technologies

MDPI

Review

On Distributed Denial of Service Current Defense Schemes

Seth Djane Kotey [1], Eric Tutu Tchao [1,2,*] and James Dzisi Gadze [1,2]

[1] Department of Computer Engineering, Kwame Nkrumah University of Science and Technology (KNUST), PMB, Kumasi, Ghana; kotey.seth11@gmail.com (S.D.K.); jdgadze.coe@knust.edu.gh (J.D.G.)
[2] Department of Electrical Engineering, Kwame Nkrumah University of Science and Technology (KNUST), PMB, Kumasi, Ghana
* Correspondence: ettchao.coe@knust.edu.gh; Tel.: +233-2499-79837

Received: 20 December 2018; Accepted: 16 January 2019; Published: 30 January 2019

Abstract: Distributed denial of service (DDoS) attacks are a major threat to any network-based service provider. The ability of an attacker to harness the power of a lot of compromised devices to launch an attack makes it even more complex to handle. This complexity can increase even more when several attackers coordinate to launch an attack on one victim. Moreover, attackers these days do not need to be highly skilled to perpetrate an attack. Tools for orchestrating an attack can easily be found online and require little to no knowledge about attack scripts to initiate an attack. Studies have been done severally to develop defense mechanisms to detect and defend against DDoS attacks. As defense schemes are designed and developed, attackers are also on the move to evade these defense mechanisms and so there is a need for a continual study in developing defense mechanisms. This paper discusses the current DDoS defense mechanisms, their strengths and weaknesses.

Keywords: distributed denial of service (DDoS); detection; defense; traceback; attack

1. Introduction

As of December 2017, it was estimated that the number of internet users was over 4 billion with a user growth of over 1000% within the past 18 years [1]. With the increase in dependability of the internet comes with it an important challenge: data availability. Data availability is a key requirement for a network system to be considered secure.

Distributed denial of service attacks are intentional attempts by malicious users to disrupt or degrade the quality of a network or service [2]. These attacks involve a number of compromised connected online devices, known as a botnet [3]. The use of botnets makes it easier for attackers to launch massive attacks due to the fact that they harness the power of a lot of devices for an attack. Attacks involving botnets also make it difficult to determine the exact source of the attack. Differentiating between flash crowds also poses a major challenge. Spoofed source addresses to hide source addresses make it even more difficult. These attacks have been around for several years, but in recent years, the scale of attacks have increased drastically. On 28 February 2018, Akamai reported a 1.3 TBps attack on GitHub [4]. A few days later, Arbor Networks reported a 1.7 TBps attack [5]. With attacks like these, there is the need to develop robust defense schemes to protect internet networks and services. In this paper, we present current mechanisms for defending against DDoS attacks. We categorize these defense mechanisms according to the main functions they perform. We discuss their strengths and weaknesses and make some comparisons among them. This will help researchers further improve the current defense mechanisms for practical application. The rest of the paper is organized as follows: related surveys are presented in Section 2, Section 3 gives the criteria for papers included in this review, Section 4 describes DDoS attacks, Section 5 presents the reviewed

mechanisms to defend against attacks, Section 6 discusses and compares the mechanisms and Section 7 concludes the paper.

2. Related DDoS Defense Surveys

Zargar et al. presented a comprehensive overview of DDoS attacks and a classification of them based on network layer and application layer [6]. They also presented a classification of DDoS defense mechanisms based on OSI layer operation and the deployment location of the defense system; source based, destination based, network based, and hybrid (distributed) mechanisms. They further discussed the features, advantages, and disadvantages of defense mechanisms based on their deployment location. In addition, the authors categorized the defense systems by the point in time in which they begin operation (before the attack, during the attack, or after the attack). They then compared the performance of the defense mechanisms according to the classifications they used.

Carlin et al. reviewed intrusion detection and prevention systems to mitigate potential DDoS attacks in the cloud [7]. They listed the most popular DDoS attacks on cloud systems, classified and discussed intrusion detection systems along with their challenges. Classification of the intrusion detection systems was knowledge-based and anomaly-based. Sub categorizations were done based on the scalability of the management system, user authentication and the response mechanism.

Rashmi V. Deshmukh and Kailas K. Devadkar discussed DDoS attacks and provided a taxonomy of attacks in the cloud environment [8]. They classified defense mechanisms based on prevention, detection, and response to detection techniques. They also listed some factors to consider in selecting a defense solution in the cloud environment.

Somani et al. presented insights into the characterization, prevention, detection, and mitigation mechanisms of DDoS attacks in the cloud environment [9]. They presented features of cloud computing which aid in mounting a successful DDoS attack. A taxonomy of DDoS solutions was also presented and the solutions were categorized under prevention, detection, and mitigation. They concluded by discussing considerations to be made in selecting a defense solution.

Mahjabin et al. presented a review on different DDoS attacks [10]. They discussed attack phases in a DDoS attack, variations and evolutions of attacks as well as attackers' targets and motivations. They classified and analyzed prevention and mitigation techniques based on their underlying principle of operation. Underlying principles for the prevention methods reviewed were filters, secure overlay service, load balancing, honeypots, and awareness-based prevention systems. Mitigation techniques were also categorized broadly based on detection, response, and tolerance-based systems. They concluded by listing the key features, advantages, and limitations of the prevention and detection mechanisms reviewed.

Kalkan et al. presented DDoS attack scenarios in software defined networks (SDNs) [11]. Solutions to attacks were also broadly classified as intrinsic (having inherent properties) and extrinsic (depending on external factors). Solutions were also classified according to their defense function (detection, mitigation, and both detection and mitigation) and SDN switch intelligence (capable switch vs. dumb switch).

Zare et al. came up with a paper reviewing papers on DDoS attacks and countermeasures between the years 2000 and 2016 [12]. They discussed intrusion detection systems and analyzed countermeasures against DDoS attacks based on the location of the defense mechanism; source-end, core-end, victim-end, and distributed defense.

There has been a great number of surveys on DDoS defenses in previous years. DDoS attacks are on the rise and there is a constant need to present the state-of-the-art defense mechanisms to aid researches in their attempt to combat such attacks. In previous surveys however, most defense categorizations were done based on their underlying principle of operation and not the function they perform in defending against attacks. Defense mechanisms were also not discussed into detail to give a greater understanding of their operation. Also, the individual defense mechanisms were not compared with each other, but rather, comparisons were mainly done based on the underlying principle of operation or location of deployment of the defense mechanism.

In our paper, we present a taxonomy of DDoS attacks in the network layer and application layer. We categorize state of the art defense mechanisms based on the function they perform, that is, detection only, traceback only, mitigation only, and detection and mitigation mechanisms. We discuss the defense mechanisms reviewed into detail and compare them fairly using performance metrics affecting the effectiveness of a defense solution whilst giving recommendations for future research.

3. Survey Methodology

Literature was collected by searching indexing databases. Cited papers in articles considered for review were also collected. Papers included for consideration provided DDoS detection, prevention or mitigating mechanisms. Papers published before 2013 were excluded mainly because attack rates and volumes have increased drastically in the past few years, which may render older solutions obsolete.

The titles and abstracts of the collected literature were reviewed. Papers which did not propose solutions to DDoS attacks from their abstracts were also excluded from the review. Papers proposing solutions to only denial of service (DoS) attacks were also excluded, due to the relative simplicity of defending against DoS attacks.

Papers containing defense techniques which have been improved upon in subsequent papers were also excluded with the papers containing the most current improvement included.

Finally, 15 papers were selected to be reviewed in this paper.

4. DDoS Attack Taxonomy

The distributed nature of DDoS attacks tends to make it very difficult to defend against. Figure 1 shows a simplified attack network. Real attack networks involve a lot more devices. The main aim of such an attack is to degrade networks, deplete network resources and to prevent legitimate users from having access to network resources [6].

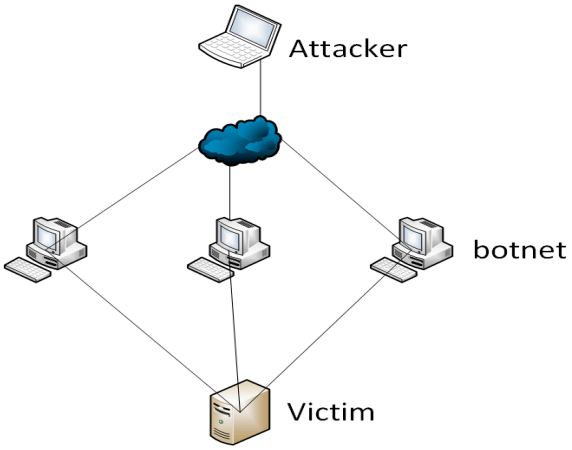

Figure 1. Simplified attack network.

A denial of service (DoS) attack is an attack in which an attacker seeks to overload a system to prevent the system from serving legitimate requests. DoS attacks are relatively simple and unsophisticated, involving just a single attacker. DoS attacks are easier to detect and defend against. Distributed DoS however, is a more sophisticated form of DoS. DDoS attacks involve a single attacker or multiple attackers, commandeering a large number of compromised devices (zombies or bots), which are collectively known as a botnet, to launch a denial of service attack on a target system (victim). Traffic originating from different sources makes it difficult to detect and defend against. Most attackers craft their attack traffic to look legitimate and may use spoofed addresses to launch the attacks, which make it more difficult to combat.

A DDoS attack is normally launched in two major phases. The attacker first builds the botnet by targeting poorly secured devices over the internet. DDoS attack programs are then installed on these

devices. These zombies also scan for poorly secured devices and compromise them as well, building a large botnet. The second phase is launching the attack itself. Large volumes of traffic are generated by the zombies and are forwarded to the target system to overload it. The distributed nature of the source of traffic make it nearly impossible to distinguish from legitimate traffic.

Earlier, attackers had to perform each step of the attack manually. This meant an attacker needed to have a lot of information about the target system and also had to be very skillful to be able to launch an attack. In recent years however, DDoS attack tools have been developed and are available on the internet for download freely or for a little amount of money [13,14]. Such tools can be used by even the least skillful attacker. They do not require an attacker to understand much about the target system or even how the attack is perpetrated. Botnets can also be hired on the internet for very low amounts of money. These have made it easier for people to perpetrate attacks [15].

DDoS attacks can be classified under many categories. In this paper, attacks are classified under two broad headings: network/transport layer attacks and application layer attacks.

Network/transport layer: Network/transport layer attacks are targeted at the network or transport layer. They are more common due to the ease of initiating them. We discuss a few of these attacks here.

TCP SYN flooding: With the TCP (transmission control protocol) SYN flooding attack, the attacker exploits the connection-oriented nature of TCP. A flood of packets is sent to the victim requesting connection without completing the TCP handshake. This leaves many half-open connections in the victim's connection state memory, denying legitimate users from connecting. [16]

UDP flooding: With the UDP (user datagram protocol) attack, a large stream of UDP packets is sent to the victim, creating a congestion on the victim's network. [17]

ICMP Smurf: With this attack, the attacker spoofs the victim's IP address as the source address to broadcast a large number of ICMP (Internet control message protocol) packets on a network using an IP (Internet protocol) broadcast address. The resultant response from devices on the network is a very large number of packets destined for the victim. [18]

Ping of death: The attacker in a ping of death attack tries to crash a victim server by sending malformed or oversized packets with the ping command. Packets are fragmented and sent and when reassembled by the victim, end up with an oversized packet (packets larger than 65,535 bytes, which is the maximum size of a ping packet). A variation of this attack is the ping flood in which the attacker sends numerous ICMP packets in a ping command without waiting for a reply. [19]

Application layer: Application layer attacks target the application layer. They tend to disrupt operation of a specific application or server to prevent legitimate users from having access to it. We discuss some of these attacks here.

