataset as malicious if the flag value comes out to be 1. In such cases, the Open vSwitches notify the SDN controller to update the flow table and drop the attack traffic.

Jia et al. (2017) put forward an innovative detection mechanism by combining various component classifiers based on Singular Value Decomposition (SVD) and Rotation Forest Method (RFM). It involves a voting system that outputs the final classification mechanism for attack detection. Alsirhani et al. (2019) utilized a similar approach by employing a Fuzzy Logic system to yield a dynamic classification algorithm which is used to detect DDoS attack traffic from normal traffic. Only a single classification algorithm, out of Naïve Bayes, Entropy-based Decision Tree, Gini Decision Tree, and Random Forest algorithm, is used at any specific time for classification of incoming data traffic.

4.5 DDoS Detection Based on Deep Learning

Recently, the deep learning has emerged as a popular class of machine learning though it was first proposed way back in 2009 by Hinton (2009). Table 6 outlines a comparison of various DDoS defense mechanisms that rely on deep learning approaches along with their relative strengths and weaknesses. The deep learning can be considered as a powerful extension of the conventional machine learning approaches that utilizes Artificial Neural Networks (ANN) to imitate the working of human brain. The major benefits of Deep Learning, which have made it a preferable research option over the traditional machine learning models, are the maximal utilization of unstructured data, improved quality results, and the elimination of feature engineering as well as data labelling.

Li et al. (2018) came up with a DDoS defense architecture centered on Deep Learning that utilizes Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN) in SDN-based environment. In this approach, after constructing a feature matrix of all the features extracted from data packets of OpenFlow switch, a detector module determines whether

Table 6. Comparison of DDoS defences utilizing Deep Learning and Neural Networks

Contributions	Proposed Detection Approach	Strength	Weakness
Li et al. (2018)	RNN, LSTM, CNN and SDN	High detection accuracy	Expensive SDN switches
Yuan et al. (2017)	RNN using Historical Data	Minimal error rate	Complex training
Roopak et al. (2019)	MLP, CNN, LSTM, CNN+LSTM	Minimal supervision	Resource-intensive
Doriguzzi-Corin et al. (2020)	CNN	Low computation overhead	Time-intensive training
Hussain et al. (2020)	CNN using Real Network Data	Lightweight	Wrong Instantaneous attacks detection

the data packets are malicious in nature or not. If so, a statistics module examines the frequency of all features and establishes a weight, according to which a flow entry is recorded in the flow table. The OpenFlow switch deals with the attack packets as specified in the table.

Yuan et al. (2017) used another Deep Learning defense approach, called DeepDefense, that creates a DDoS detection system using RNN and CNN. The detection system utilizes the historical information of data packets to determine their legitimacy. For detecting and locating the repeated patterns in the incoming data that represents a DDoS attack, this historical information is fed to the RNN and attack is detected.

Roopak et al. (2019) came forward with an approach to detect cyber-attacks in IoT networks using four different Deep Learning classification models: Multi-Layer Perceptron (MLP) model, CNN model, Long-Short Term Memory (LSTM), and CNN+LSTM hybrid model. These approaches are highly resource-intensive and unpractical to be deployed into real world.

Doriguzzi-Corin et al. (2020) proposed LUCID, a lightweight Deep Learning technique that utilizes CNN to differentiate malicious traffic from safe traffic. After usual computation of CNN like preprocessing, feature extraction, padding, and normalization, the output of CNN is passed through a sigmoidal function, constraining the final output to 0 and 1. The data flow is considered safe if output is below 0.5.

Another Deep Learning approach by Hussain et al. (2020) employs CNN for early detection of DDoS attacks in a 5G cellular network. The model assembles already-available call detail record (CDR) data containing three activity values: outgoing calls, outgoing SMS and Internet usage, associated with every cell in the network. The CNN detects the legitimacy of new incoming traffic based on training process over past CDR data.

4.6 DDoS Detection Based on Big Data Analysis

Big data analysis refers to the comprehensive analysis of large amounts of data to identify recurring patterns and interconnections. Big Data Analysis takes faster and better decisions and a considerably low cost for storing large volumes of data, making it a quite advantageous approach for DDoS detection. Several novel Big Data centered approaches have been proposed by researchers in past few years. Table 7 presents a comparison of multiple DDoS defense mechanisms that utilize approaches based on big data analytics, along with their strengths and weaknesses.

Hameed & Ali (2016) have proposed a Hadoop based approach for DDoS flooding attack detection, called HADEC, by applying Hadoop Distributed File System (HDFS) and MapReduce. HADEC consists of a capturing server and a detection server. After the capturing server captures live traffic based on the parameters configured by the admin, the results are stored in a log file which is sent to the detection server. The detection server saves the file in HDFS and applies MapReduce based detection algorithm before notifying the administrator about the results.

Mizukoshi & Munetomo (2015) came up with a Hadoop based clustering approach utilizing Genetic Algorithm (GA) and entropy-based DDoS detection. In this approach, the incoming network

Table 7. Comparison of related works utilizing Big Data Analytics

Contributions	Proposed Detection Approach	Strength	Weakness
Hameed & Ali (2016)	HDFS, MapReduce	Scalable	Speed-time trade-off
Mizukoshi & Munetomo (2015)	Genetic Algorithm	Scalable and adaptive	Dataset dependent result
Xu et al. (2019)	Deep Forest Model	High accuracy	Slow real-time prediction
Cheng et al. (2018)	Detection from Historical Data	Low computation complexity	No universal time interval

traffic is captured and examined by two different modules. The module containing Genetic Algorithm develops a packet profile by detecting the features of incoming packets and stores the profile in the DDoS filtering rule base module. The entropy-base module detects DDoS attack by analyzing the frequencies of packet source IP addresses. In case of an attack, the DDoS filtering rule base module is notified, which detects attack packet features and filters out packets arriving from malicious clients, thus blocking a DDoS attack.

Xu et al. (2019) proposed a detection method for DRDoS using Deep Forest model in a Big Data environment. The model utilizes statistical information of DRDoS attack flow. Based on this information, a Host based DRDoS Threat Index (HDTI) is created. Using the HDTI, each IP address in the network flow is classified into one of the four categories: normal, upstream, downstream, or mixed upstream and downstream (MUD). The data packets coming from normal IP addresses are allowed to pass. All the service requests coming from upstream identified IP addresses and all the service responses sending to the downstream identified IP address are filtered out. If the IP address is recognized as MUD, all the request and response packets for that IP are filtered.

Cheng et al. (2018) proposed a prediction approach where the model is trained based on normal and attack traffic extracted from the network. Based on the training, a network flow abnormal index value is created to detect the attack flow in the network. For a certain time, interval, a feature value, called the PDRA, is calculated based on some parameters that includes number of new users, average accessing rate of each new user, number of old users, etc. The PDRA is passed on to the classifier which classifies the traffic into one of the three categories: normal, hotspot event, or DDoS attack.

4.7 DDoS Detection in Blockchain

The blockchain has grabbed the attention of both, attackers and researchers owing to its decentralized infrastructure and consensus-based mechanism. A flurry of papers has been published highlighting new form of potential attacks in a blockchain-based environment. The growing concern due to these evolved attacks has inspired several researchers to come up with effective countermeasures in order to mitigate the attack and prevent the blockchain from crash (Wen et al., 2021). Table 8 presents a comparison of various DDoS defense mechanisms that successfully prevent, detect, and mitigate the potential attack scenarios in blockchains, along with their strengths and weaknesses.

Wu et al. (2020) addressed the dynamic nature of mining pools and proposed a novel approach based on Game-Theory to mitigate a mining pool DDoS attack. Mining pools refers to a consortium of miners that are unable to find a block individually, pooling their individual limited amount of mining powers and distributing the reward proportionally. To detect an attack, Wu et al. (2020) proposed

Table 8. Comparison of DDoS defences in Blockchain-environment

Contributions	Proposed Detection Approach	Strength	Weakness
Wu et al. (2020)	Game-theory and Nash Learning	Higher mining payoff	Theoretical Analysis
Saad et al. (2018)	Fee-based Approach	Increased attack cost	Affects both legitimate/ illegitimate users
Saad et al. (2019)	Age-based Approach	Reduced attack time window	Fast transactions cannot be verified
Mirkin et al. (2020)	Uncle Block Mechanism	Expected profit not reduced	Ineffective against consensus blockchains
Abou El Houda et al. (2019)	SDN, Ethereum Smart Contracts	Flexible, secure and low effective	Cost-intensive implementation

a game-model defined by a random probability distribution. A Nash learning algorithm is used to define the competition among multiple mining pools as random game.

Saad et al. (2018) came up with a fee-based solution to detect and prevent the mempool flooding DDoS attacks. In the fee-based system, a minimum relay fee is charged to every incoming transaction at the memory pool, until a threshold mempool size is reached. Post that threshold, a minimum mining fee is imposed alongside the relay fee to be accepted in the mempool queue. Charging these fees guarantees the elimination of invalidated transactions that are generated just to fill the mempool. Further steps can also be taken to prevent the attacks, like increasing the minimum fee and prioritizing the mempool queue according to the paid fee.

Another solution presented by Saad et al. (2019), is an age-based solution that prevents the flooding of mempool. The proposed solution is based on the assumption that the attacker has paid both the mining and relay fees to get the malicious transaction accepted. Instead of accepting transactions in the mempool first and then discarding it after miners eventually rejects them, a minimum age criterion is applied for getting accepted into the mempool. The average age of all new transactions is calculated with the help of all their parent transactions. Without having a minimum number of confirmations or "minimum age limit", no transaction is accepted in the mempool. The only way of getting accepted in the mempool requires the attacker to get the transactions verified and wait for them to acquire sufficient confirmations.

Mirkin et al. (2020) demonstrated some countermeasures for mitigating BDoS attacks. Except Bitcoin, BDoS attacks in other contemporary blockchains, like Ethereum, can be prevented using Uncle Block mechanism which rewards the miners regardless of whether they are the first ones to mine it. This eliminates the reduction in profitability, a key element for BDoS attacks, that discourages the rational miners from mining. Another means to thwart BDoS attacks is by defining a time interval between reception of the block header and the reception of the actual block. Note that in case of an attack, only the block header is provided. If the actual block is not presented before the timeout expires, the header is not considered as a potential block.

