A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks

Saman Taghavi Zargar, *Member, IEEE,* James Joshi, *Member, IEEE,* and David Tipper, *Senior Member, IEEE*

***Abstract*—Distributed Denial of Service (DDoS) flooding attacks are one of the biggest concerns for security professionals. DDoS flooding attacks are typically explicit attempts to disrupt legitimate users' access to services. Attackers usually gain access to a large number of computers by exploiting their vulnerabilities to set up attack armies (i.e., Botnets). Once an attack army has been set up, an attacker can invoke a coordinated, large-scale attack against one or more targets. Developing a comprehensive defense mechanism against identified and anticipated DDoS flooding attacks is a desired goal of the intrusion detection and prevention research community. However, the development of such a mechanism requires a comprehensive understanding of the problem and the techniques that have been used thus far in preventing, detecting, and responding to various DDoS flooding attacks.**

In this paper, we explore the scope of the DDoS flooding attack problem and attempts to combat it. We categorize the DDoS flooding attacks and classify existing countermeasures based on where and when they prevent, detect, and respond to the DDoS flooding attacks. Moreover, we highlight the need for a comprehensive distributed and collaborative defense approach. Our primary intention for this work is to stimulate the research community into developing creative, effective, efficient, and comprehensive prevention, detection, and response mechanisms that address the DDoS flooding problem before, during and after an actual attack.

***Index Terms*—Distributed Denial of Service (DDoS) flooding attack, intrusion detection systems, intrusion prevention systems, distributed DDoS defense, collaborative DDoS defense.**

I. Introduction

DENIAL of Service (DoS) attacks, which are intended attempts to stop legitimate users from accessing a specific network resource, have been known to the network research community since the early 1980s. In the summer of 1999, the Computer Incident Advisory Capability (CIAC) reported the first Distributed DoS (DDoS) attack incident [1] and most of the DoS attacks since then have been distributed in nature. Currently, there are two main methods to launch DDoS attacks in the Internet. The first method is for the attacker to send some malformed packets to the victim to confuse a protocol or an application running on it (i.e., vulnerability attack [2]). The other method, which is the most common one, involves an attacker trying to do one or both of the following:

(i) disrupt a legitimate user's connectivity by exhausting bandwidth, router processing capacity or network resources; these are essentially network/transport-level flooding attacks [2]; or

(ii) disrupt a legitimate user's services by exhausting the server resources (e.g., sockets, CPU, memory, disk/database bandwidth, and I/O bandwidth); these essentially include application-level flooding attacks [3].

Today, DDoS attacks are often launched by a network of remotely controlled, well organized, and widely scattered Zombies[1] or Botnet computers that are simultaneously and continuously sending a large amount of traffic and/or service requests to the target system. The target system either responds so slowly as to be unusable or crashes completely [2], [4]. Zombies or computers that are part of a botnet are usually recruited through the use of worms, Trojan horses or backdoors [5]–[7]. Employing the resources of recruited computers to perform DDoS attacks allows attackers to launch a much larger and more disruptive attack. Furthermore, it becomes more complicated for the defense mechanisms to recognize the original attacker because of the use of counterfeit (i.e., spoofed) IP addresses by zombies under the control of the attacker [8].

Many DDoS flooding attacks had been launched against different organizations since the summer of 1999 [1]. Most of the DDoS flooding attacks launched to date have tried to make the victims' services unavailable, leading to revenue losses and increased costs of mitigating the attacks and restoring the services. For instance, in February 2000, Yahoo! experienced one of the first major DDoS flooding attacks that kept the company's services off the Internet for about 2 hours incurring a significant loss in advertising revenue [9]. In October 2002, 9 of the 13 root servers[2] that provide the Domain Name System (DNS) service to Internet users around the world shut down for an hour because of a DDoS flooding attack [10]. Another major DDoS flooding attack occurred in February 2004 that made the SCO Group website inaccessible to legitimate users

Manuscript received 3 Jun. 2012; revised 28 Dec. 2012; accepted 11 Feb. 2013; published online Feb. 2013.