DNS Amplification: Attacks of this nature rely on public DNS (domain name system) servers and a vulnerability in DNS server which makes them reply small queries with a large payload to flood a victim. The attacker sends queries with the victim's IP address as the source and the much larger response is directed to the victim. Botnets are used to scale up the impact of the attack. [20]

HTTP fragmentation attack: With the HTTP (hypertext transfer protocol) fragmentation attack, the attacker establishes a connection with the victim server and sends traffic in fragments as slowly as possible. This causes the servers to maintain longer sessions as they wait to receive traffic from the attacker. [6]

5. Defense Mechanisms

There are several defense mechanisms proposed to defend against DDoS attacks. Classification of these mechanisms can be done based on several criteria. In this paper, four categories are used in classification: attack detection only mechanisms, attack traceback only mechanisms, attack mitigation only mechanisms, and detection and mitigation mechanisms.

5.1. Detection Only Mechanisms

Detection only mechanisms are defense mechanisms with the aim to detect the presence of an attack and report to the network administrator to take action.

Service-Oriented DDoS Detection Mechanism Using Pseudo State (SDM-P)

Park et al. proposed a bidirectional flow-based detection scheme know as service-oriented DDoS detection mechanism using a pseudo state (SDM-P) [21]. SDM-P runs a bidirectional key hashing algorithm on flow routers and maintains a hashing table and pseudo state machine. Pseudo session states represent service procedure states as bit sequences. This enables flow routers which are optimized to analyze packets as bit sequences to understand and process service procedure state representations efficiently. Some proposed schemes related to this are [22–24]. The authors in [22,23] used a principal component analysis (PCA) tool to reduce the number of variables needed to analyze network traffic, reducing the amount of information needed to detect an attack. The authors in [24] used a support vector machine (SVM) for traffic classification. Both PCA and SVMs have good detection rates but are computationally costly due to their requirement of several memory read and writes. This makes them unsuitable for use in routers for analyzing real-time network traffic.

Figure 2 depicts the operation of a flow router. When a flow router receives a packet, it takes the 5-tuple and calculates a hash key. The 5-tuple consists of the source and destination IP addresses, source and destination ports, and protocol in use. If the packet flow is a new one, the destination is checked by the algorithm. If it is not to the protected server, it is forwarded normally. If it is to the protected server, the system consults the pseudo state machine and inserts an entry for the new flow in the flow table. Once there is a flow entry for the flow in the table, the pseudo state machine checks if the flow is attack traffic. If the flow does not follow the procedure stated in the pseudo state machine, it is considered to be malicious. Appropriate actions can then be taken to stop the attack.

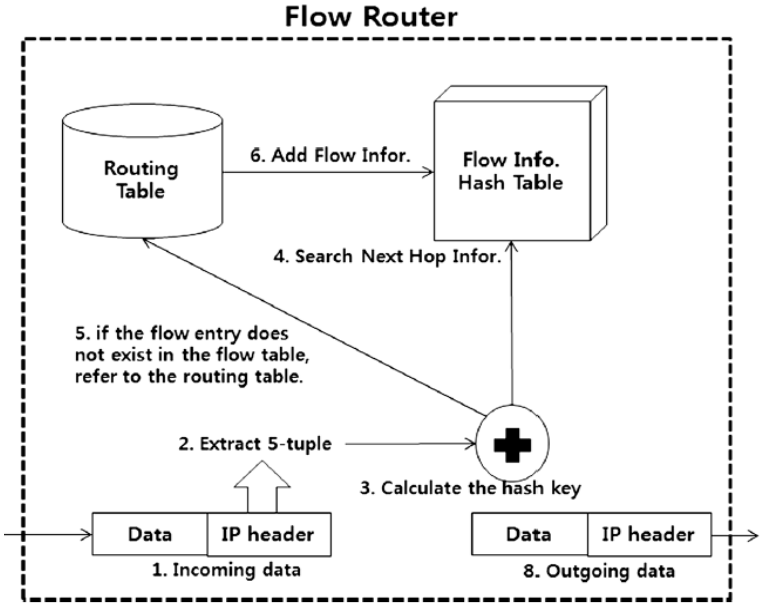

Figure 2. Operation of algorithm in flow router [21].

Though it works well against TCP, UDP and HTTP attacks, the bidirectional hashing algorithm used in SDM-P is computational expensive and uses more memory for the hash generation and creation of the hash tables. It may perform poorly in the presence of a huge attack.

5.2. Traceback Only Mechanisms

Traceback only mechanisms seek to trace the attack as close to the source as possible. These involve gathering certain information about attack packets and analyzing to determine an attack path.

Dynamical Deterministic Packet Marking (DDPM)

Yu et al. proposed dynamic packet marking scheme based on the deterministic packet marking mechanism for traceback, capable of tracing to routers as close to the attack source as possible [25]. The proposed scheme uses a marking on demand (MOD) method to give out marking identities (IDs) dynamically. The system is capable of tracing DDoS attacks to all possible sources of attack. The marking is done in a way to only capture routers related to the attack in cooperating internet domains. A MOD server is set up centrally, along with a DDoS attack detector which monitors network flow at each router or gateway. The MOD server is responsible for generating unique IDs for each router and keeping a record of the ID-router IP in its database. The attack detector requests the ID and marks the suspicious flow with the ID. If an attack is confirmed, the victim picks out the ID from the attack packets and identifies the IP addresses related to the attack (source addresses) from the MOD server's database.

Figure 3 shows the architecture and workflow of the DDPM scheme. When there is a sudden increase in network flow volume, the attack detector requests for an ID for the flow (1). The MOD server gives out the ID and stores information (source IP, timestamp) related to the ID given out in its database (2). The gateway embeds the ID into the header of the packets. A source IP query is sent to the MOD server with the ID if an attack is confirmed by the victim (3). The MOD server replies with the source IP address of the attacker(s) (4).

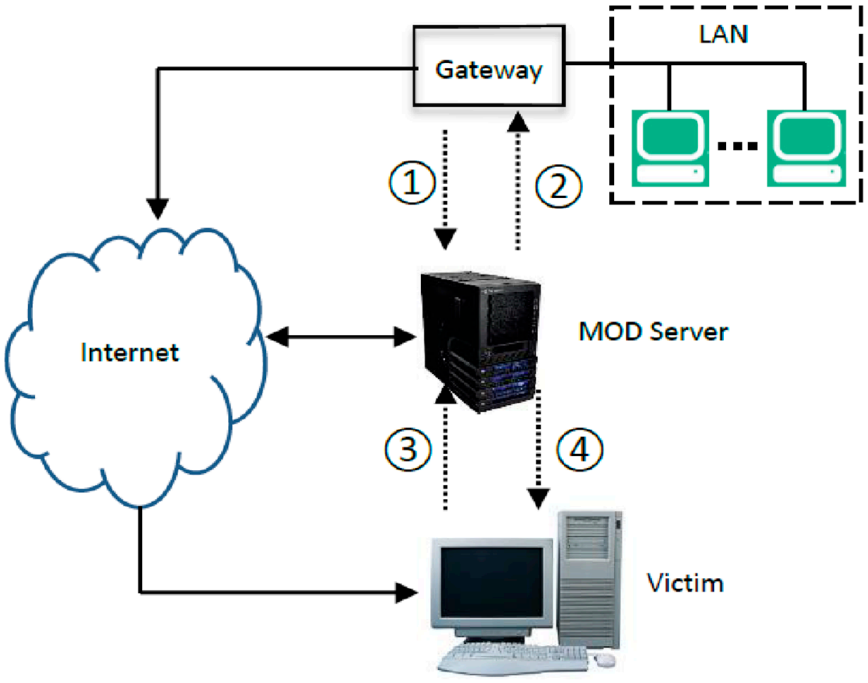

Figure 3. MOD scheme architecture [25].

According to the authors, this scheme is more scalable than traditional DPMs, however it has some limitations.

- Since the whole system hinges on the performance of the MOD server, it can easily be an attack target. Once it is knocked down, the entire system will fail.
- Also, because of the amount of data the MOD server's database will handle, it reduces the speed of the network when information is being retrieved by the victim.
- Network address translation (NAT) techniques enable IP address sharing. To traceback to a specific source, the related MAC (Media Access Control) addresses have to be included in the MOD database, which increases the information stored in the database.

Related to this work are [26–28].

The authors in [26] proposed a deterministic packet marking method. With this method, the packet source router divides its IP address in two (16 bits each) and embeds it in two packets to enable the victim trace the source router of the packet. The fragment ID (16 bits) and reserved flag (1 bit) are used in marking the packet. In [27], the authors bettered the marking scheme used in and the authors in [28] proposed a flexible marking scheme, which varied the length of the marking ID depending on the network protocols in operation by including the type of service (TOS) field (8 bits) in the marking process. Compared with related works, the DDPM is scalable and uses lower storage for IDs.

5.3. Detection and Mitigation Mechanisms

Detection and mitigation schemes attempt to detect the presence of an attack and implement an attack mitigation mechanism autonomously. Legitimate traffic is differentiated from malicious ones and actions are taken by the system against malicious traffic.

5.3.1. TDFA: Traceback-Based Defense against DDoS Flooding Attacks

Foroushani and Heywood proposed a defense against flooding attack based on traceback. TDFA has detection, traceback, and traffic control components [29]. The main aim of the scheme is to filter packets as close to the source as possible, thereby limiting the amount of attack packets received by the victim. This, if successfully implemented, reduces load on filtering routers and also maximizes the number of legitimate packets to reach their destination. The system works with coordination between victim end and source end defense systems.

The TDFA detection component resides on the edge network at the victim end and detects anomalies in network traffic. The authors proposed to use existing detection tools and not to propose a new one.

The traceback component uses the deterministic flow marking [30] (DFM) technique, which only requires participation of some edge routers on the attack path. The DFM consists of a DFM encoding module (DFME) running on the edge routers and a decoding module (DFMD) on the victim end. The DFME marks flows instead of packets and uses three identifiers for marking: the outside interface IP address of the edge router, the network interface identifier (NI-ID) and a node identifier (node-ID). The NI-ID is an identifier given to either the network MAC address interface or VLAN ID of the edge router. The node-ID is an identifier given to every source MAC address seen on incoming traffic from local networks. Only the edge routers are involved in marking.

The traffic control component is made up of traffic adjustment (TA) and packet filtering (PF) modules. The TA operates on the victim network's border gateway device and the TF operates on every edge router. After traceback, when the source of attack is found out, the TA notifies the defense system at the source end with a message containing information of the attack traffic. The PF does packet filtering on packets headed to the victim based on information received from the TA. A variable rate limit algorithm is proposed to control bandwidth consumption by routers forwarding the attack traffic. The varying rates are chosen based on traffic histories of the routers.

Attack traffic is filtered as close to the source as possible to prevent an entry into the victim network. However, packet forwarding rate is reduced continually till packet drop rate is zero, and this

will have an impact on legitimate connections. Also, attacks which are orchestrated to look like legitimate traffic will not be detected by the system.