Abou El Houda et al. (2019) proposed Cochain-SC, utilizing SDN and Ethereum smart contracts, which involves detection and mitigation of DDoS attacks on both, inter-domain and intra-domain levels. Cochain-SC involves strong inter-domain collaboration between multiple domains using smart contracts which contains different collaborators and an intra-domain model containing an entropy-based scheme for detecting the attack, a Naïve Bayes-based scheme for classifying the detected traffic as malign or benign and a mitigation scheme for dropping the illegitimate traffic. Once the attack is mitigated, the list of malicious IP addresses is updated and shared with all the collaborators in the smart contract.

5. PERFORMANCE EVALUATION METRICS

In order to determine the nature of incoming data traffic, a standard set of indices are necessary to evaluate the performance of DDoS detection and mitigation systems. Table 9 presents a detailed comparison of datasets and performance metrics utilized by multiple defense approaches proposed in multiple literatures over the years. Over the years, the following characteristics are admitted by multiple researchers (Bhuyan et al., 2015; Mirkovic et al., 2006; Mirkovic & Reiher, 2004; Mölsä, 2005; Zargar et al., 2013) as vital for a detection mechanism to run effectively and efficiently:

1. **Strength:** The strength of a DDoS detection system is of paramount importance for achieving effective results. The defense strength of a detection and mitigation system is defined by the ability of the system in efficiently preventing an attack from taking place, detecting attacks in their early phases, and mitigating an existing attack as quickly as possible. Strength of a system

Table 9. Datasets and performance metrics utilized in related works

Contributions	Dataset Used	Performance Metrics
Abadeh et al. (2007)	DARPA 1998	DR, FPR
Baig et al. (2013)	KDD-Cup 1999	Accuracy, TPR, Precision, FPR, FNR, ROC Curve
Chen et al. (2007)	DARPA 1998	FPR, FNR
Feng et al. (2014)	KDD-Cup 1999	DR, FPR, FNR
Eesa et al. (2015)	KDD-Cup 1999	DR, FPR, Accuracy, ROC Curve
Hoque et al. (2017)	CAIDA DDoS 2007	TNR, TPR, Accuracy
Shawahna et al. (2018)	NA	DR, Cost
Fouladi et al. (2016)	NA	TNR, TPR, Accuracy
Sarasamma et al. (2005)	KDD-Cup 1999	DR, FPR
Zhang et al. (2005)	KDD-Cup 1999	DR, FPR
Liu et al. (2007)	DARPA 1998	DR, FPR
Hu et al. (2008)	KDD-Cup 1999	DR, FPR, Run Time
Tong et al. (2009)	DARPA 1998	DR, FPR
Wang et al. (2010)	KDD-Cup 1999	Precision, TPR, F-measure
Shon & Moon (2007)	DARPA 1998	DR, FPR, FNR
Xiang et al. (2008)	KDD-Cup 1999	DR, Run Time
Somani et al. (2017a)	NA	DR, Service Downtime, Service Response Time
Li & Guo (2007)	KDD-Cup 1999	TPR, FPR
Khan et al. (2007)	DARPA 1998	FPR, FNR, Accuracy
Somani et al. (2017b)	NA	DR, Service Response Time, Service Downtime
Hansen et al. (2007)	KDD-Cup 1999	DR
Wu et al. (2019)	NA	TNR, TPR, Accuracy
Sangkatsanee et al. (2011)	KDD-Cup 1999	DR
Wei et al. (2008)	CAIDA ITDK	Convergence, Stability
Perez-Diaz et al. (2020)	CAIDA DDoS 2007	Accuracy, FPR, F-measure, Precision, TPR

is measured by multiple indexes that depend on the assumptions made by the detection system, resulting in four possible results:

a. *True Negative (TN)* results are attained if all the normal data records are correctly classified as safe traffic.
b. *True Positive (TP)* results are attained if all the attack data records are correctly classified as malicious traffic.
c. *False Positive (FP)* results are attained if all the normal data records are incorrectly classified as malicious traffic.
d. *False Negative (FN)* results are attained if all the attack data records are incorrectly classified as safe traffic.

Based on these outcomes, Stehman (1997) has introduced six primary indices for performance evaluation:

- Accuracy is defined as the ratio of all the correct results of defense mechanism to total results of the defense mechanism:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} *100$$

- Precision is defined as the ratio of true positive results to all the results classified as positive by the defense mechanism:

$$Precision = \frac{TP}{TP + FP} *100$$

- Reliability is defined as the ratio of false positive results to all the results classified as positive by the defense mechanism:

$$Reliability = \frac{FP}{FP + TP} *100$$

- Sensitivity is defined as the ratio of true positive results to total actual positive results:

$$Sensitivity = \frac{TP}{TP + FN} *100$$

- Specificity is defined as the ratio of true negative results to total actual negative results:

$$Specificity = \frac{TN}{TN + FP} *100$$

- False Negative Rate is defined as the ratio of false negative results to all the results classified as negative by the defense mechanism:

$$False\ Negative\ Rate = \frac{FN}{FN + TP} *100$$

2. **Scalable and Adaptive:** The defense mechanism needs to be scalable as the number of users, both legitimate and illegitimate, increases in order to successfully monitor the incoming traffic. Apart from the users, if the attack traffic increases significantly in volume, the mechanism also needs to adapt to such conditions. A non-scalable and non-adaptive defense mechanism may present a potential attack avenue for the attackers by suffering from bottleneck of growing user demands.

3. **Quick:** Detecting an attack in its early phase is another primary feature that, a defense mechanism needs to have. The defense mechanism needs to validate the incoming data traffic as soon as possible to reduce the response time and upgrade user experience.
4. **Accuracy:** Accuracy of a defense mechanism is defined by its ability to correctly differentiate attack traffic from normal data traffic. Accuracy is measured by the amount of attack traffic correctly recognized by the firewall i.e., true positive rate, and the amount of attack traffic it passed on to the server i.e., false negative rate.
5. **Service Response Time:** The period between a service request is sent by the users to the server and the response received by the them defines the service response time. An efficient defense mechanism needs to monitor the traffic swiftly and reduce the response time as much as possible to improve user experience.
6. **Simple Implementation:** For an effective defense mechanism, it is an important characteristic to have a simple implementation. The implementation needs to be non-complex as well as realistic i.e., the approach needs to be effective with large-scale, real-time network traffic instead of being effective only in a simulated environment. Feasibility is a predominant metric while implementing a defense mechanism.
7. **Low Computation Overhead:** While scanning the network for potential attacks, if the defense mechanism suffers from high computational overheads, the user experience can be severely degraded. As mentioned above, the occurrence of bottleneck at the firewall can introduce new methods for attacking the target.
8. **Proactive or Reactive:** Another predominant characteristic for a defense mechanism, is being proactive instead of reactive. Proactive approach is defined by a system's ability to detect DDoS attacks in their early stages, while reactive measures try to mitigate the attack after it is successfully executed.
9. **Cost:** One of the most significant performance metrics, is the cost incurred while detecting and mitigating an attack. For an effective defense system, this cost needs to be less than the amount of losses caused by the DDoS attack. The cost of handling an attack is determined by multiple factors, like bandwidth, available resources, computation, memory storage, etc.

6. TOOLS FOR PERFORMING DDOS ATTACKS

In this section, we briefly discuss some of the well-known and commonly utilized tools for carrying out DDoS attacks. Apart from the Slowloris and RUDY attack tools discussed in Section 3.4, myriads of tools, with a diverse range of severity and impact, are available on the Internet today for executing DDoS attacks against a target server (Behal & Kumar, 2017; Hoque et al., 2014). Table 10 presents a detailed analysis and comparison of various functionalities provided by DDoS attack tools that are included in this exposition. It is important to note that all DDoS tools are not harmful in nature. Based on the reason behind their development, a DDoS tool can be classified into different categories, like tools for attacking a specific target, tools for testing a network for potential vulnerabilities, or defensive tools for detecting attacks by network monitoring:

1. **LOIC:** Low Orbit Ion Canon or LOIC (Praseed & Thilagam, 2018) is an open-source DDoS attack tool which is highly prevalent, highly simplified and easily accessible on the Internet. Once started, LOIC can be utilized to establish multiple connections with the target server and eventually, carry out TCP, UDP, or HTTP attacks. The attackers do not require any prerequisite technical knowledge to use LOIC for DDoS attack purposes as long as they know is the IP address/ URL of the website. The HIVEMIND mode of LOIC can be utilized to control remote systems to execute an attack using a voluntary botnet. The only major drawback of LOIC is that it does not hide the attackers' IP address since the original purpose behind the development of LOIC was stress testing one's own server or website. After being utilized in perpetrating some major

cyber-attacks like Project Chanology in 2008 (Singel, 2008) and Operation Payback (Addley & Halliday, 2017) by the notorious hacktivist group Anonymous, LOIC has gained notable importance as a favorable tool for constituting DDoS attacks.