S. T. Zargar and D. Tipper are with the Telecommunications and Networking Program, School of Information Sciences, University of Pittsburgh, Pittsburgh, PA 15260 USA e-mail: (stzargar, dtipper@sis.pitt.edu)

J. Joshi is with the School of Information Sciences, University of Pittsburgh, Pittsburgh, PA 15260 USA e-mail: (jjoshi@sis.pitt.edu)

Digital Object Identifier 10.1109/SURV.2013.031413.00127

[1]Those devices (e.g., computers, routers, etc.) controlled by attackers are called zombies or bots which derives from the word "robot." The term bots is commonly referred to software applications running as an automated task over the Internet (Wikipedia, "Internet bot")

[2]DNS root servers translate logical addresses such as www.google.com into a corresponding physical IP address, so that users can connect to websites through more easily remembered names rather than numbers.

1553-877X/13/$31.00 © 2013 IEEE

[11]. This attack was launched by using systems that had previously been infected by the Mydoom virus [11]. The virus contained code that instructed thousands of infected computers to access SCO's website at the same time. The Mydoom virus code was re-used to launch DDoS flooding attacks against major government news media and financial websites in South Korea and the United States in July 2009 [12], [13]. On December 2010, a group calling themselves "Anonymous" orchestrated DDoS flooding attacks on organizations such as Mastercard.com, PayPal, Visa.com and PostFinance [14]. The attack brought down the Mastercard, PostFinance, and Visa websites. Most recently since September 2012, online banking sites of 9 major U.S. banks (i.e., Bank of America, Citigroup, Wells Fargo, U.S. Bancorp, PNC, Capital One, Fifth Third Bank, BB&T, and HSBC) have been continuously the targets of series of powerful DDoS flooding attacks launched by a foreign hacktivist group called "*Izz ad-Din al-Qassam Cyber Fighters*" [15]. Consequently, several online banking sites have slowed or grounded to a halt before they get recovered several minutes later.

Recent advances in DDoS defense mechanisms have put an end to the era in which script-kiddies could download a tool and launch an attack against almost any website. In today's DDoS attacks, attackers use more complicated methods to launch an attack. Despite all of the efforts towards decreasing the number of DDoS attack incidents, they have expanded rapidly in the frequency and the size of the targeted networks and computers. In a recent survey commissioned by VeriSign, it has been found that 75% of respondents had experienced one or more attacks between July 2008 and July 2009 [16]. Furthermore, a recent report from Arbor Networks[3] indicate similar data. In their results, they showed that 69% of the respondents had experienced at least one DDoS attack from October 2009 through September 2010, and 25% had been hit by ten such attacks per month [17]. According to Prolexic Technologies, which offers services to protect against DDoS attacks, there are 7000 DDoS attacks observed daily and they believe this number is growing rapidly [18]. DDoS attacks are also increasing in size, making them harder to defend against. Arbor Networks found that there has been around 100% increase in the attack size over 2010, with attacks breaking the 100Gbps barrier for the first time [17]. Therefore, protecting resources from these frequent and large DDoS attacks necessitates the research community to focus on developing a comprehensive DDoS defense mechanism that can appropriately respond to DDoS attacks before, during and after an actual attack.

Several taxonomies of DDoS attacks and defense mechanisms tailored to particular environments have been proposed in the literature [19]–[21]. *Geng et al.* focus on aspects of DDoS attacks unique to wireless ad hoc networks in [19]. *Wood et al.* concentrate on distinct features of DDoS attacks unique to wireless sensor networks in [20].

In this paper, we focus on DDoS flooding attacks and defense mechanisms in wired networked systems. Here, our goal is to categorize the existing DDoS flooding attacks and to provide a comprehensive survey of defense mechanisms categorized based on where and when they detect and respond to DDoS flooding attacks. Such a study of DDoS flooding attacks and the presented survey is important to understand the critical issues related to this important network security problem so as to build more comprehensive and effective defense mechanisms.