Figure 4 shows a sample network with locations of the TDFA modules. The legitimate hosts are represented with 'Hx', the TA module communicates with the PF modules on the edge routers of the attack networks.

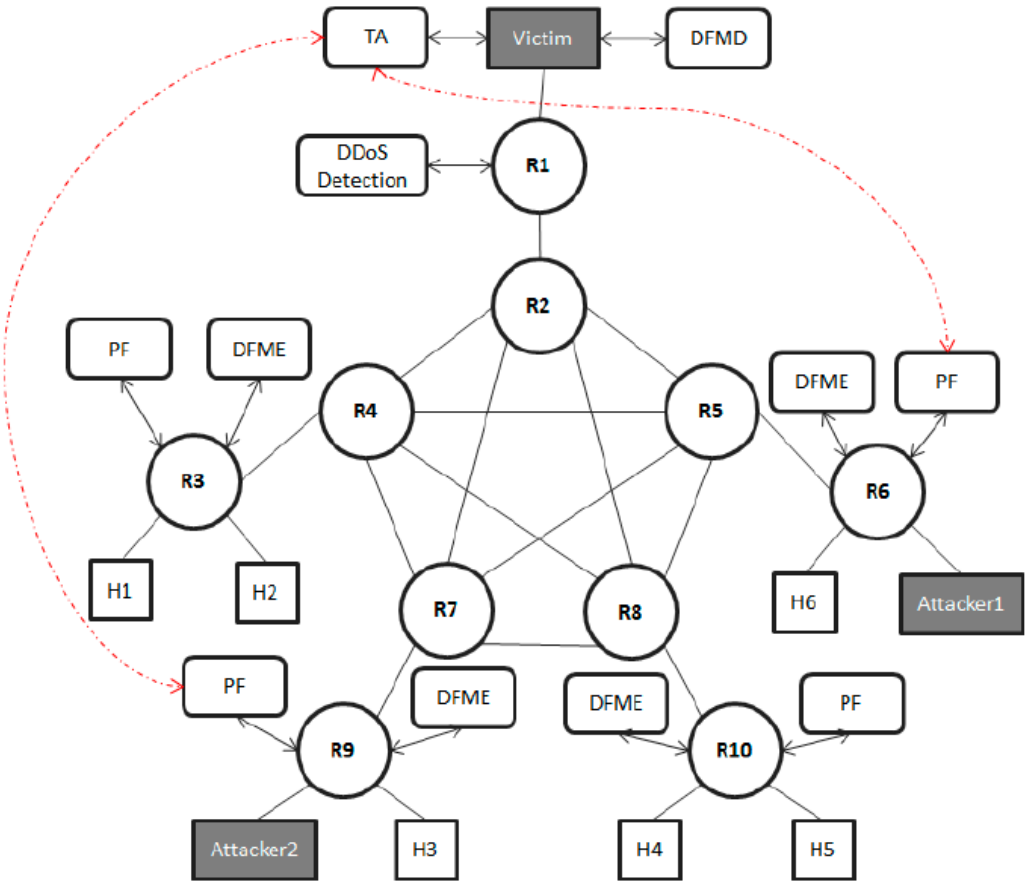

Figure 4. Sample module placement in a network [29].

The TDFA system is capable of mitigating attacks from the source end, ensuring the victim does not feel the impact of the attack much before mitigation. However, it requires cooperation from all the edge routers in the network, a scenario which might be difficult to achieve in a large network.

References [31–33] are some works related to TDFA.

The authors in [31] proposed a defense system to rely only on the edge router for traceback and packet filtering. This system however can easily be defeated if the attack is from different sources spread across the internet. In [32,33], the authors proposed a path identification approach which places a path fingerprint in every packet. This method fails if the victim's bandwidth is consumed by an attack since they only rely on packet filtering at the victim end.

5.3.2. Classifier System for Detecting and Preventing DDoS (CS_DDoS)

Sahi et al. proposed a classifier system known as CS_DDoS for DDoS TCP flood attack detection and prevention in the public cloud [34]. During detection, the system determines if a packet is legitimate or from an attacker and during prevention, packets classified as malicious are dropped and source IP blacklisted. According to the authors, the least squares support vector machine (LS-SVM) classifier produced the best performance when tested as against the K-nearest, naïve Bayes, and multilayer perceptron [35]. The system architecture and workflow are shown in Figure 5.

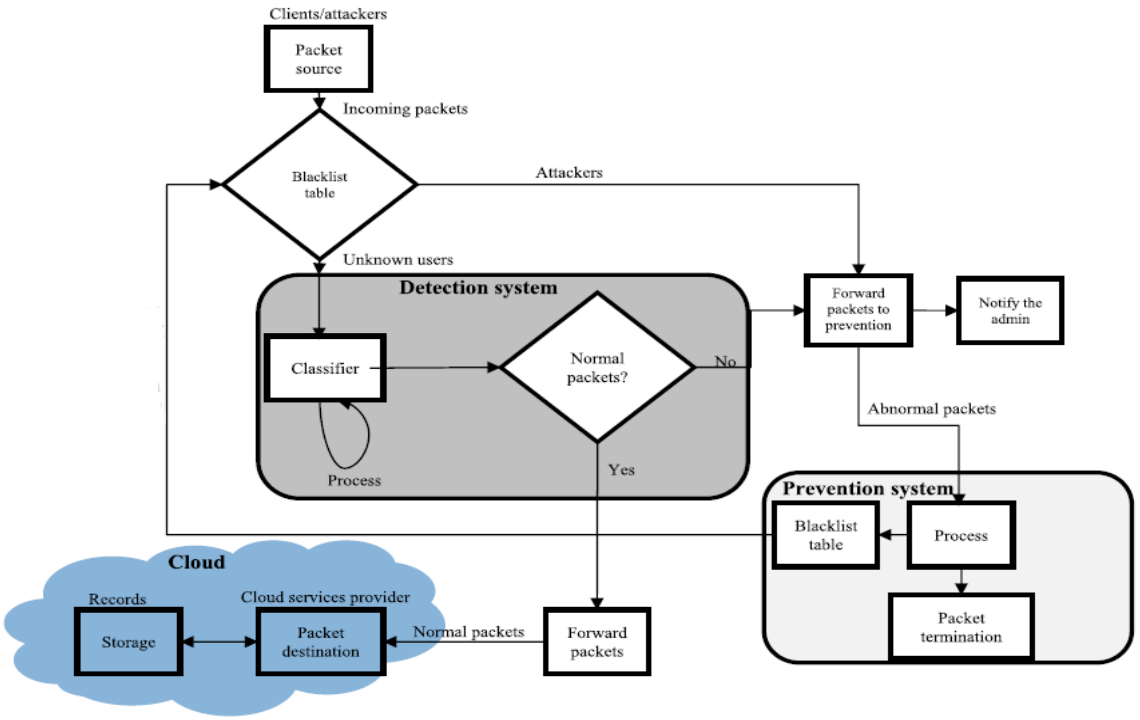

Figure 5. System architecture [34]

The scheme assumes that source IP addresses are not spoofed. During detection, the detection unit captures incoming packets within a time window and checks them against a blacklist to determine if their source addresses have been blacklisted. If the source is blacklisted, the packet is sent to the prevention unit without any further processing. If the source is not blacklisted, the packet is directed through a classifier to determine whether the packet is legitimate or malicious. A packet is considered malicious if its source requests to connect to the same destination more frequently than a set threshold. Legitimate packets are forwarded to their destination and malicious ones are forwarded to the prevention unit.

The prevention unit has three main functions; it sends a signal to the system administrator of the attack, then adds the source address to the blacklist if it does not already exist in the blacklist, and finally drops the packet. Evaluation of the system was performed under single and multiple source attack scenarios and was found to be consistent, with 97 percent and 94 percent accuracy, respectively. However, the system might be ineffective in an attack which involves a large botnet, with each device having a different IP address and sending requests below the connection requests threshold.

5.3.3. Application Layer DDoS Detection and Defense

Zhou et al. proposed a defense technique against application layer DDoS (AL-DDoS) attacks in heavy backbone web traffic. According to the authors, the mechanism can differentiate between flash crowds and DDoS attacks by examining entropy of traffic using a real-time frequency vector (RFV) [36]. The system architecture and flow of the system are shown in Figure 6. The RFV characterizes traffic in real time as a set of models. The defense system is modularized, consisting of a head-end sensor, a detection unit, and a traffic filter.

The head-end sensor, known as the abnormal traffic detection module (ATDM) tracks the amount of HTTP 'Get' requests per second. Based on the previous amount, the next amount can be predicted using the Kalman filter. The deviation between the predicted value and the actual value is compared with a set threshold and an abnormal event is reported if the deviation surpasses the threshold. The ATDM then sends an 'ATTENTION' signal to the detection module.

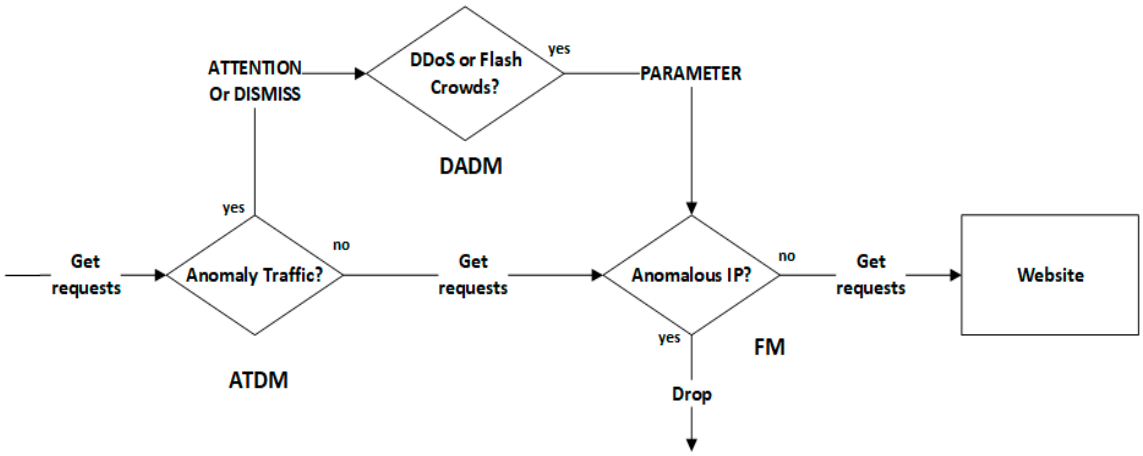

Figure 6. System architecture and flow [36].

The detection module, known as DDoS attack detection module (DADM) runs only when it receives the 'ATTENTION' signal. The DADM is works with four variations of attack traffic:

1. Repeated request AL-DDoS attack traffic: this attack traffic is directed towards the homepage or a 'hot' webpage.
2. Recursive request AL-DDoS Attack traffic: attack traffic is scattered over different web pages.
3. Repeated eorkload AL-DDoS Attack traffic: similar to the repeated request attack but employs less bots in attack which continually send large image or database searching operations requests.
4. Flash crowds: this describes a surge in visits to a website, mostly after the announcement of a new service or free software download.

The DADM traces each source address and requested webpage and records in the RFV the average frequencies. The current record is compared with a presumed normal record to determine if the type of traffic it is most likely to be using the distribution of the sources and destinations. Entropy is calculated and traffic is distinguished between attack traffic and flash crowds. Flash crowds usually have smaller ratio of entropy values.