2. **HOIC:** High Orbit Ion Canon or HOIC is another open-source, easy to use DDoS attack tool available legally as a stress testing tool. Developed by the hacktivist group Anonymous, HOIC is the successor of the previously discussed LOIC and can have destructive consequences caused by an attacker with limited or even no technical knowledge. HOIC is similar to its predecessor in carrying out attack with minimum parameters and provides a traffic speed controller to masquerade attack traffic as normal network traffic. Another similarity to LOIC is that HOIC also does not hide the attack source address. Apart from all the similarities, unlike LOIC, HOIC can only be utilized to carry out HTTP GET and POST attacks and not TCP and UDP floods. Another major benefit of using HOIC is that it can attack up to 256 different domains simultaneously allowing as few as 50 attackers to manually coordinate and perpetrate a serious DDOS flooding attack against a single target. HOIC was first revealed as a DDoS attack tool during Operation Megaupload (PCMag, 2010) against multiple websites including FBI and the Justice Department of US carried out by Anonymous in 2012.
3. **XOIC:** XOIC is a DDoS attack tool that was created as a copy of LOIC. It has an easy-to-use GUI which allows the attackers to carry out DDoS attacks on target IP addresses by specifying the port number and the protocol. The XOIC can carry out IRC-based DDoS attacks using HTTP/UDP/TCP/ICMP packets. It has three attack modes: test mode, basic DoS attack mode, and Dos attack with HTTP/UDP/TCP/ICMP messages. It is an efficient attack tool for executing attacks against small websites. One major drawback of XOIC is that the attack perpetrated by it is easily detectable and therefore could be blocked.
4. **Hping3:** Hping3 is a TCL-based DDoS attack tool that could be used for various attacking as well as testing purposes, like scanning the devices for open ports and vulnerabilities, testing the efficacy of network firewall for multiple attack scenarios, etc. The hping3 provides the ability to created malformed TCP/IP packets with spoofed IP addresses. The spoofed IP address could either be a fake one or a legitimate IP address of any other device, including the target itself. Apart from IP spoofing, hping3 also allows attacker to modify the attack traffic according to size i.e., fragmentation of packets into arbitrary sizes. Once the destination address is registered, the attacker can determine the desired attack volume and hping3 begins to strike the specified IP address with manipulated attack traffic.
5. **DDoSIM:** As the name suggests, DDoS Simulator or DDoSIM (Praseed & Thilagam, 2018) is another attack tool to carry out a protocol-based and application-based DDoS attack. DDoSIM reveals the capacity of the target server or website to handle successful DDoS attacks. After that, DDoSIM replicates multiple zombies with fake random IP addresses to secure the anonymity of the attacker since the attack appears to be constituted by a botnet. Each of these simulated zombies establishes a valid full TCP connection with the target server and floods it with attack traffic on random network ports, once the connection is set up. DDoSIM can successfully perpetrate an HTTP flooding attack with valid as well as invalid service requests along with SMTP and TCP flooding attacks.
6. **DAVOSET:** The DDoS Attacks Via Other Sites Execution Tool, or DAVOSET, exploit the vulnerabilities of various sites in order to execute DDoS attacks on the target site. The Abuse of Functionality and XML External Entity (XXE) are the major vulnerabilities exploited by this PERL-based command line attack tool. The Abuse of Functionality is an attack strategy in which a website's own aspects and attributes are abused to carry out attacks against itself or other websites. The XXE vulnerability is a security flaw that enables attacker to corrupt the XML data processing of a website. Instead of IP spoofing, DAVOSET provides attacker with the ability to create a botnet by generating multiple zombies and command those zombies to carry out HTTP-based attacks.

7. **HULK:** The HTTP Unbearable Load King (HULK) (Praseed & Thilagam, 2018) is an attack tool for web servers, developed for research purposes. To emphasize how straightforward, it is to attack a web server and eventually crash it, HULK was introduced as a proof-of-concept. It is a free, easy to use tool that can generate huge volumes of data traffic towards a web server to paralyze it. The HULK triggers massive floods of HTTP GET requests that are hard to detect and directly hit the resource pool of the server, bypassing the traditional defense mechanisms. A feature of HULK that makes it a strong tool for DDoS attacks is the generation of dynamic requests. For each request, the HULK generates unique headers with invalid and counterfeit fields. Subsequently, it attaches a random user-agent to the request, from a list of user-agents, which hides the request from conventional caching mechanisms. Apart from this, the HULK also comes with a safety option to abort the process and terminate the attack in the middle.
8. **Tor's Hammer:** Tor's Hammer is an application-layer DDoS attack tool that perpetrates slow post attacks. It is a Python-based attack tool that spoofs the source IP address using the Tor network. The attack traffic easily bypasses the server defense mechanisms because it is confused with legitimate traffic owing to the normal rate and low volume. Therefore, it keeps on consuming the server resources and eventually brings down the target. One major drawback of using this tool is that the user interface is not straightforward. The users can launch effective DDoS attacks efficiently only if they possess a little knowledge about this tool.
9. **GoldenEye:** The GoldenEye is a simple yet effective open-source tool for DDoS attacks on web servers. It creates a single zombie that generates a high attack volume using multiple legitimate HTTP requests to the target server. It establishes a valid TCP connection with the server and employs the HTTP KeepAlive messages to prevent server from timing out. It exhausts the resources of the server by consuming all the available HTTP/S sockets by utilizing Cache-Control options in order to disallow socket connections from busting. The attack traffic generated by this Python-based DDoS attack tool is highly randomized by incorporating both HTTP GET and POST requests. This magnifies the complexity of attack detection.
10. **PyLoris:** PyLoris (Praseed & Thilagam, 2018) is a platform independent tool for testing the network vulnerabilities by directly executing a DDoS attack on the service. Unlike other tools setting up TCP connections, PyLoris carries out an attack by utilizing SSL connections and SOCKS proxies. It has an easy-to-use interface where the attack can configure the multiple attack parameters like number of connections, speed of the attack traffic, proxy type, and address of the host etc. Once configured, PyLoris opens multiple connections and keeps them open for as long as server timeout permits, eventually creating a Denial-of-Service condition. PyLoris provides a total of 500 simultaneous connections in the form of 50 threads with 10 connections limit on every thread. Multiple protocols like HTTP, FTP, SMTP, IMAP, and Telnet can be attacked by PyLoris. One major limitation of using PyLoris is the Python dependencies that renders the installation process difficult for users.

7. OPEN CHALLENGES AND FUTURE RESEARCH DIRECTIONS

In this paper, we have presented a broad classification of DDoS attacks and their consequences on various environments along with a brief analysis of multiple defense approaches proposed by researchers. In this section, we discuss some of the major features of these environments that hinders their proper utilization in defensive strategies. We also present some open opportunities for further research and various short-term and long-term goals that we expect to see from researchers and service providers in the future:

1. Accurate detection of an attack is inversely proportional to successful mitigation of the attack i.e., the most accurate detection of an attack can be done when it has already reached its destination target but the best mitigation strategy is the one that stops the attack at its source. For an effective

Table 10. Comparison of DDoS attack tools

Attack Tool	Year	Type of Interface	Type of Attack Traffic	Botnet Formation	IP Spoofing	Language
LOIC	2008	GUI	TCP, UDP, HTTP, ICMP	Yes	No	C-sharp
HOIC	2012	GUI	HTTP	Yes	No	Basic
XOIC	2010	GUI	TCP, UDP, ICMP	Yes	No	C-sharp
Hping3	2005	CLI	TCP, UDP, ICMP	No	Yes	TCL
DDoSIM	2009	CLI	TCP, UDP, HTTP, SMTP	Yes	No	C++
DAVOSET	2010	CLI	HTTP	Yes	No	Perl
HULK	2012	CLI	HTTP	No	No	Python
Tor's Hammer	2009	CLI	HTTP	Yes	No	Python
GoldenEye	2012	CLI	HTTP	No	No	Python
PyLoris	2010	CLI	TCP, UDP, HTTP, SMTP, IMAP, FTP, Telnet	Yes	Yes	Python

and efficient DDoS defense mechanism, the attack needs to be detected as close to its source as possible so that it can abuse as fewer network resources as possible on its way. Cooperation among network nodes is extremely crucial for this to happen.

2. Attackers' incentives also need to be kept in calculations while developing defense strategies. Establishment of stringent laws against cyber-criminals and well-defined cyber-insurance policies can lead to a DDoS defense mechanism incorporating attackers' motivation behind perpetrating cyber-attacks.
3. The extent of collaborative environments like cloud computing (Bhushan & Gupta, 2018; Joshi et al., 2012; Mishra et al., 2021; Xiao & Xiao, 2012; Yan & Yu, 2015; Yan et al., 2015) and IoT (Chui et al., 2019; Cvitić et al., 2021; Evans, 2011; Kolias et al., 2017; Marr, 2021; Munshi et al., 2020; Sambandam et al., 2018; Zargar et al., 2011) present more and more new avenues for attackers to exploit, like the IoT botnets that include dedicated systems as well as mobile devices, simple home appliances, and IoV-based automotive, etc. DDoS-for-hire services have also increased due to such simple targets since these devices lack significantly in terms of security. This amplifies the requirement of more effective and more rugged security protocols that are easy to implement in these devices. On account of such vulnerabilities in these fields, several researches have emerged with novel techniques, like authentication using RFID tags and detection based on relevant packet metadata (Tewari & Gupta, 2020; Vishnoi et al., 2021).
4. The SDN has many characteristics that assist the researchers in developing some strong DDoS detection mechanisms (Abdulkarem & Dawod, 2020; Dharma et al., 2015; Hong et al., 2017; Li et al., 2018; Mishra et al., 2021; Mladenov, 2019; Sun et al., 2019; Thomas & James, 2017; Yan and Yu, 2015), like dynamic updates for flow rule, centralized control of the entire network, software-based analysis, separating data plane from control plane, etc. However, the above characteristics also render SDN a potential target for several types of overloading DDoS attacks (Bhushan & Gupta, 2019), e.g., an attack against the centralized SDN controller can crash the entire SDN-based cloud. Even though some studies have been conducted in this area (Bessani, 2011; Garcia et al., 2011; Giacomoni, 2013; Lee et al., 2014; Wei et al., 2014; Yu et al., 2010), the method of inducting SDN controller in defense mechanism needs further investigation in order to effectively mitigate DDoS attacks.