The rest of this paper is organized as follows: In Section II, we provide some insights into the motivation of attackers in launching DDoS attacks. In section III, we describe our categorization of different DDoS flooding attacks. We categorize DDoS flooding attacks into two types based on the protocol level that is targeted: network/transport-level attacks and application-level attacks. Then we enumerate some of the major attacks in each category. In section IV, we briefly review the structure of botnets and major botnet types that could be employed by attackers to launch DDoS flooding attacks. In section V, we describe our classification of the defense mechanisms for DDoS flooding attacks and discuss various defense mechanisms against DDoS flooding attacks. We classify the defense mechanisms against the two types of DDoS flooding attacks that we present in section III using two criteria. First we classify both the defense mechanisms against network/transport-level DDoS flooding attacks and the defense mechanisms against application-level DDoS flooding attacks based on the location where prevention, detection, and response to the DDoS flooding attacks occur. Then we classify both types of defense mechanisms based on the point in time when they prevent, detect, and respond to DDoS flooding attacks. Finally, we highlight the need for a comprehensive distributed and collaborative defense solution against DDoS flooding attacks by enumerating some of the important advantages of distributed DDoS defense mechanisms over centralized ones. In section VI, we enumerate some of the metrics that can be used in evaluating various DDoS defense mechanisms; we have also qualitatively compared the defense mechanisms against DDoS flooding attacks based on their deployment location. In section VII, we briefly describe the cyber-insurance policies and their role, as part of the cyber risk management of a complete cyber defense strategy, against DDoS flooding attacks. Finally, section VIII concludes our paper and provides some insights for implementing a comprehensive distributed collaborative defense mechanism against DDoS flooding attacks.

II. DDoS: Attackers' Incentives

DDoS attackers are usually motivated by various incentives. We can categorize DDoS attacks based on the motivation of the attackers into five main categories:

1) *Financial/economical gain*: These attacks are a major concern of corporations. Because of the nature of their incentive, attackers of this category are usually the most technical and the most experienced attackers. Attacks that are launched for financial gain are often the most dangerous and hard-to-stop attacks.
2) *Revenge*: Attackers of this category are generally frustrated individuals, possibly with lower technical skills, who usually carry out attacks as a response to a perceived injustice.

[3]Arbor networks include 111 IP network operators worldwide.

3) *Ideological belief*: Attackers who belong to this category are motivated by their ideological beliefs to attack their targets [22]. This category is currently one of the major incentives for the attackers to launch DDoS attacks. For instance, political incentives have led to recent sabotages in Estonia 2007 [23], Iran 2009 [24] and WikiLeaks 2010 [25].
4) *Intellectual Challenge*: Attackers of this category attack the targeted systems to experiment and learn how to launch various attacks. They are usually young hacking enthusiasts who want to show off their capabilities. Nowadays, there exist various easy to use attack tools and botnets to rent that even a computer amateur can avail of in order to launch a successful DDoS attack.
5) *Cyberwarfare*: Attackers of this category usually belong to the military or terrorist organizations of a country and they are politically motivated to attack a wide range of critical sections of another country [26], [27]. The potential targets of these attacks include, but not limited to, executive civilian departments and agencies, private/public financial organizations (e.g., national/commercial banks), energy/water infrastructures (e.g., [28]), and telecommunications and mobile service providers. Cyberwar attackers can be considered as very well trained individuals with ample resources. Attackers expend a great deal of time and resources towards disruption of services, which may severely paralyze a country and incur significant economic impacts.

There have been a few papers in the literature that focus on analyzing the attackers' incentives and how those incentives could be modeled in such a way that decision-making models could be established to stop and respond to these attacks [22], [29]. For instance in [29], the authors aim to model and infer attackers' intents, objectives, and strategies in order to provide a predictive or proactive cyber defense. In a similar study recently conducted by *Fultz et al.* [22], attackers' motives and behaviors when they are faced with diverse defense patterns, strategies, and the degree of in-dependency have been analyzed. In doing so, *Fultz et al.* [22] propose a game theoretic approach to model security decision-making in which attackers aim to deny service and defenders try hard to secure their assets at the same time. Results show that the threat of prosecution could be enough to prevent an attacker from attacking the system; however, when the number of attackers increases, this equilibrium becomes increasingly unbalanced.