The filter module (FM) receives malicious source IP addresses and stops the attack. A bloom filter [37] is used to filter traffic from the attack IP addresses.

A few limitations and open issues were discussed by the authors:

1. The autoregressive (AR) model does not present HTTP 'Get' traffic accurately and is used because of its simplicity.
2. Attackers may be able to avoid detection by the system by increasing attack traffic slowly until it surpasses the threshold.
3. The proposed system and method incorporates many variables and parameters whose optimized values may take time to determine.
4. Bloom filters in the FM need to be reset after an attack since bots used in an attack may be recovered by the legitimate users.

5.3.4. DDoS Attack Mitigation Architecture Using Software-Defined Networking (DaMask)

Wang et al. proposed an attack mitigation architecture in cloud computing using SDN (software defined networks) known as DaMask [38]. DaMask consists of two components: DaMask-D and DaMask-M. DaMask-D is an anomaly-based detection module and DaMask-M is the attack mitigation module. DaMask was designed to accomplish three main objectives; to be able to protect services in both public and private clouds, to incur little computational overhead and to have a low deployment cost.

The system separates network traffic for different destinations by virtualization. Each virtual network for the different destinations is known as a slice and is isolated from other slices.

DaMask-D uses anomaly-detection for attack detection. If an attack is detected, an alert is issued to the DaMask-M module, along with the packet. If the attack type is new, analysis is made on the packet to update the detection model.

The DaMask-M module performs two operations: countermeasure selection and log generation. When an attack alert is received, a countermeasure is selected to react. By default, the countermeasure is to drop attack packets. Other operations which can be used are forward and modify for advanced defense logic. After the countermeasure is selected, the policy is sent to the switch to take action. The attack packet with its information is stored in the log database. Figure 7 shows the workflow of the DaMask system.

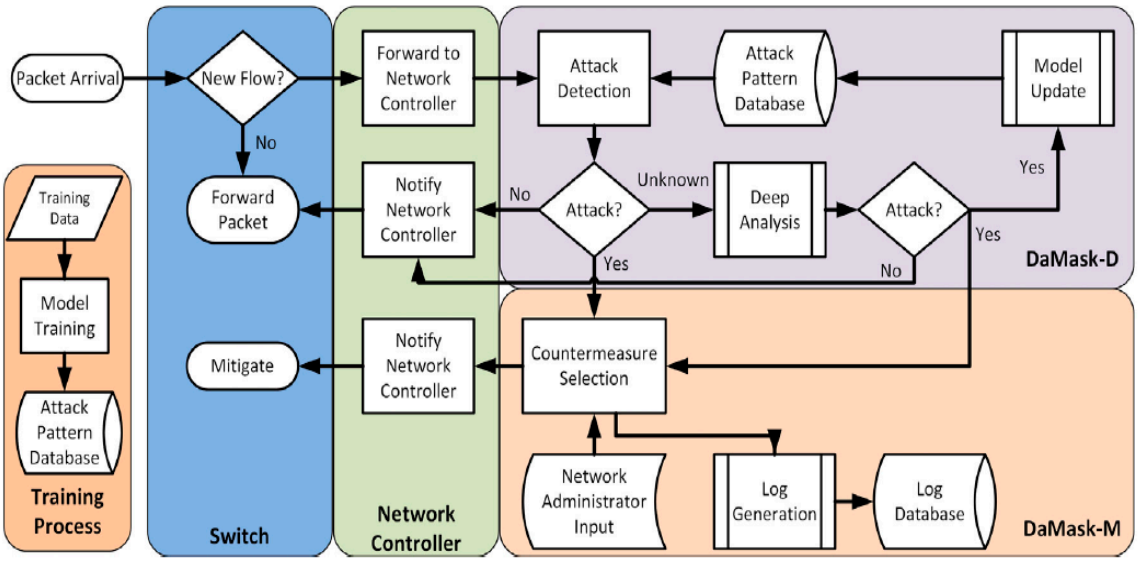

Figure 7. System workflow [38].

The DaMask system enables individual defense mechanisms to be applied to individual user spaces in the cloud without affecting other users and also periodically updates itself to address the issues associated with dataset shift. However, it introduces some communication overhead and requires a few changes to the cloud service architecture.

5.3.5. Reinforcing Anti-DDoS Actions in Realtime (RADAR)

Zheng et al. proposed a technique named reinforcing anti-DDoS action in realtime (RADAR), a real-time defense application built on commercial off-the-shelf (COTS) unmodified SDN switches to detect and defend against various flooding attacks through adaptive correlation analysis [39]. Attacks are detected by identifying certain attack features in suspicious flows.

RADAR is an application built and placed in an SDN network. It has three main units; the collector unit, the detector unit and locator unit. The architecture of RADAR is shown in Figure 8.

The RADAR collector obtains rules, both static and dynamic, to extract information of suspicious flows and keeps them in the controller to enable the RADAR detector to be able to identify if there is an attack and also the RADAR locator to determine which flows contain the attacking traffic. It only extracts information about the flow when the flow is suspicious, that is, anomalies are detected in the flow.

The RADAR detector is alerted by the collector and identifies attacks using correlative analysis of the patterns of the anomalous flows received from different switches. It is able to detect which paths carry attack traffic. It then signals the locator to accurately locate the attack traffic.

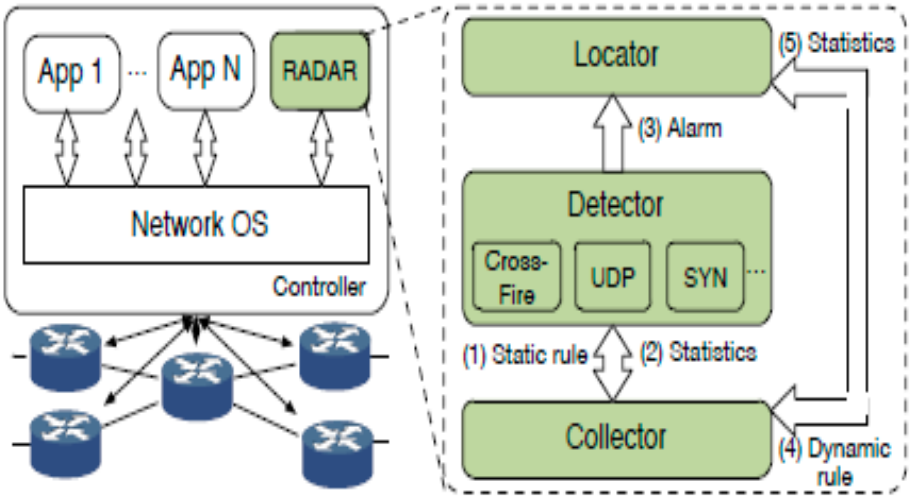

Figure 8. RADAR Architecture [39].

The RADAR locator uses an adaptive correlation analysis of the changes of flow statistics on each path and victim as detected by the detector. Traffic is tagged attack if the locator detects the rate of statistic change is consistent with aggregated flows on the victim link. The attack traffic is identified and extracts the source and destination addresses and attack traffic is blocked.

RADAR performance evaluation was conducted by the authors using both real hardware and simulation [40,41]. An advantage of the system is that it does not require specialized switches, it works with off-the-shelf SDN switches. However, in the presence of a large attack, communication overheads can increase greatly due to the number of flow rules being used.

Closely related to the work done the authors of RADAR are [42–44].

In [42], the authors proposed a traffic-engineering model to track traffic source rate change but requires major changes in the SDN switches and detects only link flooding. Lee et al. [43] proposed a collaborative traffic-engineering model to differentiate between low-rate attacks and legitimate traffic. In [44], the authors presented a traffic-engineering based analytical framework for defending link-flooding attacks.

5.3.6. Software Defined Anti-DDoS (SD-Anti-DDoS)

Cui et al. proposed a mechanism for defending DDoS attacks in SDNs. The mechanism, named software defined anti-DDoS (SD-Anti-DDoS), consists of four modules: a detection trigger, a detection module, a traceback module and a mitigation module [45]. Numerous attack mitigation schemes had been proposed already but lacked detection methods which jump into action only when an attack is in play. The authors proposed a scheme which includes a trigger mechanism for detection, instead of periodically checking for attacks. Figure 9 shows the workflow and component modules of the system.

The attack detection trigger module controls the launching of the detection module. It uses the packet_in abnormal detection algorithm message which is available only in SDN. The message contains relevant information about the ID of the switch and the reason why the packet is being sent to the controller.

The attack detection module is made up of a neural network model training stage and the real-time detection stage. It takes advantage of neural networks to differentiate legitimate traffic from malicious ones. Certain information of a flow is taken by the controller and the eigen values are taken and sent to the neural network for classification, whether it is legitimate or malicious. The back propagation neural network (BPNN) [46] is used to classify traffic flow accordingly. The neural network is trained at start-up using previously obtained extracted features. In the real-time detection stage, flow statistic messages about flows are sent to the controller by the switches. If the flow is considered to be malicious,

its destination IP address is added to an attack list. If the malicious flow entries reach a pre-set upper threshold, an attack alert is generated.

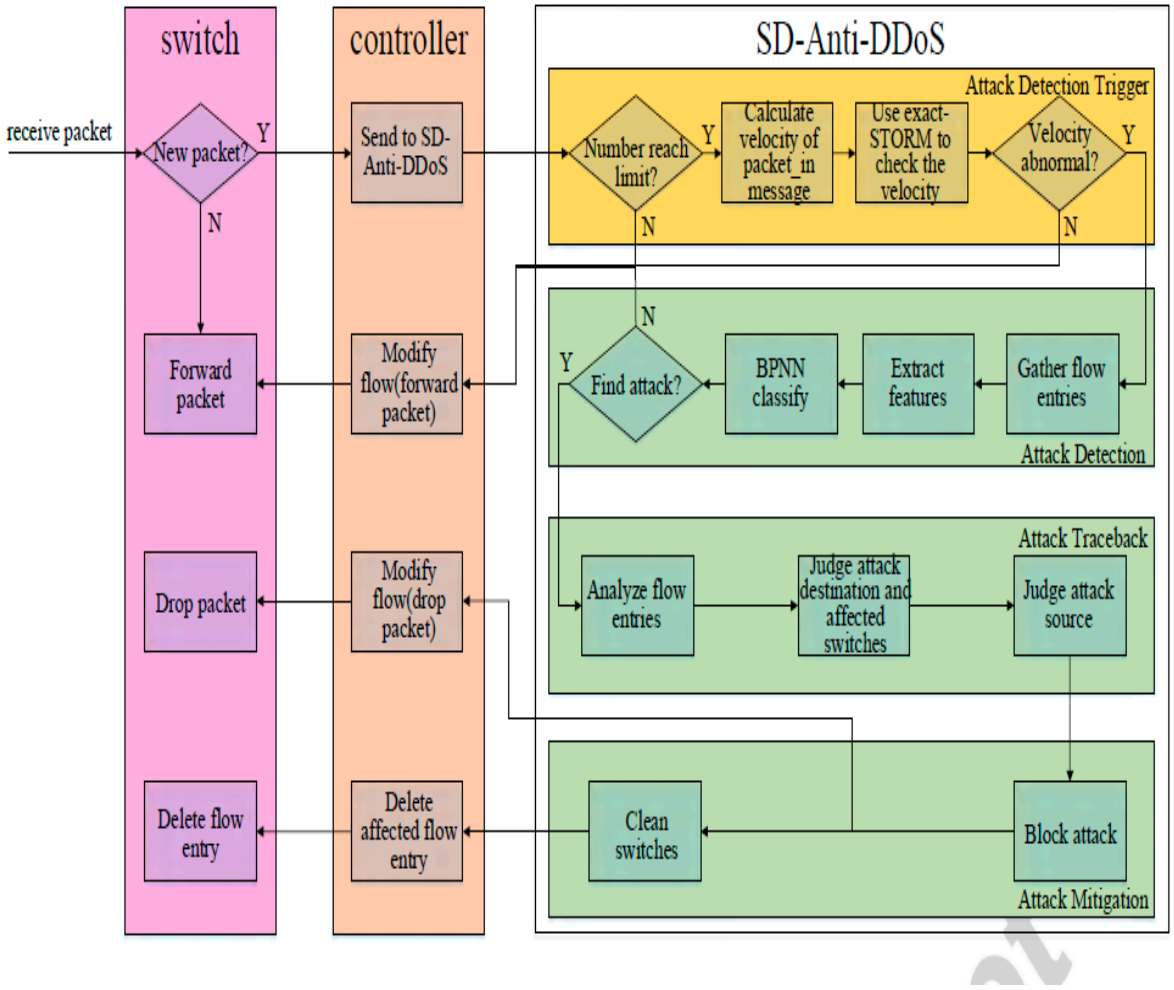

Figure 9. Workflow and components [45].