5. Owing to the centralized problem of SDN-based architecture, blockchain-based infrastructure is becoming increasingly prevalent in recent years with multiple approaches presented in several domains (Fernández-Caramés & Fraga-Lamas, 2018; Karame, 2016; Lee et al., 2016; Noizat, 2015; Perboli et al., 2018; Ron & Attias, 2017; Tschorsch & Scheuermann, 2016), like health care, electronic voting systems, and supply chain, etc. There are some open challenges that needs to be handled for a successful consolidation of blockchains in such delicate environments. A strong and secure consensus mechanism needs to be developed that can surrogate proof-of-work and proof-of-concept (Saad et al., 2020). Improving the security and management of smart contracts with robust technologies, like deep learning, also needs to be explored (Wen et al., 2021).
6. One of the eminent features offered by a peer-to-peer (P2P) network is the non-existence of a centralized authority. This, sequentially presents the network with a robust as well as a highly scalable infrastructure. Though a decentralized architecture augments the security standards of a network due to the lack of any single point of failure, the same could be the reason for P2P-targeted DDoS attacks (Qi & Yang, 2009). The challenge of providing scalability, accuracy and reliability in a distributed environment is still a deterrent in the development of an efficient P2P-based DDoS defense system.
7. The high volumes of structured, semi-structured and unstructured data generated at an express rate renders conventional methods to powerless in terms of management and analysis. Therefore, big data analytics is becoming increasingly appealing in recent times owing to its capacity to perform a detailed analysis on a variety of data (Cheng et al., 2018; Lan & Jun, 2013; Mahmood & Afzal, 2013; Raj, 2014; Singh et al., 2014; Xu et al., 2019). Still unexplored, big data analytics could prove as a silver bullet against mitigating botnet-based DDoS attacks in various environment.
8. The classification algorithms utilized in machine learning have glaringly taken the benchmark for DDoS attack detection to a new level. Though a number of different approaches have been put forward over the years for detecting and preventing DDoS attacks that generate significant results, the domain is still highly unexplored due to its inherent challenges. Training a machine learning model requires substantial amount of resources for operation and generates high temporal cost. Apart from this, high-error susceptibility is present because an extremely accurate and refined dataset is required to properly train the model in order to generate precise outcomes. These challenges, if approached assiduously, can assist in development of a dependable defense system against DDoS attacks.
9. In case of mobile devices, once a malicious application is installed in a device, it is quite easy to infect it and utilize it as a bot due to the vulnerabilities and backdoors opened by the application. Therefore, implementing strong security protocols in order to prevent the attackers from planting such malware-containing applications is of cardinal importance. Even though some novel and reliable defense approaches have been introduced in the recent years (Abbas et al., 2018; Chhabra et al., 2013; Mamolar et al., 2019; Vishnoi et al., 2021), there is still a need to develop some staunch and formidable defensive mechanisms considering the accrescent number of mobile devices. These malware-infected zombies constitute a giant powerful bot army that is capable of bringing down any device or enterprise. A mechanism to detect potential mobile botnets can aid in preventing a multitude of DDoS attacks in environments like cellular networks, mobile devices, wireless networks, smart cities, etc.
10. In the search for new platforms for developing quick and accurate defense mechanisms, multiple new environments are being surveyed, like incentive-based defense mechanism centered on QoS and budget constraints (Dahiya & Gupta, 2019; Dahiya & Gupta, 2021a; Dahiya & Gupta, 2021b), game-theory (He et al., 2021; Michalas et al., 2011; Poisel et al., 2013; Selvi & Shebin, 2016; Shi & Lian, 2008; Sung & Hsiao, 2019; Wu et al., 2020) and deep learning (Doriguzzi-Corin et al., 2020; Hussain et al., 2020; Li et al., 2018; Roopak et al., 2019; Yuan et al., 2017). The inherent drawbacks of these technologies present a layer of complexities that needs to be addressed for achieving our goal. Mani et al. (2021) have discussed a number of different

adversarial attacks on ResNet image recognition deep learning model. Though the results from deep learning models are relatively much better than those from corresponding machine learning models, the remarkably large amount of data required to train a model and the black box nature of neural networks increase both, the computation cost and complexity of the model, rendering it difficult to be implemented by less skilled service providers.

8. CONCLUSION

The raging threat posed by a continual occurrence of DDoS attacks resulting in immense amount of damage and depreciation has burgeoned the need for some reliable and resolute defenses. The development of a unified approach towards tackling such a notorious enemy is of utmost importance. With this goal in mind, this paper aims toward providing helpful insights about DDoS attacks and their consequences in various areas of the Internet, ranging from traditional networks to blockchain-based decentralized environment. In this paper, we have talked about numerous detection schemes proposed by researchers in various domains till date. While these defense approaches yield respectable outcomes, plenty is yet to be explored in order to meet the open challenges faced by these domains. We hope that the work presented here provides a ground-level understanding of the issue that is desired to develop elegant DDoS defense systems.

FUNDING INFORMATION

The publisher has waived the Open Access Publication fee for this article.

REFERENCES

A10. (2020). *The state of DDoS weapons*. A10 Networks. https://www.a10networks.com/wp-content/uploads/A10-EB-The-State-of-DDoS-Weapons-Report.pdf

Aamir, M., & Zaidi, M. A. (2013). A survey on DDoS attack and defense strategies: From traditional schemes to current techniques. *Interdisciplinary Information Sciences*, *19*(2), 173–200. doi:10.4036/iis.2013.173

Aamir, M., & Zaidi, M. A. (2014). *Ddos attack and defense: Review of some traditional and current techniques.* arXiv preprint arXiv:1401.6317.

Abadeh, M. S., Habibi, J., Barzegar, Z., & Sergi, M. (2007). A parallel genetic local search algorithm for intrusion detection in computer networks. *Engineering Applications of Artificial Intelligence*, *20*(8), 1058–1069. doi:10.1016/j.engappai.2007.02.007

Abbas, S., Faisal, M., Rahman, H. U., Khan, M. Z., & Merabti, M. (2018). Masquerading attacks detection in mobile ad hoc networks. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 55013–55025. doi:10.1109/ACCESS.2018.2872115

Abdulkarem, H. S., & Dawod, A. (2020, October). DDoS Attack Detection and Mitigation at SDN Data Plane Layer. In *2020 2nd Global Power, Energy and Communication Conference (GPECOM)* (pp. 322-326). IEEE. doi:10.1109/GPECOM49333.2020.9247850

Abou El Houda, Z., Hafid, A. S., & Khoukhi, L. (2019). Cochain-SC: An intra-and inter-domain DDoS mitigation scheme based on blockchain using SDN and smart contract. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 98893–98907. doi:10.1109/ACCESS.2019.2930715

Addley, E., & Halliday, J. (2017, November 27). Operation Payback cripples MasterCard site for WikiLeaks ban. *The Guardian*. https://www.theguardian.com/media/2010/dec/08/operation-payback-mastercard-website-wikileaks

Agrawal, N., & Tapaswi, S. (2017, November). A lightweight approach to detect the low/high rate IP spoofed cloud DDoS attacks. In *2017 IEEE 7th International Symposium on Cloud and Service Computing (SC2)* (pp. 118-123). IEEE.

Agrawal, N., & Tapaswi, S. (2019). Defense mechanisms against DDoS attacks in a cloud computing environment: State-of-the-art and research challenges. *IEEE Communications Surveys and Tutorials*, *21*(4), 3769–3795. doi:10.1109/COMST.2019.2934468

Aladaileh, M. A., Anbar, M., Hasbullah, I. H., Chong, Y. W., & Sanjalawe, Y. K. (2020). Detection techniques of distributed denial of service attacks on software-defined networking controller–a review. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 143985–143995. doi:10.1109/ACCESS.2020.3013998

Al-Nawasrah, A., Almomani, A. A., Atawneh, S., & Alauthman, M. (2020). A survey of fast flux botnet detection with fast flux cloud computing. *International Journal of Cloud Applications and Computing*, *10*(3), 17–53. doi:10.4018/IJCAC.2020070102

Alsirhani, A., Sampalli, S., & Bodorik, P. (2019). DDoS detection system: Using a set of classification algorithms controlled by fuzzy logic system in apache spark. *IEEE eTransactions on Network and Service Management*, *16*(3), 936–949. doi:10.1109/TNSM.2019.2929425

Ashford, W. (2017, March 3). Businesses blame rivals for DDoS attacks. *Computer Weekly*. https://www.computerweekly.com/news/450414239/Businesses-blame-rivals-for-DDoS-attacks

Asosheh, A., & Ramezani, N. (2008). A comprehensive taxonomy of DDOS attacks and defense mechanism applying in a smart classification. *WSEAS Transactions on Computers*, *7*(4), 281–290.

AWS. (2020). *Threat landscape report – q1 2020*. Amazon Web Services. https://aws-shield-tlr.s3.amazonaws.com/2020-Q1_AWS_Shield_TLR.pdf

Baig, Z. A., Sait, S. M., & Shaheen, A. (2013). GMDH-based networks for intelligent intrusion detection. *Engineering Applications of Artificial Intelligence*, *26*(7), 1731–1740. doi:10.1016/j.engappai.2013.03.008

Bannister, A. (2020, July 14). *Remote working during coronavirus pandemic leads to rise in cyber-attacks, say security professionals.* The Daily Swig | Cybersecurity News and Views. https://portswigger.net/daily-swig/remote-working-during-coronavirus-pandemic-leads-to-rise-in-cyber-attacks-say-security-professionals

Behal, S., & Kumar, K. (2017). Characterization and Comparison of DDoS Attack Tools and Traffic Generators: A Review. *International Journal of Network Security*, *19*(3), 383–393.

Bessani, A. N. (2011, June). From byzantine fault tolerance to intrusion tolerance (a position paper). In *2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W)* (pp. 15-18). IEEE.

Bhardwaj, A., Subrahmanyam, G. V. B., Avasthi, V., Sastry, H., & Goundar, S. (2016, October). DDoS attacks, new DDoS taxonomy and mitigation solutions—a survey. In *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)* (pp. 793-798). IEEE. doi:10.1109/SCOPES.2016.7955549

Bhushan, K., & Gupta, B. B. (2018, February). Detecting DDoS attack using software defined network (SDN) in cloud computing environment. In *2018 5th international conference on signal processing and integrated networks (SPIN)* (pp. 872-877). IEEE.

Bhushan, K., & Gupta, B. B. (2019). Distributed denial of service (DDoS) attack mitigation in software defined network (SDN)-based cloud computing environment. *Journal of Ambient Intelligence and Humanized Computing*, *10*(5), 1985–1997. doi:10.1007/s12652-018-0800-9

Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2015). An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection. *Pattern Recognition Letters*, *51*, 1–7. doi:10.1016/j.patrec.2014.07.019

Bing, C. (2016, October 27). *You can now buy a Mirai-powered botnet on the dark web.* CyberScoop. https://www.cyberscoop.com/mirai-botnet-for-sale-ddos-dark-web/

Chae, C. J., Lee, S. H., Lee, J. S., & Lee, J. K. (2007, October). A study of defense ddos attacks using ip traceback. In *The 2007 International Conference on Intelligent Pervasive Computing (IPC 2007)* (pp. 402-408). IEEE. doi:10.1109/IPC.2007.89

Chen, L. C., Longstaff, T. A., & Carley, K. M. (2004). Characterization of defense mechanisms against distributed denial of service attacks. *Computers & Security*, *23*(8), 665–678. doi:10.1016/j.cose.2004.06.008

Chen, W., Xiao, S., Liu, L., Jiang, X., & Tang, Z. (2020). A DDoS attacks traceback scheme for SDN-based smart city. *Computers & Electrical Engineering*, *81*, 106503. doi:10.1016/j.compeleceng.2019.106503

Chen, Y., Abraham, A., & Yang, B. (2007). Hybrid flexible neural-tree-based intrusion detection systems. *International Journal of Intelligent Systems*, *22*(4), 337–352. doi:10.1002/int.20203

Cheng, J., Xu, R., Tang, X., Sheng, V. S., & Cai, C. (2018). An abnormal network flow feature sequence prediction approach for DDoS attacks detection in big data environment. *Comput. Mater. Continua*, *55*(1), 95–119.

Chhabra, M., Gupta, B., & Almomani, A. (2013). A novel solution to handle DDOS attack in MANET. *Journal of Information Security*, *4*(3), 165–179. doi:10.4236/jis.2013.43019

Chui, M., Löffler, M., & Roberts, R. (2019, February 13). *The Internet of Things.* McKinsey & Company. https://www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/the-internet-of-things

Criscuolo, P. J. (2000). *Distributed denial of service: Trin00, tribe flood network, tribe flood network 2000, and stacheldraht ciac-2319*. California Univ Livermore Radiation Lab. doi:10.2172/792253

Cvitić, I., Peraković, D., Gupta, B., & Choo, K. K. R. (2021). Boosting-based DDoS Detection in Internet of Things Systems. *IEEE Internet of Things Journal*.

Dahiya, A., & Gupta, B. B. (2019). A PBNM and economic incentive-based defensive mechanism against DDoS attacks. *Enterprise Information Systems*, 1–21. doi:10.1080/17517575.2019.1700553

Dahiya, A., & Gupta, B. B. (2021a). A QoS ensuring two-layered multi-attribute auction mechanism to mitigate DDoS attack. *Mobile Networks and Applications*, *26*(3), 1043–1058. doi:10.1007/s11036-020-01665-6

Dahiya, A., & Gupta, B. B. (2021b). A reputation score policy and Bayesian game theory based incentivized mechanism for DDoS attacks mitigation and cyber defense. *Future Generation Computer Systems*, *117*, 193–204. doi:10.1016/j.future.2020.11.027

Dantas Silva, F. S., Silva, E., Neto, E. P., Lemos, M., Venancio Neto, A. J., & Esposito, F. (2020). A taxonomy of DDoS attack mitigation approaches featured by SDN technologies in IoT scenarios. *Sensors (Basel)*, *20*(11), 3078. doi:10.3390/s20113078 PMID:32485943

Dharma, N. G., Muthohar, M. F., Prayuda, J. A., Priagung, K., & Choi, D. (2015, August). Time-based DDoS detection and mitigation for SDN controller. In *2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)* (pp. 550-553). IEEE. doi:10.1109/APNOMS.2015.7275389

Dong, S., & Sarem, M. (2019). DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 5039–5048. doi:10.1109/ACCESS.2019.2963077

Dong, S., Abbas, K., & Jain, R. (2019). A survey on distributed denial of service (DDoS) attacks in SDN and cloud computing environments. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 80813–80828. doi:10.1109/ACCESS.2019.2922196

Doriguzzi-Corin, R., Millar, S., Scott-Hayward, S., Martinez-del-Rincon, J., & Siracusa, D. (2020). LUCID: A practical, lightweight deep learning solution for DDoS attack detection. *IEEE eTransactions on Network and Service Management*, *17*(2), 876–889. doi:10.1109/TNSM.2020.2971776

Douligeris, C., & Mitrokotsa, A. (2004). DDoS attacks and defense mechanisms: Classification and state-of-the-art. *Computer Networks*, *44*(5), 643–666. doi:10.1016/j.comnet.2003.10.003

Džaferović, E., Sokol, A., Abd Almisreb, A., & Norzeli, S. M. (2019). DoS and DDoS vulnerability of IoT: A review. *Sustainable Engineering and Innovation*, *1*(1), 43–48. doi:10.37868/sei.v1i1.36

Eesa, A. S., Orman, Z., & Brifcani, A. M. A. (2015). A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. *Expert Systems with Applications*, *42*(5), 2670–2679. doi:10.1016/j.eswa.2014.11.009

Eliyan, L. F., & Di Pietro, R. (2021). DoS and DDoS attacks in Software Defined Networks: A survey of existing solutions and research challenges. *Future Generation Computer Systems*, *122*, 149–171. doi:10.1016/j.future.2021.03.011

Eslahi, M., Salleh, R., & Anuar, N. B. (2012, November). Bots and botnets: An overview of characteristics, detection and challenges. In *2012 IEEE International Conference on Control System, Computing and Engineering* (pp. 349-354). IEEE. doi:10.1109/ICCSCE.2012.6487169

Evans, D. (2011, April). *The Internet of Things: How the Next Evolution of the Internet Is Changing Everything?* https://www.cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

Farahmandian, S., Zamani, M., Akbarabadi, A., Moghimi, Y., Mirhosseini Zadeh, S. M., & Farahmandian, S. (2013). A survey on methods to defend against DDoS attack in cloud computing. *System, 6*(22), 26.

Feng, W., Zhang, Q., Hu, G., & Huang, J. X. (2014). Mining network data for intrusion detection through combining SVMs with ant colony networks. *Future Generation Computer Systems*, *37*, 127–140. doi:10.1016/j.future.2013.06.027

Fernández-Caramés, T. M., & Fraga-Lamas, P. (2018). A Review on the Use of Blockchain for the Internet of Things. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 32979–33001. doi:10.1109/ACCESS.2018.2842685

Firch, J. (2021, August 6). *2021 Cyber Security Statistics: The Ultimate List of Stats, Data & Trends*. https://purplesec.us/resources/cyber-security-statistics

Fouladi, R. F., Kayatas, C. E., & Anarim, E. (2016, June). Frequency based DDoS attack detection approach using naive Bayes classification. In *2016 39th International Conference on Telecommunications and Signal Processing (TSP)* (pp. 104-107). IEEE. doi:10.1109/TSP.2016.7760838

Garcia, M., Bessani, A., Gashi, I., Neves, N., & Obelheiro, R. (2011, June). OS diversity for intrusion tolerance: Myth or reality? In *2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN)* (pp. 383-394). IEEE.

Giacomoni, J. (2013). *Extending SDN architectures with F5's L4-7 application and gateway services*. F5 Networks, Inc.

Gu, G., Perdisci, R., Zhang, J., & Lee, W. (2008a). *Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection*. Academic Press.

Gu, G., Zhang, J., & Lee, W. (2008b). *BotSniffer: Detecting botnet command and control channels in network traffic*. Academic Press.

Gupta, A., Verma, T., Bali, S., & Kaul, S. (2013, January). Detecting MS initiated signaling DDoS attacks in 3G/4G wireless networks. In *2013 Fifth International Conference on Communication Systems and Networks (COMSNETS)* (pp. 1-60). IEEE. doi:10.1109/COMSNETS.2013.6465568

Hameed, S., & Ali, U. (2016, April). Efficacy of live DDoS detection with Hadoop. In *NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium* (pp. 488-494). IEEE. doi:10.1109/NOMS.2016.7502848

Hanna, A. (2021, July 29). *The Invisible U.S.-Iran Cyber War*. The Iran Primer. https://iranprimer.usip.org/blog/2019/oct/25/invisible-us-iran-cyber-war

Hansen, J. V., Lowry, P. B., Meservy, R. D., & McDonald, D. M. (2007). Genetic programming for prevention of cyberterrorism through dynamic and evolving intrusion detection. *Decision Support Systems*, *43*(4), 1362–1374. doi:10.1016/j.dss.2006.04.004

Haworth, J. (2020, August 26). *New Zealand stock exchange hit by series of DDoS attacks*. The Daily Swig | Cybersecurity News and Views. https://portswigger.net/daily-swig/new-zealand-stock-exchange-hit-by-series-of-DDoS-attacks

Haworth, J. (2021a, February 16). *UK cryptocurrency exchange EXMO knocked offline by 'massive' DDoS attack*. The Daily Swig | Cybersecurity News and Views. https://portswigger.net/daily-swig/uk-cryptocurrency-exchange-exmo-knocked-offline-by-massive-ddos-attack

Haworth, J. (2021b, April 21). *Telecoms industry facing increased DDoS attacks, report warns*. The Daily Swig | Cybersecurity News and Views. https://portswigger.net/daily-swig/telecoms-industry-facing-increased-DDoS-attacks-report-warns

He, L., Yan, Z., & Atiquzzaman, M. (2018). LTE/LTE-A network security data collection and analysis for security measurement: A survey. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 4220–4242. doi:10.1109/ACCESS.2018.2792534

He, Q., Wang, C., Cui, G., Li, B., Zhou, R., Zhou, Q., Xiang, Y., Jin, H., & Yang, Y. (2021). A game-theoretical approach for mitigatingedge ddos attack. *IEEE Transactions on Dependable and Secure Computing*, 1. doi:10.1109/TDSC.2021.3055559