The attack traceback module works with the detection module to trace the attack switch. The switches in the attack path are found using the BPNN model and the whole attack path is located using the network topology, attacked destination, and marked attack switches.

The attack mitigation module begins operation when the attack path and source are found out. Traffic from the attack source is then blocked.

The performance of the entire system and each module was implemented on the RYU [47] framework and evaluated by the authors using Mininet [48,49]. A critical advantage of this system is that it only sends an attack trigger message when an attack is detected, reducing CPU and network load used by the defense mechanism when there is no attack.

5.3.7. Game Theoretical Holt-Winters for Digital Signature (GT-HWDS)

De Assis et al. proposed an independent defense scheme for software defined networks (SDN) [50]. Game Theoretical Holt-Winters for Digital Signature (GT-HWDS) combines an independent decision-making model built on game theory (GT) with anomaly detection and identification from a Holt-Winters for Digital Signature (HWDS) system [51,52]. The defense scheme was proposed to defend against attacks aimed at the control plane of SDNs. The system design and network topology are shown in Figure 10.

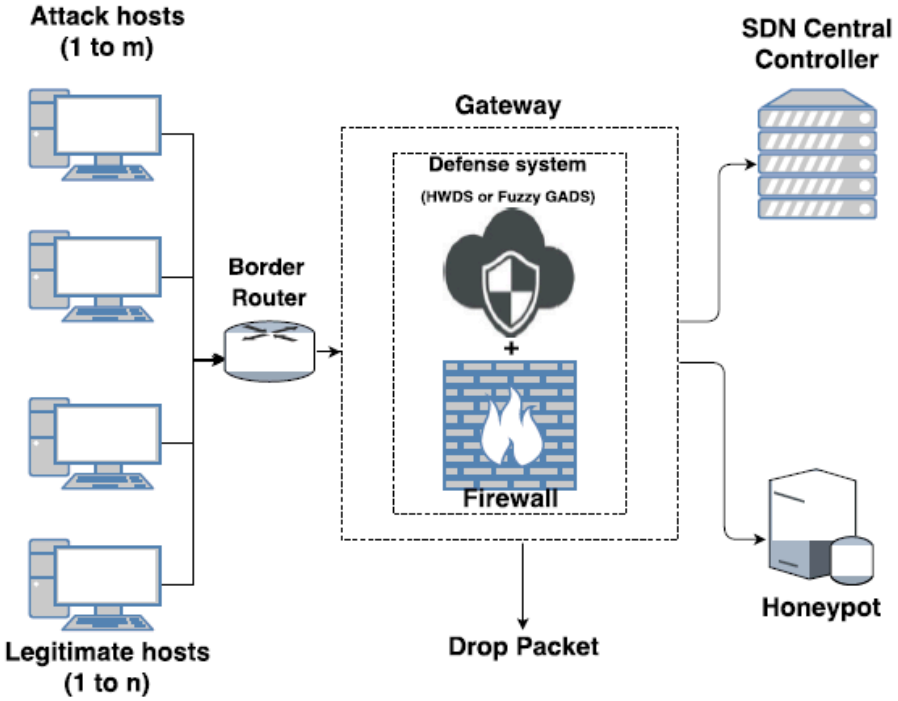

Figure 10. System design [50].

The system composes of three main parts; detection, identification and mitigation modules. The processes performed by these modules are done before data enters into the network making it operable with any type of SDN setup.

The HWDS system performs both the detection and information processes. The detection process uses a method that analyses seven parallel IP flow dimensions to characterize traffic. The dimensions analyzed are the entropy of the source and destination IP addresses and ports, bits per second, packets per second and flows per second. The HWDS system creates a signature, known as the digital signature of network segment using flow analysis (DSNSF) for each analyzed dimension. The signatures are compared with signatures of known attacks and if found to be similar, the information and mitigation modules are activated. The information module relays information about the anomaly to both the network administrator and the mitigation module.

The mitigation module uses the GT approach to take measures autonomously to mitigate the attack. Calculations are made for an optimal defense strategy and the system performs the packet dropping and redirection to a honeypot for further analysis. The defense strategy is kept in play for an hour before the network resumes its normal operation mode. If another attack is detected within this period, a new defense strategy is calculated and the defense parameters will be updated. Only packets with new source addresses are dropped or redirected to the honeypot. New sources are seen by the information module.

The HWDS system was compared with a Fuzzy-GADS (genetic algorithm with double strings) system [53] and was found to perform better with most of the tests. It works with known attacks hence new or modified attacks may not be detected. The framework of this system is related to what was proposed by [54] but is more precise in detection, employing a deeper search of signatures of attacks.

5.3.8. Detection and Defense Algorithms of Different Types of DDoS Attacks (DDAD)

Yusof et al. proposed a DDoS detection and defense algorithm to defend against four types of attacks: UDP and TCP flooding, Smurf, and ping of death attacks [55]. The proposed detection algorithm checks if incoming traffic is normal or malicious. If packet arrival rate is greater than 100 packets/second, it is tagged malicious and will be dropped. A hybrid of Snort and IPTables is used by the defense system for deep packet inspection and to reduce incoming packet speed to protect

the network in the presence of an attack. The attacks are then logged with details of the attack type, packet size, severity of the attack, duration of the attack, and attacker source address. Figure 11 shows the flow graph of the system. The algorithm is yet to be tested and evaluated by the authors.

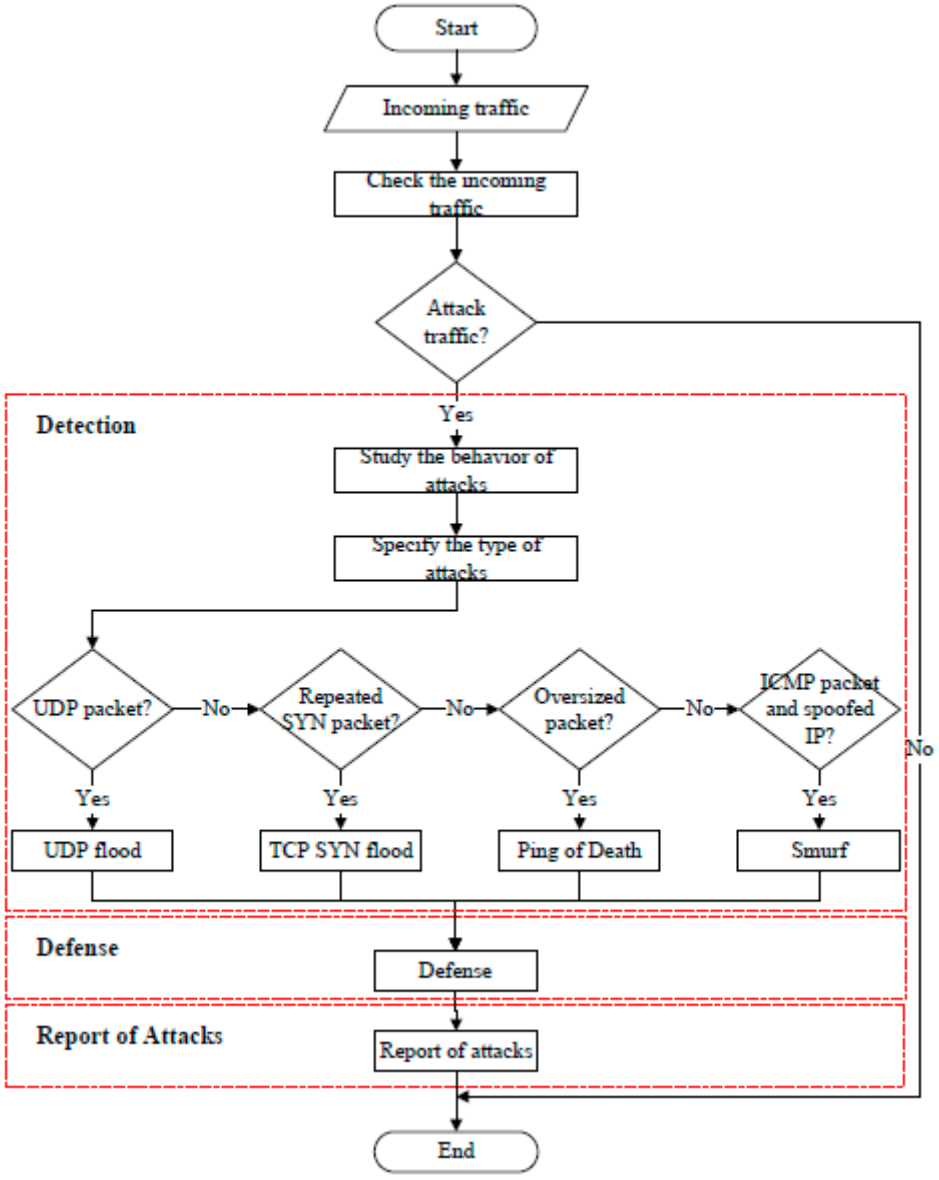

Figure 11. Proposed design flow [55].

5.4. Mitigation Only Mechanisms

Mitigation only schemes are designed to mitigate an attack. They assume attack detection is performed by an already existing algorithm.

5.4.1. VFence

Jakaria et al. proposed a defense technique based on the network function virtualization (NFV) technology. VFence defends against attacks through a dynamic traffic filtering network [56]. The mechanism intercepts packets during an attack with the help of dynamic network agents. An overview of the system topology is shown in Figure 12.