Hinton, G. E. (2009). Deep belief networks. *Scholarpedia*, *4*(5), 5947. doi:10.4249/scholarpedia.5947

Hong, K., Kim, Y., Choi, H., & Park, J. (2017). SDN-assisted slow HTTP DDoS attack defense method. *IEEE Communications Letters*, *22*(4), 688–691. doi:10.1109/LCOMM.2017.2766636

Hoque, N., Bhattacharyya, D. K., & Kalita, J. K. (2015). Botnet in DDoS attacks: Trends and challenges. *IEEE Communications Surveys and Tutorials*, *17*(4), 2242–2270. doi:10.1109/COMST.2015.2457491

Hoque, N., Bhuyan, M. H., Baishya, R. C., Bhattacharyya, D. K., & Kalita, J. K. (2014). Network attacks: Taxonomy, tools and systems. *Journal of Network and Computer Applications*, *40*, 307–324. doi:10.1016/j.jnca.2013.08.001

Hoque, N., Kashyap, H., & Bhattacharyya, D. K. (2017). Real-time DDoS attack detection using FPGA. *Computer Communications*, *110*, 48–58. doi:10.1016/j.comcom.2017.05.015

Hsu, F. H., Ou, C. W., Hwang, Y. L., Chang, Y. C., & Lin, P. C. (2017). Detecting web-based botnets using bot communication traffic features. *Security and Communication Networks*, *2017*, 2017. doi:10.1155/2017/5960307

Hu, W., Hu, W., & Maybank, S. (2008). Adaboost-based algorithm for network intrusion detection. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics*, *38*(2), 577–583. doi:10.1109/TSMCB.2007.914695 PMID:18348941

Humayed, A., Lin, J., Li, F., & Luo, B. (2017). Cyber-physical systems security—A survey. *IEEE Internet of Things Journal*, *4*(6), 1802–1831. doi:10.1109/JIOT.2017.2703172

Huntley, S. (2020, October 16). *How we're tackling evolving online threats.* Google. https://blog.google/threat-analysis-group/how-were-tackling-evolving-online-threats/

Hussain, B., Du, Q., Sun, B., & Han, Z. (2020). Deep learning-based DDoS-attack detection for cyber–physical system over 5G network. *IEEE Transactions on Industrial Informatics*, *17*(2), 860–870. doi:10.1109/TII.2020.2974520

IC3. (2020). *Internet crime report 2020.* Internet Crime Complaint Center. https://www.ic3.gov/Media/PDF/AnnualReport/2020_IC3Report.pdf

IDG. (2020, July 20). *2020 Cloud computing study*. International Data Group. https://www.idg.com/tools-for-marketers/2020-cloud-computing-study/

ITIC. (2019, May 16). *Hourly Downtime Costs Rise: 86% of Firms Say One Hour of Downtime Costs $300,000+; 34% of Companies Say One Hour of Downtime Tops $1Million.* Information Technology Intelligence Consulting. https://itic-corp.com/blog/2019/05/hourly-downtime-costs-rise-86-of-firms-say-one-hour-of-downtime-costs-300000-34-of-companies-say-one-hour-of-downtime-tops-1million/

Jia, B., Huang, X., Liu, R., & Ma, Y. (2017). A DDoS attack detection method based on hybrid heterogeneous multiclassifier ensemble learning. *Journal of Electrical and Computer Engineering*, *2017*, 2017. doi:10.1155/2017/4975343

Joshi, B., Vijayan, A. S., & Joshi, B. K. (2012, January). Securing cloud computing environment against DDoS attacks. In *2012 International Conference on Computer Communication and Informatics* (pp. 1-5). IEEE. doi:10.1109/ICCCI.2012.6158817

Juniper. (2020, March 31). *IoT connections to reach 83 billion by 2024, driven by maturing industrial use cases*. Juniper Research. https://www.juniperresearch.com/press/iot-connections-to-reach-83-bn-by-2024

Kamboj, P., Trivedi, M. C., Yadav, V. K., & Singh, V. K. (2017, October). Detection techniques of DDoS attacks: A survey. In *2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON)* (pp. 675-679). IEEE. doi:10.1109/UPCON.2017.8251130

Karame, G. (2016, October). On the security and scalability of bitcoin's blockchain. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security* (pp. 1861-1862). doi:10.1145/2976749.2976756

Kaspersky. (2021, May 26). *IT threats during the 2016 Olympic Games in Brazil*. https://www.kaspersky.com/about/press-releases/2016_it-threats-during-the-2016-olympic-games-in-brazil

Khan, L., Awad, M., & Thuraisingham, B. (2007). A new intrusion detection system using support vector machines and hierarchical clustering. *The VLDB Journal*, *16*(4), 507–521. doi:10.1007/s00778-006-0002-5

Khattak, S., Ramay, N. R., Khan, K. R., Syed, A. A., & Khayam, S. A. (2013). A taxonomy of botnet behavior, detection, and defense. *IEEE Communications Surveys and Tutorials*, *16*(2), 898–924. doi:10.1109/SURV.2013.091213.00134

Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. *Computer*, *50*(7), 80–84. doi:10.1109/MC.2017.201

Koo, T. M., Chang, H. C., & Wei, G. Q. (2011, June). Construction P2P firewall HTTP-Botnet defense mechanism. In *2011 IEEE International Conference on Computer Science and Automation Engineering* (*Vol. 1*, pp. 33-39). IEEE.

Kumar, R., Arun, P., & Selvakumar, S. (2009, March). Distributed denial-of-service (ddos) threat in collaborative environment-a survey on ddos attack tools and traceback mechanisms. In *2009 IEEE International Advance Computing Conference* (pp. 1275-1280). IEEE.

Lan, L., & Jun, L. (2013, December). Some special issues of network security monitoring on big data environments. In *2013 IEEE 11th International Conference on Dependable, Autonomic and Secure Computing* (pp. 10-15). IEEE. doi:10.1109/DASC.2013.30

Lee, J., Uddin, M., Tourrilhes, J., Sen, S., Banerjee, S., Arndt, M., . . . Nadeem, T. (2014, June). mesdn: Mobile extension of sdn. In *Proceedings of the fifth international workshop on Mobile cloud computing & services* (pp. 7-14). Academic Press.

Lee, K., James, J. I., Ejeta, T. G., & Kim, H. J. (2016). Electronic voting service using block-chain. *Journal of Digital Forensics. Security and Law*, *11*(2), 8.

Leyden, J. (2020, June 18). *DDoS attacks continue to surge during coronavirus pandemic.* The Daily Swig | Cybersecurity News and Views. https://portswigger.net/daily-swig/ddos-attacks-continue-to-surge-during-coronavirus-pandemic

Li, C., Jiang, W., & Zou, X. (2009, December). Botnet: Survey and case study. In *2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC)* (pp. 1184-1187). IEEE. doi:10.1109/ICICIC.2009.127

Li, C., Wu, Y., Yuan, X., Sun, Z., Wang, W., Li, X., & Gong, L. (2018). Detection and defense of DDoS attack–based on deep learning in OpenFlow-based SDN. *International Journal of Communication Systems*, *31*(5), e3497. doi:10.1002/dac.3497

Li, Y., & Guo, L. (2007). An active learning based TCM-KNN algorithm for supervised network intrusion detection. *Computers & Security*, *26*(7-8), 459–467. doi:10.1016/j.cose.2007.10.002

Lin, W. C., Ke, S. W., & Tsai, C. F. (2015). CANN: An intrusion detection system based on combining cluster centers and nearest neighbors. *Knowledge-Based Systems*, *78*, 13–21. doi:10.1016/j.knosys.2015.01.009

Liu, G., Yi, Z., & Yang, S. (2007). A hierarchical intrusion detection model based on the PCA neural networks. *Neurocomputing*, *70*(7-9), 1561–1568. doi:10.1016/j.neucom.2006.10.146

Mahmood, T., & Afzal, U. (2013, December). Security analytics: Big data analytics for cybersecurity: A review of trends, techniques and tools. In *2013 2nd national conference on Information assurance (ncia)* (pp. 129-134). IEEE.

Mamolar, A. S., Pervez, Z., Wang, Q., & Alcaraz-Calero, J. M. (2019, June). Towards the detection of mobile ddos attacks in 5g multi-tenant networks. In *2019 European Conference on Networks and Communications (EuCNC)* (pp. 273-277). IEEE. doi:10.1109/EuCNC.2019.8801975

Mani, N., Moh, M., & Moh, T. S. (2021). Defending deep learning models against adversarial attacks. *International Journal of Software Science and Computational Intelligence*, *13*(1), 72–89. doi:10.4018/IJSSCI.2021010105

Marr, B. (2021, July 13). *What Is The Internet of Things (IoT) And How Will It Change Our World?* Bernard Marr. https://bernardmarr.com/what-is-the-internet-of-things-iot-and-how-will-it-change-our-world/

Masdari, M., & Jalali, M. (2016). A survey and taxonomy of DoS attacks in cloud computing. *Security and Communication Networks*, *9*(16), 3724–3751. doi:10.1002/sec.1539

Mavoungou, S., Kaddoum, G., Taha, M., & Matar, G. (2016). Survey on threats and attacks on mobile networks. *IEEE Access: Practical Innovations, Open Solutions*, *4*, 4543–4572. doi:10.1109/ACCESS.2016.2601009

Michalas, A., Komninos, N., & Prasad, N. R. (2011, February). Multiplayer game for ddos attacks resilience in ad hoc networks. In *2011 2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE)* (pp. 1-5). IEEE. doi:10.1109/WIRELESSVITAE.2011.5940931

Mirkin, M., Ji, Y., Pang, J., Klages-Mundt, A., Eyal, I., & Juels, A. (2020, October). BDoS: Blockchain denial-of-service. In *Proceedings of the 2020 ACM SIGSAC conference on Computer and Communications Security* (pp. 601-619). doi:10.1145/3372297.3417247

Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. *Computer Communication Review*, *34*(2), 39–53. doi:10.1145/997150.997156

Mirkovic, J., Arikan, E., Wei, S., Thomas, R., Fahmy, S., & Reiher, P. (2006, October). Benchmarks for DDoS defense evaluation. In *MILCOM 2006-2006 IEEE Military Communications conference* (pp. 1–10). IEEE.

Mishra, A., Gupta, N., & Gupta, B. B. (2021). Defense mechanisms against DDoS attack based on entropy in SDN-cloud using POX controller. *Telecommunication Systems*, *77*(1), 47–62. doi:10.1007/s11235-020-00747-w

Mizukoshi, M., & Munetomo, M. (2015, May). Distributed denial of services attack protection system with genetic algorithms on Hadoop cluster computing framework. In *2015 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1575-1580). IEEE. doi:10.1109/CEC.2015.7257075

Mladenov, B. (2019, May). Studying the DDoS attack effect over SDN controller southbound channel. In *2019 X National Conference with International Participation (ELECTRONICA)* (pp. 1-4). IEEE. doi:10.1109/ELECTRONICA.2019.8825601

Mölsä, J. (2005). Mitigating denial of service attacks: A tutorial. *Journal of Computer Security*, *13*(6), 807–837. doi:10.3233/JCS-2005-13601

Munshi, A., Alqarni, N. A., & Almalki, N. A. (2020, March). Ddos attack on IoT devices. In *2020 3rd International Conference on Computer Applications & Information Security (ICCAIS)* (pp. 1-5). IEEE. doi:10.1109/ICCAIS48893.2020.9096818

Murynets, I., & Jover, R. P. (2013, June). Anomaly detection in cellular machine-to-machine communications. In *2013 IEEE International Conference on Communications (ICC)* (pp. 2138-2143). IEEE. doi:10.1109/ICC.2013.6654843

Nagpal, B., Sharma, P., Chauhan, N., & Panesar, A. (2015, March). DDoS tools: Classification, analysis and comparison. In *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 342-346). IEEE.

Naoumov, N., & Ross, K. (2006, May). Exploiting p2p systems for ddos attacks. In *Proceedings of the 1st international conference on Scalable information systems* (pp. 47-es). Academic Press.

Nexusguard. (2020). *Annual DDoS threat report 2020.* https://blog.nexusguard.com/threat-report/annual-threat-report-2020

Noizat, P. (2015). Blockchain electronic vote. In *Handbook of digital currency* (pp. 453–461). Academic Press. doi:10.1016/B978-0-12-802117-0.00022-9

Osborne, C. (2020, June 20). *DDoS surge driven by attacks on education, government, and coronavirus information sites*. The Daily Swig | Cybersecurity News and Views. https://portswigger.net/daily-swig/ddos-surge-driven-by-attacks-on-education-government-and-coronavirus-information-sites

Parashar, M., Poonia, A., & Satish, K. (2019, July). A survey of attacks and their mitigations in software defined networks. In *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)* (pp. 1-8). IEEE. doi:10.1109/ICCCNT45670.2019.8944621

Patel, C. M., & Borisagar, A. P. V. H. (2012). Survey on taxonomy of ddos attacks with impact and mitigation techniques. *International Journal of Engineering Research & Technology (Ahmedabad)*, *1*(9).

PCMag. (2010, October 29). *"Anonymous" DDoS Takes Down RIAA Site.* https://www.pcmag.com/archive/anonymous-ddos-attack-takes-down-riaa-site-256328

Peng, T., Leckie, C., & Ramamohanarao, K. (2007). Survey of network-based defense mechanisms countering the DoS and DDoS problems. *ACM Computing Surveys*, *39*(1), 3. doi:10.1145/1216370.1216373

Perboli, G., Musso, S., & Rosano, M. (2018). Blockchain in logistics and supply chain: A lean approach for designing real-world use cases. *IEEE Access: Practical Innovations, Open Solutions*, *6*, 62018–62028. doi:10.1109/ACCESS.2018.2875782

Pérez-Díaz, J. A., Valdovinos, I. A., Choo, K. K. R., & Zhu, D. (2020). A flexible SDN-based architecture for identifying and mitigating low-rate DDoS attacks using machine learning. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 155859–155872. doi:10.1109/ACCESS.2020.3019330

Poisel, R., Rybnicek, M., & Tjoa, S. (2013, March). Game-based simulation of Distributed Denial of Service (DDoS) attack and defense mechanisms of Critical Infrastructures. In *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)* (pp. 114-120). IEEE.

Praseed, A., & Thilagam, P. S. (2018). DDoS attacks at the application layer: Challenges and research perspectives for safeguarding web applications. *IEEE Communications Surveys and Tutorials*, *21*(1), 661–685. doi:10.1109/COMST.2018.2870658

Qi, M. (2009, August). P2P network-targeted DDoS attacks. In *2009 Second International Conference on the Applications of Digital Information and Web Technologies* (pp. 843-845). IEEE.

Qi, M., & Yang, Y. (2009, August). P2P DDoS: challenges and countermeasures. In *2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery* (Vol. 7, pp. 265-268). IEEE.

Qwasmi, N., Ahmed, F., & Liscano, R. (2011, September). simulation of ddos attacks on p2p networks. In *2011 IEEE International Conference on High Performance Computing and Communications* (pp. 610-614). IEEE.

Raj, P. (Ed.). (2014). *Handbook of research on cloud infrastructures for big data analytics*. IGI Global. doi:10.4018/978-1-4666-5864-6

Riorey. (2012). *Taxonomy of DDoS attacks*. RioRey: The DDoS Specialist. https://www.riorey.com/types-of-ddos-attacks/

Rochlis, J. A., & Eichin, M. W. (1989). With microscope and tweezers: The worm from MIT's perspective. *Communications of the ACM*, *32*(6), 689–698. doi:10.1145/63526.63528

Ron, T. I., & Attias, S. (2017). Case analysis for the effect of blockchain technology in the gaming regulatory environment. *Gaming Law Review*, *21*(6), 459–460. doi:10.1089/glr2.2017.21613

Roopak, M., Tian, G. Y., & Chambers, J. (2019, January). Deep learning models for cyber security in IoT networks. In *2019 IEEE 9th annual computing and communication workshop and conference (CCWC)* (pp. 452-457). IEEE. doi:10.1109/CCWC.2019.8666588

Saad, M., Njilla, L., Kamhoua, C., Kim, J., Nyang, D., & Mohaisen, A. (2019, May). Mempool optimization for defending against ddos attacks in pow-based blockchain systems. In *2019 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)* (pp. 285-292). IEEE. doi:10.1109/BLOC.2019.8751476

Saad, M., Spaulding, J., Njilla, L., Kamhoua, C., Shetty, S., Nyang, D., & Mohaisen, D. (2020). Exploring the attack surface of blockchain: A comprehensive survey. *IEEE Communications Surveys and Tutorials*, *22*(3), 1977–2008. doi:10.1109/COMST.2020.2975999

Saad, M., Thai, M. T., & Mohaisen, A. (2018, May). POSTER: deterring ddos attacks on blockchain-based cryptocurrencies through mempool optimization. In *Proceedings of the 2018 on Asia Conference on Computer and Communications Security* (pp. 809-811). doi:10.1145/3196494.3201584

Sambandam, N., Hussein, M., Siddiqi, N., & Lung, C. H. (2018, December). Network security for iot using sdn: Timely ddos detection. In *2018 IEEE Conference on Dependable and Secure Computing (DSC)* (pp. 1-2). IEEE. doi:10.1109/DESEC.2018.8625119

Sangkatsanee, P., Wattanapongsakorn, N., & Charnsripinyo, C. (2011). Practical real-time intrusion detection using machine learning approaches. *Computer Communications*, *34*(18), 2227–2235. doi:10.1016/j.comcom.2011.07.001

Sarasamma, S. T., Zhu, Q. A., & Huff, J. (2005). Hierarchical Kohonenen net for anomaly detection in network security. *IEEE Transactions on Systems, Man, and Cybernetics. Part B, Cybernetics*, *35*(2), 302–312. doi:10.1109/TSMCB.2005.843274 PMID:15828658

Schonfeld, E. (2010, November 28). *WikiLeaks Reports It Is Under a Denial of Service Attack.* TechCrunch. https://techcrunch.com/2010/11/28/wikileaks-ddos-attack/

Selvi, V., & Shebin, R. (2016, March). Game theory based mitigation of Interest flooding in Named Data Network. In *2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (pp. 685-689). IEEE. doi:10.1109/WiSPNET.2016.7566220

Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. A. (2019, October). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In *2019 International Carnahan Conference on Security Technology (ICCST)* (pp. 1-8). IEEE. doi:10.1109/CCST.2019.8888419

Shawahna, A., Abu-Amara, M., Mahmoud, A. S., & Osais, Y. (2018). EDoS-ADS: An enhanced mitigation technique against economic denial of sustainability (EDoS) attacks. *IEEE Transactions on Cloud Computing*, *8*(3), 790–804. doi:10.1109/TCC.2018.2805907

Shi, P., & Lian, Y. (2008, April). Game-theoretical effectiveness evaluation of DDoS defense. In *Seventh International Conference on Networking (icn 2008)* (pp. 427-433). IEEE. doi:10.1109/ICN.2008.121

Shidaganti, G. I., Inamdar, A. S., Rai, S. V., & Rajeev, A. M. (2020). Scef: A model for prevention of ddos attacks from the cloud. *International Journal of Cloud Applications and Computing*, *10*(3), 67–80. doi:10.4018/IJCAC.2020070104

Shon, T., & Moon, J. (2007). A hybrid machine learning approach to network anomaly detection. *Information Sciences*, *177*(18), 3799–3821. doi:10.1016/j.ins.2007.03.025

Singel, R. (2008, January 23). *War Breaks Out Between Hackers and Scientology – There Can Be Only One.* Wired. https://www.wired.com/2008/01/anonymous-attac/

Singh, K., Guntuku, S. C., Thakur, A., & Hota, C. (2014). Big data analytics framework for peer-to-peer botnet detection using random forests. *Information Sciences*, *278*, 488–497. doi:10.1016/j.ins.2014.03.066

Somani, G., Gaur, M. S., Sanghi, D., Conti, M., & Rajarajan, M. (2017a). DDoS victim service containment to minimize the internal collateral damages in cloud computing. *Computers & Electrical Engineering*, *59*, 165–179. doi:10.1016/j.compeleceng.2016.12.004

Somani, G., Gaur, M. S., Sanghi, D., Conti, M., & Rajarajan, M. (2017b). Scale inside-out: Rapid mitigation of cloud DDoS attacks. *IEEE Transactions on Dependable and Secure Computing*, *15*(6), 959–973. doi:10.1109/TDSC.2017.2763160

Specht, S., & Lee, R. (2003). *Taxonomies of distributed denial of service networks, attacks, tools and countermeasures. CEL2003-03*. Princeton University.