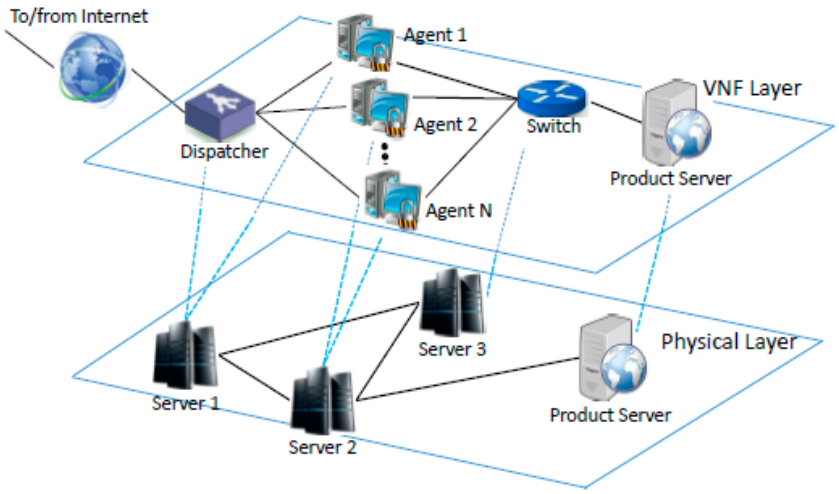

Figure 12. VFence defense topology [56].

A dispatcher directs incoming traffic to an agent for processing. The distribution of packets is based on a dynamic agent assignment algorithm, which is performed in conformity with a forwarding table. The forwarding table maps flows to handling agents and is updated each time a flow is established, expires or ends. Agents are deployed or retired based on the network flow. Agents are added or deleted from the mapping as and when necessary. The agent, a dynamically created packet handler, filters out traffic with spoofed source IPs or known attack IPs and forwards legitimate traffic to the destination. The agents can also act as a dynamic firewall to drop suspicious flows. The agents relay information on their utilization, termination, and completion of connections to the dispatcher for forwarding table updates. The number of active agents in the system at each point in time is dependent on the flow of incoming packets. The higher the flow, the more the agents. The dispatchers and agents are virtualized network functions (VNF). A load balancing algorithm is used in directing packets to agents to evenly distribute the load among the agents and to avoid having a more than necessary number of agents. The system however, introduces a delay in the network

5.4.2. Moving Target Defense Mechanism Against (MOTAG) Internet DDoS

Wang et al. proposed MOTAG, a moving target defense mechanism against internet DDoS attacks [57]. The mechanism defends against DDoS attacks by employing dynamic proxies to transmit traffic between authenticated servers and clients. The scheme was aimed at protecting sensitive online services against attacks. Clients are shuffled across hidden proxies, that is, proxies with IP addresses known to only legitimate clients, in the event of an attack using a greedy algorithm. The system architecture of MOTAG is shown in Figure 13. The authors assumed that using an effective authentication scheme eliminates the possibility of attacks from outside sources and so the system was designed to counter attacks from authenticated clients. An external attack means an insider made known a node address to an external source for an attack, thus, isolating and blocking that insider deals with the attack. Each authenticated client is randomly given an active proxy node and sees only the IP address of the proxy node assigned to it. MOTAG comprises four main inter-connected units; the authentication server, the filter ring, the proxy nodes and the application server. The application server is the provider of the online service to be protected.

The authentication server gives out a token for every client-to-proxy session and limits the client's throughput to a specific number of packets per session, which helps to detect the presence of an attack. Cryptographic puzzles must be solved by clients before they are authenticated. When an attack is ongoing, an additional number of nodes are created. All the nodes are grouped under either serving proxy nodes or shuffling proxy nodes. Serving nodes are relatively unchanging, providing service to known innocent users. Shuffling nodes are shuffled between suspicious clients and are replaced if

found to be under attack. Shuffling is done till the malicious clients are found out. The main idea is that since the suspicious clients are being moved around different nodes with different IP addresses, if a new node is attacked, the insider can only be connected to the new node. Shuffles are done till the actual malicious client is pinpointed.

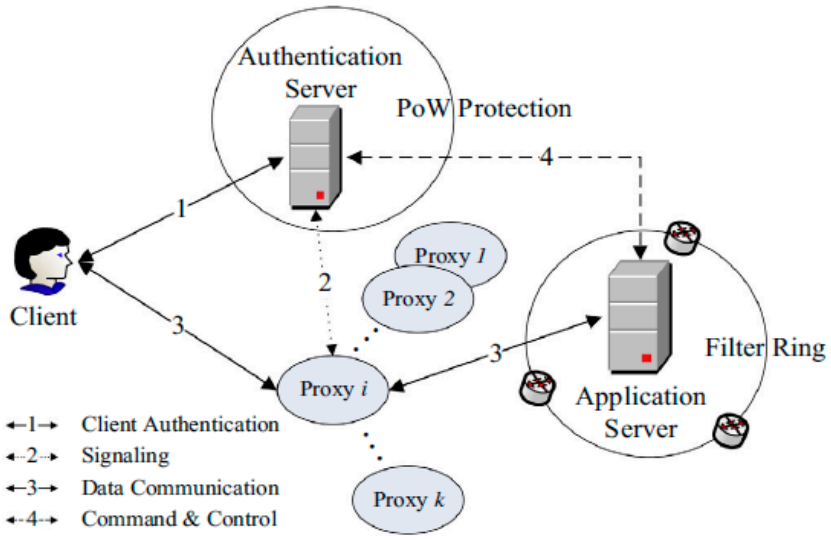

Figure 13. MOTAG Architecture [57].

Shuffling and pairing will become huge if there are a lot of clients assigned to a proxy node. Attacks from external sources cannot be fully prevented and an attack will go on for a long period unyil offending client(s) are pinpointed. Computing capacity and bandwidth is reserved to meet needs of proxy nodes as well as to provide sufficient capacity to create a large number of additional proxy nodes. MOTAG is only capable of mitigating network flooding attacks on Internet services that mandate client authentication. It is not suitable for securing open Internet services designed for anonymous users.

Nevertheless, MOTAG is resistant to brute-force attacks from outsiders due to the hidden proxy nodes.

5.4.3. Catch Me If You Can (CMIYC)

Jia et al. proposed a cloud-enabled moving target defense system based on shuffling. The approach uses server replication and client reassignment to make victim servers moving targets to isolate attacks [58]. The scheme improves on concepts proposed by the authors of MOTAG [57]. During an attack, server replication is done quickly and selectively in the cloud. The attacked servers are replaced with the replicated servers and are taken offline after clients are moved to the replica servers. The network locations of the replica servers are only known to the clients assigned to them. Shuffling of the servers is done when attackers attack the replica servers as well to determine the malicious clients. A greedy algorithm is used in the shuffling process. The architecture of the mechanism is shown in Figure 14.

A load balancer connects every new client to an active replica server using any assignment algorithm. Replica servers admit only the clients whose IP addresses are confirmed by the referring load balancer. During an attack, replacement replica servers are created across the cloud in different locations and reassignment of clients is done. After reassignment is done, the attacked servers are taken offline and reused. Reassignment of clients to replica servers is done till the attacking clients are isolated to one replica server. The coordination server keeps the entire state of the defense system and controls real-time actions against the attacks. It is responsible for running the shuffling algorithm and computing the optimal shuffling plan.

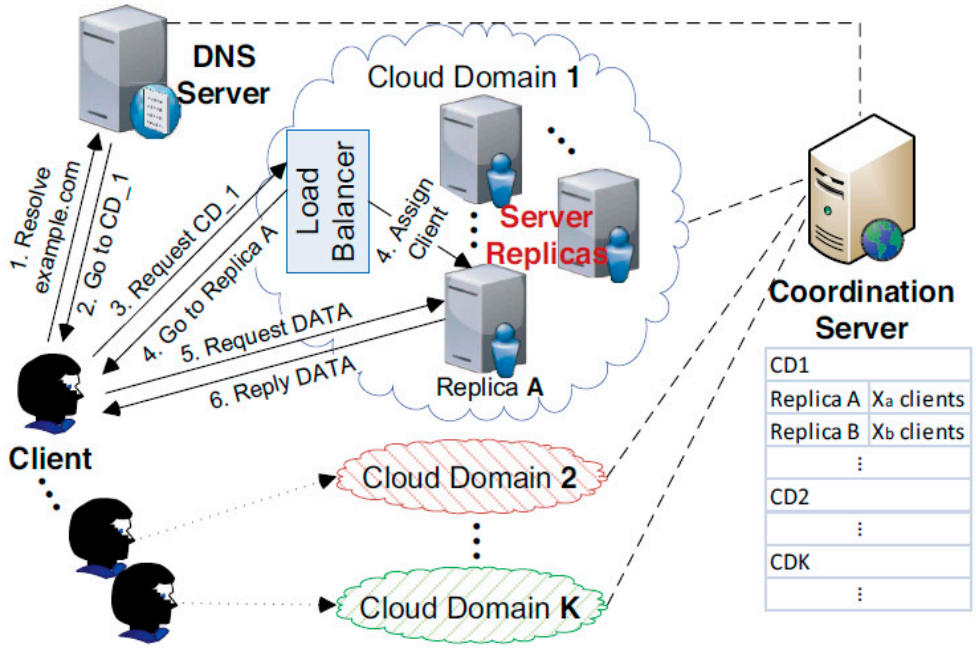

Figure 14. System architecture [58].

The system was evaluated with simulation with MATLAB as well as a prototype implemented using the Amazon EC2 [59] as a coordination server. The system is very effective on attacks aimed at static locations and from naïve bots. However, the system introduces some amount of latency in the network. Also, latter shuffles separate bots from benign users less effectively than early shuffles.

5.4.4. MultiQ

Lim et al. proposed a scheduling-based architecture for the controller in SDNs for continued operation in the presence of an attack [60]. The assumption was that the attacker does not know the exact location or IP address of the SDN controller and therefore floods a known server to overwhelm the controller. The proposed scheme, known as MultiQ, simply modifies the controller model to divide the single request processing queue into a number of logical queues, each corresponding to a flow switch. The queues are then served with a scheduling algorithm. The proposed scheme was compared with two others, Static and SingleQ and found to perform better. However, the system may be crippled if the attack is generated from different networks.

5.4.5. An Effective DDoS Defense Scheme for SDN (EDSS)

Huang et al. proposed a defense scheme to protect the controller in an SDN from attack [61]. The system predicts the number of new requests based on the Taylor series and directs requests with a prediction value beyond a previously set threshold to a security gateway. Rules are set in the security gateway and requests that cause a drastic decrease in entropy are filtered out. The rules are sent to the controller, which in turn forwards them to each switch to direct flows consistent with the rules to a honeypot. The system architecture is shown in Figure 15.

Upon arrival of a packet at the Openflow switch, a check is made for rules matching the packet. If there is a rule matching the packet, action is taken accordingly, else the packet is sent to the controller if the prediction value is below the threshold or to the security gateway if above the threshold. The security gateway determines if there is an attack or not by entropy and known attack patterns.

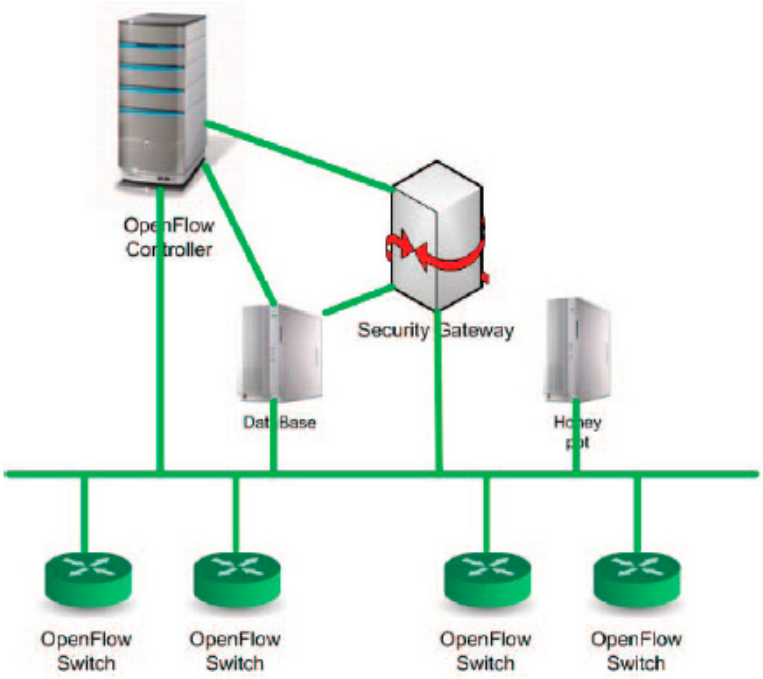

Figure 15. System architecture [61].

6. Discussion

A number of defense mechanisms have been discussed in the previous section. Table 1 shows a summary of evaluations conducted by the various authors on their proposed mechanisms. The absence of a standard set of comparison metrics informed our decision to select some metrics to better have a fair comparison. Five performance metrics were chosen and we define them below.

Table 1. Comparison of techniques

Defensive Mechanism	Scalability	Detection Rate (Accuracy)	Benign Packet Loss	Precision	Overhead Cost
SDM-P [6] D	LOW	GOOD	N/A	N/A	MEDIUM
DDPM [10] T	MEDIUM	N/A	N/A	LOW	MEDIUM
TDFA [14] DM	N/A	N/A	N/A	N/A	MEDIUM
CS_DDoS [19] DM	GOOD	HIGH	N/A	HIGH	MEDIUM
AL_DDoS Def. [22] DM	GOOD	GOOD	N/A	GOOD	N/A
DaMask [24] DM	N/A	GOOD	N/A	N/A	HIGH
RADAR [25] DM	MEDIUM	HIGH	N/A	N/A	LOW
SD Anti-DDoS [31] DM	N/A	HIGH	N/A	GOOD	LOW
GT-HWDS [36] DM	MEDIUM	HIGH	N/A	GOOD	N/A
DDAD [41] DM	N/A	N/A	N/A	N/A	N/A
VFence [42] M	N/A	N/A	LOW	N/A	N/A
MOTAG [43] M	LOW	N/A	LOW	N/A	high
CMIYC [44] M	GOOD	N/A	LOW	N/A	LOW
MultiQ [46] M	N/A	N/A	MEDIUM	N/A	N/A
EDSS [47] M	N/A	HIGH	LOW	HIGH	N/A

1. Scalability: Scalability is the ability of the defense mechanisms to function effectively in the event where there is an increase in attack traffic and attackers.

2. Detection rate (Accuracy): Accuracy is the ability of the attack detection system to correctly detect the presence or absence of an attack, that is, the true positives and negatives.
3. Benign packet loss: Benign packet loss is the number of benign packets lost during an attack.
4. Precision: Precision of the system is the ability of the system to detect the presence of an attack (true positives) consistently.
5. Overhead cost: The overhead cost is the additional computational or communication cost the defense system adds to the network.

D is used to indicate detection mechanisms in the table, **T** is used for traceback, **M** is used for mitigation, and **DM** is used for detection and mitigation mechanisms.

The metrics chosen provide an idea of how seamless the defense mechanism can be incorporated into already existing systems and networks whilst still being effective. Any network security defense mechanism should be able to cope with a large data set over a large network with many connected devices, should have a high success rate in ensuring the network is secure, should ensure legitimate users are still able to have access to resources even in the event of an attack and should ensure there is no or negligible slowdown in the network. In view of these requirements, these five metrics were chosen.

Scalability gives an idea of how the proposed system will cope if employed in a real-world scenario with a large network and multiple devices, as well as multiple attack sources. Scalability is an important metric which determines how effective or not a solution will be when adopted. Accuracy of the system helps to determine how correctly the system will detect the presence or absence of an attack. An inaccurate system will raise a lot of false alarms of an attack and will also not always detect an attack, making attack traffic flow through the system unnoticed. Benign packet loss provides information on how much benign packers are lost during an attack. One important reason for employing a DDoS defense mechanism is to ensure legitimate users have access to the network or service. Having a lot of lost benign traffic means a lot of retransmissions or requests from legitimate users or an inability of the requests to be processed, ultimately inconveniencing them. Precision helps to determine the consistency of the system in detecting the presence of an attack. The more consistent a system is, the easier it is to predict its effectiveness in managing an attack. Changes to the system to improve on its capabilities can also be done in future easily when it performs functions consistently. Overhead costs introduce some latency in the network. Most end users (legitimate users) accessing a network or service are interested in how quickly they can have their requests processed. Having a noticeable lag in the network inconveniences the end user.

Evaluation of defense schemes with respect to the metrics chosen was done with information provided by the authors of the respective schemes in their published papers. The tests and results in these papers provided an idea of how the schemes performed with respect to the metrics chosen. However, not much information was provided for some schemes, making it impossible to evaluate them with respect to certain metrics. Also, defense mechanisms which were not tested were not given any evaluation report.

A good detection or traceback mechanism should be quick to react, highly accurate, and precise in detecting an attack and should be able to perform effectively in the presence of large attack traffic. Traceback mechanisms should also be able to trace the attack back as close to the attack source or sources as possible. A good mitigation solution should have very low benign packets lost and should perform effectively in the presence of large volumes of attack traffic. Additionally, all defense mechanisms should have low overheads to avoid degrading the network further in the presence of an attack.

Based on the results provided by the authors, we realized that the overhead costs for detection and traceback was significant and had impact on the network. RADAR, SD-Anti-DDoS, and CMIYC were found to have the lowest overhead costs to the network. Scalability was found to be generally medium for most of the systems, though more real-world tests would have to be conducted to concretely

determine. Detection rates, precision, and benign packet loss were found to be generally acceptable across board.

Further optimization needs to be done to reduce overhead costs for the mechanisms. Defense mechanisms need to have low computational and communication costs on the network as this would determine how fast the defense mechanism would operate in the event of an attack. A mechanism which is fast in communication alerts all necessary systems in the defense process quickly enough to prevent a creation of a time-gap, in which attack traffic can cause some damage before it is stopped.

Scalability is also an issue which needs to be addressed further. With the increasing rates, volume and number of bots used in attacks, defense mechanisms should be able to perform effectively in the presence of a large attack. This is necessary to prevent an attack from overwhelming the defense mechanism to the point of ineffectiveness or increased degradation to the network, in which case the DDoS would have been successful.

To be able to come up with a robust defense system, scalability and overhead costs should be optimized for real-world situations, since these metrics would determine how practical a solution is.

7. Conclusions

With the increase in the scale, sophistication, and volume of modern-day DDoS attacks, it is important that more research is conducted to come up with very robust defenses to combat attacks.

We have discussed the current DDoS defense mechanisms in this paper. We classified the defense mechanisms according to the main functions they perform, that is, detection, traceback, mitigation, and detection and mitigation. We also discussed their strengths and weaknesses. We found out that most of the solutions struggle with scalability and may not be able to perform effectively in the real world, considering the increased volume of attack bots and traffic involved in modern attacks. Most of the current solutions also added some significant extra computation and communication overhead to the network, which would have an impact on the network, possibly slowing it down some more, in a real-world scenario with large volumes of attack traffic.

A comparison was made of the various mechanisms, however, not all solutions had results for the necessary metrics we used in the comparison. For such mechanisms, more test will have to be conducted to determine their actual performance based on the metrics we chose. Overall, most of the defense solutions reviewed performed acceptably well and more research should go into making them perform better, especially for a real-world scenario.

Author Contributions: Conceptualization, S.D.K. and E.T.T.; methodology, S.D.K. and E.T.T.; formal analysis, S.D.K., E.T.T. and J.D.G.; writing—original draft preparation, S.D.K.; writing—review and editing, S.D.K., E.T.T. and J.D.G.; supervision, E.T.T. and J.D.G.

Funding: The APC was partially funded by stipend payments from the TWAS-ENEA Post-Doctoral Training Programme.

Acknowledgments: This work forms part of a research project on Cloud-Based Integration Solution for Electricity Transmission in Ghana. The supervisor of this project is grateful to the Dr. Maria Valenti of ENEA Portici Research Centre and the TWAS-ENEA Post-Doctoral Training Programme for creating the platform to enable the main supervisor on this project share ideas with EU researchers.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Internet World Stats. Available online: https://www.internetworldstats.com/stats (accessed on 12 October 2018).
2. Grance, T.; Kent, K.; Kim, B. *NIST Computer Security Incident Handling Guide*; Special Publication (NIST SP): Gaithersburg, MD, USA, 2004.
3. What Does DDoS mean? Imperva Incapsula Publication. Available online: https://www.incapsula.com/ddos/denial-of-service (accessed on 12 October 2018).