Stehman, S. V. (1997). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, *62*(1), 77–89. doi:10.1016/S0034-4257(97)00083-7

Su, S. C., Chen, Y. R., Tsai, S. C., & Lin, Y. B. (2018). Detecting p2p botnet in software defined networks. *Security and Communication Networks*, *2018*, 2018. doi:10.1155/2018/4723862

Sudar, K. M., Beulah, M., Deepalakshmi, P., Nagaraj, P., & Chinnasamy, P. (2021, January). Detection of Distributed Denial of Service Attacks in SDN using Machine learning techniques. In *2021 International Conference on Computer Communication and Informatics (ICCCI)* (pp. 1-5). IEEE. doi:10.1109/ICCCI50826.2021.9402517

Sun, W., Li, Y., & Guan, S. (2019, August). An improved method of DDoS attack detection for controller of SDN. In *2019 IEEE 2nd International Conference on Computer and Communication Engineering Technology (CCET)* (pp. 249-253). IEEE. doi:10.1109/CCET48361.2019.8989356

Sung, K. Y., & Hsiao, S. W. (2019, December). Mitigating DDoS with PoW and Game Theory. In *2019 IEEE International Conference on Big Data (Big Data)* (pp. 6223-6225). IEEE. doi:10.1109/BigData47090.2019.9006081

Taj, A., & Khalil, I. (2018, November). DDoS defence mechanisms and challenges. *International Journal of Basic & Applied Sciences*, *6*(11).

Tariq, U., Hong, M., & Lhee, K. S. (2006, August). A comprehensive categorization of DDoS attack and DDoS defense techniques. In *International Conference on Advanced Data Mining and Applications* (pp. 1025-1036). Springer. doi:10.1007/11811305_112

Tewari, A., & Gupta, B. B. (2020). Secure Timestamp-Based Mutual Authentication Protocol for IoT Devices Using RFID Tags. *International Journal on Semantic Web and Information Systems*, *16*(3), 20–34. doi:10.4018/IJSWIS.2020070102

Thomas, R. M., & James, D. (2017, August). DDOS detection and denial using third party application in SDN. In *2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS)* (pp. 3892-3897). IEEE. doi:10.1109/ICECDS.2017.8390193

Tong, X., Wang, Z., & Yu, H. (2009). A research using hybrid RBF/Elman neural networks for intrusion detection system secure model. *Computer Physics Communications*, *180*(10), 1795–1801. doi:10.1016/j.cpc.2009.05.004

Tschorsch, F., & Scheuermann, B. (2016). Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys and Tutorials*, *18*(3), 2084–2123. doi:10.1109/COMST.2016.2535718

Tu, G. H., Li, C. Y., Peng, C., & Lu, S. (2015, September). How voice call technology poses security threats in 4g lte networks. In *2015 IEEE Conference on Communications and Network Security (CNS)* (pp. 442-450). IEEE. doi:10.1109/CNS.2015.7346856

Vishnoi, A., Mishra, P., Negi, C., & Peddoju, S. K. (2021). Android Malware Detection Techniques in Traditional and Cloud Computing Platforms: A State-of-the-Art Survey. *International Journal of Cloud Applications and Computing*, *11*(4), 113–135. doi:10.4018/IJCAC.2021100107

Vishwakarma, R., & Jain, A. K. (2019, April). A honeypot with machine learning based detection framework for defending IoT based Botnet DDoS attacks. In *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)* (pp. 1019-1024). IEEE. doi:10.1109/ICOEI.2019.8862720

Wang, G., Hao, J., Ma, J., & Huang, L. (2010). A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering. *Expert Systems with Applications*, *37*(9), 6225–6232. doi:10.1016/j.eswa.2010.02.102

Wang, H., Jin, C., & Shin, K. G. (2007). Defense against spoofed IP traffic using hop-count filtering. *IEEE/ACM Transactions on Networking*, *15*(1), 40–53. doi:10.1109/TNET.2006.890133

Wang, Y., & Li, G. (2019, July). Detect Triangle Attack on Blockchain by Trace Analysis. In *2019 IEEE 19th International Conference on Software Quality, Reliability and Security Companion (QRS-C)* (pp. 316-321). IEEE. doi:10.1109/QRS-C.2019.00066

Wei, W., Dong, Y., & Lu, D. (2008, September). Optimal control of DDoS defense with multi-resource max-min fairness. In *2008 IEEE Conference on Cybernetics and Intelligent Systems* (pp. 1285-1293). IEEE. doi:10.1109/ICCIS.2008.4670732

Wei, Z., Tang, H., Yu, F. R., Wang, M., & Mason, P. (2014). Security enhancements for mobile ad hoc networks with trust management using uncertain reasoning. *IEEE Transactions on Vehicular Technology*, *63*(9), 4647–4658. doi:10.1109/TVT.2014.2313865

Wen, Y., Lu, F., Liu, Y., & Huang, X. (2021). Attacks and countermeasures on blockchains: A survey from layering perspective. *Computer Networks*, *191*, 107978. doi:10.1016/j.comnet.2021.107978

Wu, S., Chen, Y., Li, M., Luo, X., Liu, Z., & Liu, L. (2020). Survive and thrive: A stochastic game for DDoS attacks in bitcoin mining pools. *IEEE/ACM Transactions on Networking*, *28*(2), 874–887. doi:10.1109/TNET.2020.2973410

Wu, Z., Pan, Q., Yue, M., & Liu, L. (2019). Sequence alignment detection of TCP-targeted synchronous low-rate DoS attacks. *Computer Networks*, *152*, 64–77. doi:10.1016/j.comnet.2019.01.031

Xiang, C., Yong, P. C., & Meng, L. S. (2008). Design of multiple-level hybrid classifier for intrusion detection system using Bayesian clustering and decision trees. *Pattern Recognition Letters*, *29*(7), 918–924. doi:10.1016/j.patrec.2008.01.008

Xiang, Y., & Zhou, W. (2005, July). A defense system against DDOS attacks by large-scale IP traceback. In *Third International Conference on Information Technology and Applications (ICITA'05)* (Vol. 2, pp. 431-436). IEEE. doi:10.1109/ICITA.2005.10

Xiao, Z., & Xiao, Y. (2012). Security and privacy in cloud computing. *IEEE Communications Surveys and Tutorials*, *15*(2), 843–859. doi:10.1109/SURV.2012.060912.00182

Xu, R., Cheng, J., Wang, F., Tang, X., & Xu, J. (2019). A DRDoS detection and defense method based on deep forest in the big data environment. *Symmetry*, *11*(1), 78. doi:10.3390/sym11010078

Yan, Q., & Yu, F. R. (2015). Distributed denial of service attacks in software-defined networking with cloud computing. *IEEE Communications Magazine*, *53*(4), 52–59. doi:10.1109/MCOM.2015.7081075

Yan, Q., Yu, F. R., Gong, Q., & Li, J. (2015). Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges. *IEEE Communications Surveys and Tutorials*, *18*(1), 602–622. doi:10.1109/COMST.2015.2487361

Yu, F. R., Tang, H., Mason, P. C., & Wang, F. (2010). A hierarchical identity based key management scheme in tactical mobile ad hoc networks. *IEEE eTransactions on Network and Service Management*, *7*(4), 258–267. doi:10.1109/TNSM.2010.1012.0362

Yu, J., Li, Z., Chen, H., & Chen, X. (2007, June). A detection and offense mechanism to defend against application layer DDoS attacks. In *International Conference on Networking and Services (ICNS'07)* (pp. 54-54). IEEE. doi:10.1109/ICNS.2007.5

Yuan, X., Li, C., & Li, X. (2017, May). DeepDefense: identifying DDoS attack via deep learning. In *2017 IEEE International Conference on Smart Computing (SMARTCOMP)* (pp. 1-8). IEEE. doi:10.1109/SMARTCOMP.2017.7946998

Yue, X., Qiu, X., Ji, Y., & Zhang, C. (2009, February). P2P attack taxonomy and relationship analysis. In *2009 11th International Conference on Advanced Communication Technology* (Vol. 2, pp. 1207-1210). IEEE.

Zargar, S. T., Joshi, J., & Tipper, D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Communications Surveys and Tutorials*, *15*(4), 2046–2069. doi:10.1109/SURV.2013.031413.00127

Zargar, S. T., Takabi, H., & Joshi, J. B. (2011, October). DCDIDP: A distributed, collaborative, and data-driven intrusion detection and prevention framework for cloud computing environments. In *7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)* (pp. 332-341). IEEE.

Zhang, C., Jiang, J., & Kamel, M. (2005). Intrusion detection using hierarchical neural networks. *Pattern Recognition Letters*, *26*(6), 779–791. doi:10.1016/j.patrec.2004.09.045

Zhao, S., Chen, K., & Zheng, W. (2009, August). Defend against denial of service attack with VMM. In *2009 eighth international conference on grid and cooperative computing* (pp. 91-96). IEEE. doi:10.1109/GCC.2009.14

Zhijun, W., Wenjing, L., Liang, L., & Meng, Y. (2020). Low-rate DoS attacks, detection, defense, and challenges: A survey. *IEEE Access: Practical Innovations, Open Solutions*, *8*, 43920–43943. doi:10.1109/ACCESS.2020.2976609