4. Wired. Available online: https://www.wired.com/story/github-ddos-memcached (accessed on 10 October 2018).
5. NETSCOUT. Available online: https://asert.arbornetworks.com/netscout-arbor-confirms-1-7-tbps-ddos-attack-terabit-attack-era-upon-us (accessed on 17 October 2018).
6. Zargar, S.T.; Joshi, J.; Tipper, D. A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks. *IEEE Commun. Surv. Tutor.* **2013**, *15*, 2046–2069. [CrossRef]
7. Carlin, A.; Hammoudeh, M.; Aldabbas, O. Defence for Distributed Denial of Service Attacks in Cloud Computing. *Procedia Comput. Sci.* **2015**, *73*, 490–497. [CrossRef]
8. Deshmukh, R.V.; Devadkar, K.K. Understanding DDoS Attack & Its Effect in Cloud Environment. *Procedia Comput. Sci.* **2015**, *49*, 202–210. [CrossRef]
9. Somani, G.; Gaur, M.S.; Sanghi, D.; Conti, M.; Buyya, R. DDoS attacks in cloud computing: Issues, taxonomy, and future directions. *Comput. Commun.* **2017**, *107*, 30–48. [CrossRef]
10. Mahjabin, T.; Xiao, Y.; Sun, G.; Jiang, W. A survey of distributed denial-of-service attack, prevention, and mitigation techniques. *Int. J. Distrib. Sens. Netw.* **2017**, 13. [CrossRef]
11. Kalkan, K.; Gur, G.; Alagoz, F. Defense Mechanisms Against DDoS Attacks in SDN Environment. *IEEE Commun. Mag.* **2017**, *55*, 175–179. [CrossRef]
12. Zare, H.; Azadi, M.; Olsen, P. Techniques for Detecting and Preventing Denial of Service Attacks (a Systematic Review Approach). In *Advances in Intelligent Systems and Computing—In Information Technology-New Generations*; Springer: Cham, Switzerland, 2018; Volume 558, pp. 151–157.
13. Booters, Stressers and DDoSERS. Imperva Incapsula Publication. Available online: https://www.incapsula.com/ddos/booters-stressers-ddosers.html (accessed on 17 October 2018).
14. Santanna, J.J.; van Rijswijk-Deij, R.; Hofstede, R.; Sperotto, A.; Wierbosch, M.; Granville, L.Z.; Pras, A. Booters—An analysis of DDoS-as-a-service attacks. In Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), Ottawa, ON, Canada, 11–15 May 2015.
15. Imperva Incapsula. Available online: https://www.incapsula.com/ddos/ddos-attack-scripts.html (accessed on 10 October 2018).
16. Wang, H.; Zhang, D.; Shin, K.G. Detecting SYN flooding attacks. In Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies, New York, NY, USA, 23–27 June 2002; Volume 3.
17. Criscuolo, P.J. Distributed Denial of Service Trin00, Tribe Flood Network, Tribe Flood Network 2000, And Stacheldraht. Department of Energy Computer Incident Advisory Capability Report, Feb 14, 2000. Available online: https://e-reports-ext.llnl.gov/pdf/237595.pdf (accessed on 10 October 2018).
18. Kumar, S. Smurf-based distributed denial of service (ddos) attack amplification in internet. In Proceedings of the Second International Conference on Internet Monitoring and Protection, San Jose, CA, USA, 1–5 July 2007.
19. Elleithy, K.M.; Blagovic, D.; Cheng, W.; Sideleau, P. Denial of Service Attack Techniques: Analysis, Implementation and Comparison. *J. Syst.* **2006**, *3*, 66–71.
20. Anagnostopoulos, M.; Kambourakis, G.; Kopanos, P.; Louloudakis, G.; Gritzalis, S. DNS amplification attack revisited. *Comput. Secur.* **2013**, *39*, 475–485. [CrossRef]
21. Park, P.; Yoo, S.; Ryu, H.; Park, J.; Kim, C.H.; Choi, S.I.; Ryou, J. *A Service-Oriented DDoS Detection Mechanism Using Pseudo State in a Flow Router*; Springer: Cham, Switzerland, 2014.
22. Shon, T.; Kim, Y.; Lee, C.; Moon, J. A machine learning framework for network anomaly detection using SVM and GA. In Proceedings of the 6th Annual IEEE SMC System, Man and Cybernetics Information Assurance Workshop, West Point, NY, USA, 15–17 June 2005; pp. 176–183. [CrossRef]
23. Tanachaiwiwiat, S.; Hwang, K. Differential packet filtering against DDoS flood attacks. In Proceedings of the ACM Conference on Computer and Communications Security (CCS), Los Angeles, CA, USA, 19 May 2003.
24. Lau, F.; Rubin, S.H.; Smith, M.H.; Trajkovic, L. Distributed Denial of Service Attacks. In Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Nashville, TN, USA, 8–11 October 2000; Volume 3, pp. 2275–2280. [CrossRef]
25. Yu, S.; Zhou, W.; Guo, S.; Guo, M. A Dynamical Deterministic Packet Marking Scheme for DDoS Traceback. In Proceedings of the IEEE Global Communications Conference (GLOBECOM), Atlanta, GA, USA, 9–13 December 2013; pp. 729–734. [CrossRef]
26. Belenky, A.; Ansari, N. Ip traceback with deterministic packet marking. *IEEE Commun. Lett.* **2003**, *7*, 162–164. [CrossRef]

27. Jin, G.; Yang, J. Deterministic packet marking based on redundant decomposition for IP traceback. *IEEE Commun. Lett.* **2006**, *10*, 204–206. [CrossRef]
28. Xiang, Y.; Zhou, W.; Guo, M. Flexible deterministic packet marking: An IP traceback system to find the real source of attacks. *IEEE Trans. Parallel Distrib. Syst.* **2009**, *20*, 567–580. [CrossRef]
29. Foroushani, V.A.; Zincir-Heywood, A.N. TDFA: Traceback-based Defense against DDoS Flooding Attacks. In Proceedings of the IEEE International Conference on Advanced Information Networking and Applications, Victoria, BC, Canada, 13–14 May 2014; pp. 597–604. [CrossRef]
30. Foroushani, V.A.; Zincir-Heywood, A.N. IP traceback through (authenticated) deterministic flow marking: An empirical evaluation. *EURASIP J. Inf. Secur.* **2013**, *2013*, 5. [CrossRef]
31. Chen, S.; Song, Q. Perimeter-based defense against high bandwidth DDoS attacks. *IEEE Trans. Parallel Distrib. Syst.* **2005**, *16*, 526–537. [CrossRef]
32. Yaar, A.; Perrig, A.; Song, D. StackPi: New packet marking and filtering mechanisms for DDoS and IP Spoofing Defense. *IEEE J. Sel. Areas Commun.* **2006**, *24*, 1853–1863. [CrossRef]
33. Chen, Y.; Shantanu, D.; Pulak, D. Detecting and preventing IP-spoofed distributed DoS attacks. *Int. J. Netw. Secur.* **2008**, *7*, 69–80.
34. Sahi, A.; Lai, D.; Li, Y.; Diykh, M. An Efficient DDoS TCP Flood Attack Detection and Prevention System in a Cloud Environment. *IEEE Access* **2017**, *5*, 6036–6048. [CrossRef]
35. Hameed, A.A.; Karlik, B.; Salman, M.S. Back-propagation algorithm with variable adaptive momentum. *Knowl. Based Syst.* **2016**, *114*, 79–87. [CrossRef]
36. Zhou, W.; Jia, W.; Wen, S.; Xiang, Y.; Zhou, W. Detection and defense of application-layer DDoS attacks in backbone web traffic. *Future Gener. Comput. Syst.* **2013**, *38*, 36–46. [CrossRef]
37. Broder, A.; Mitzenmacher, M. Network applications of bloom filters: A survey. *Internet Math.* **2004**, *1*, 485–509. [CrossRef]
38. Wang, B.; Zheng, Y.; Lou, W.; Hou, T.Y. DDoS attack protection in the era of cloud computing and Software-Defined Networking. *Comput. Netw.* **2015**, *81*, 308–319. [CrossRef]
39. Zheng, J.; Li, Q.; Gu, G.; Cao, J.; Yau, D.K.Y.; Wu, J. Realtime DDoS Defense Using COTS SDN Switches via Adaptive Correlation Analysis. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 1838–1853. [CrossRef]
40. Caida. Available online: https://www.caida.org/data/passive/trace_stats/chicago-B/2015/?monitor=20150219-130000.UTC (accessed on 17 October 2018).
41. Project Floodlight. Available online: https://www.projectfloodlight.org/floodlight (accessed on 17 October 2018).
42. Kang, S.K.; Gligor, V.D.; Sekar, V. SPIFFY: Inducing Cost-Detectability Tradeoffs for Persistent Link-Flooding Attacks. In Proceedings of the Internet Society—Networks, Distributed Systems and Security Symposium, San Diego, CA, USA, 21–24 February 2016.
43. Lee, S.B.; Kang, S.M.; Gligor, V.D. CoDef: Collaborative Defense Against Large-scale Link-flooding Attacks. In Proceedings of the ACM International Conference on Emerging Networking Experiments and Technologies, Santa Barbara, CA, USA, 9–12 December 2013; pp. 417–428. [CrossRef]
44. Liaskos, C.; Kotronis, V.; Dimitropoulos, X. A Novel Framework for Modeling and Mitigating Distributed Link Flooding Attacks. In Proceedings of the IEEE International Conference on Computer Communications, San Francisco, CA, USA, 10–14 April 2016; pp. 1–9. [CrossRef]
45. Cui, Y.; Yan, L.; Li, S.; Xing, H.; Pan, W.; Zhu, J.; Zheng, X. SD-Anti-DDoS: Fast and Efficient DDoS Defense in Software-Defined Networks. *J. Netw. Comput. Appl.* **2016**, *68*, 65–79. [CrossRef]
46. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning representations by back-propagating errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
47. Ryu SDN Framework. Available online: http://osrg.github.io/ryu/ (accessed on 17 October 2018).
48. Mininet: An Instant Virtual Network on Your Laptop. Available online: http://mininet.org/ (accessed on 17 October 2018).
49. Lantz, B.; Heller, B.; McKeown, N. A network in a laptop: Rapid prototyping for software-defined networks. In Proceedings of the ACM SIGCOMM Workshop on Hot Topics in Networks, Monterey, CA, USA, 20–21 October 2010. [CrossRef]
50. De Assis, M.V.O.; Hamamoto, A.H.; Abrão, T.; Proença, L. A Game Theoretical Based System Using Holt-Winters and Genetic Algorithm with Fuzzy Logic for DoS/DDoS Mitigation on SDN Networks. *IEEE Access* **2017**, *5*, 9485–9496. [CrossRef]

51. De Assis, M.V.O.; Rodrigues, J.; Proença, M.L. A novel anomaly detection system based on seven-dimensional flow analysis. In Proceedings of the IEEE Global Telecommunications Conference, Atlanta, GA, USA, 9–13 December 2013; pp. 735–740. [CrossRef]
52. De Assis, M.V.O.; Rodrigues, J.; Proença, M.L. A seven-dimensional flow analysis to help autonomous network management. *Inf. Sci.* **2014**, *278*, 900–913. [CrossRef]
53. Hamamoto, A.H.; Carvalho, L.F.; Proença, M.L. ACO and GA metaheuristics for anomaly detection. In Proceedings of the International Conference of the Chilean Computer Science Society, Santiago, Chile, 9–13 November 2015; pp. 1–6. [CrossRef]
54. Bedi, H.; Roy, S.; Shiva, S. Game theory-based defense mechanisms against DDoS attacks on TCP/TCP-friendly flows. In Proceedings of the IEEE Symposium on Computer Intelligence in Cyber Security, Paris, France, 11–15 April 2011; pp. 129–136. [CrossRef]
55. Mohd, A.M.Y.; Fakariah, H.M.A.; Darus, M. Detection and Defense Algorithms of Different Types of DDoS Attacks Using Machine Learning. In *Computational Science and Technology*; Springer: Singapore, 2018; pp. 370–379.
56. Jakaria, A.H.M.; Rashidi, B.; Rahman, M.; Fung, C.; Wei, Y. Dynamic DDoS Defense Resource Allocation using Network Function Virtualization. In Proceedings of the ACM International Workshop on Security in Software Defined Networks and Network Function Virtualization, Scottsdale, AZ, USA, 24 March 2017. [CrossRef]
57. Wang, H.; Jia, Q.; Fleck, D.; Powell, W.; Li, F.; Stavrou, A. A moving target DDoS defense mechanism. *Comput. Commun.* **2014**, *46*, 10–21. [CrossRef]
58. Jia, Q.; Wang, H.; Fleck, D.; Li, F.; Stavrou, A.; Powell, W. Catch Me if You Can: A Cloud-Enabled DDoS Defense. In Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks, Atlanta, GA, USA, 23–26 June 2014; pp. 264–275. [CrossRef]
59. Amazon Web Services. Available online: http://aws.amazon.com (accessed on 20 October 2018).
60. Lim, S.; Yang, S.; Kim, Y.; Yang, S.; Kim, H. Controller scheduling for continued SDN operation under DDoS attacks. *Electron. Lett.* **2015**, *51*, 1259–1261. [CrossRef]
61. Huang, X.; Du, X.; Song, B. An Effective DDoS Defense Scheme for SDN. In Proceedings of the IEEE International Conference on Communications (ICC), Paris, France, 21–25 May 2017; pp. 1–6. [CrossRef]