One of the fundamental attack prevention methods is to lessen the attackers' interests in attacking their targets. For instance, new policies could be developed and employed. Hence, studying the attackers' incentives in launching DDoS attacks is a promising future research direction. For instance, researchers can conduct survey or interview studies with the hackers and cyber-criminals, study recent incidents, and best/worst prevention/defense practices in order to get some insights in attackers' motivations and incentives [30]. Studying attackers' incentives help develop effective policies to prevent attacks. Such policies should eventually lead to *loss of interest* by *attackers* (e.g., attack targets become either technically impossible to attack or incur substantial financial losses, attackers face imprisonment up to life).

III. DDoS Attack: Scope and Classification

The distributed nature of DDoS attacks makes them extremely difficult to combat or traceback. Attackers normally use spoofed (fake) IP addresses in order to hide their true identity, which makes the traceback of DDoS attacks even more difficult. Furthermore, there are security vulnerabilities in many Internet hosts that intruders can exploit. Moreover, incidents of attacks that target the application layer are increasing rapidly. One of the necessary steps towards deploying a comprehensive DDoS defense mechanism is to understand all the aspects of DDoS attacks. Various classifications of DDoS attacks have been proposed in the literature over the past decade [1], [2], [31]–[34], [36]. In this survey, we are interested in providing a classification of DDoS flooding attacks based on the protocol level at which the attack works. We review various DDoS flooding incidents of each category, some of which have been well reviewed/analyzed in [1], [2], [31]–[34], [36] and the rest are recent trends of DDoS flooding attacks. In this paper, we mainly focus on DDoS flooding attacks as one of the most common forms of DDoS attacks. Vulnerability attacks, in which attackers exploit some vulnerabilities or implementation bugs in the software implementation of a service to bring that down, are not the focus of this paper.

As we mentioned earlier, DDoS flooding attacks can be classified into two categories based on the protocol level that is targeted:

A. Network/transport-level DDoS flooding attacks: These attacks have been mostly launched using TCP, UDP, ICMP and DNS protocol packets. There are four types of attacks in this category [2], [36]:

A.1 Flooding attacks: Attackers focus on disrupting legitimate user's connectivity by exhausting victim network's bandwidth (e.g., Spoofed/non-spoofed UDP flood, ICMP flood, DNS flood, VoIP Flood and etc. [32], [35]).

A.2 Protocol exploitation flooding attacks: Attackers exploit specific features or implementation bugs of some of the victim's protocols in order to consume excess amounts of the victim's resources (e.g., TCP SYN flood, TCP SYN-ACK flood, ACK & PUSH ACK flood, RST/FIN flood and etc. [32], [35]).

A.3 Reflection-based flooding attacks: Attackers usually send forged requests (e.g., ICMP echo request) instead of direct requests to the reflectors; hence, those reflectors send their replies to the victim and exhaust victim's resources (e.g., Smurf and Fraggle attacks) [32], [36].

A.4 Amplification-based flooding attacks: Attackers exploit services to generate large messages or multiple messages for each message they receive to amplify the traffic towards the victim. Botnets have been constantly used for both reflection and amplification purposes. Reflection and amplification techniques are usually employed in tandem as in the case of Smurf attack where the attackers send requests with spoofed source IP addresses (Reflection) to a large number of reflectors by exploiting IP broadcast feature of the packets (Amplification) [32], [36].

All of the above attack types with their details have been well presented in [2], [32], [35], [36]. Hence, we skip

further explanation of these attacks; instead we focus on the application-level DDoS flooding attacks as they are growing rapidly and becoming more severe problems as they are stealthier than the network/transport-level flooding attacks and they masquerade as flash crowds.

B. Application-level DDoS flooding attacks: These attacks focus on disrupting legitimate user's services by exhausting the server resources (e.g., Sockets, CPU, memory, disk/database bandwidth, and I/O bandwidth) [3]. Application-level DDoS attacks generally consume less bandwidth and are stealthier in nature compared to volumetric attacks since they are very similar to benign traffic. However, application-level DDoS flooding attacks usually have the same impact to the services since they target specific characteristics of applications such as HTTP, DNS, or Session Initiation Protocol (SIP). Here we briefly describe the DNS amplification flooding attack and the SIP flooding attack as two of the famous application-level reflection/amplification flooding attacks embracing DNS and SIP protocols. Then we classify various flavors of application-level flooding attacks that employ the HTTP protocol since these attacks are consistently reported as the major types of recent DDoS flooding attacks [38].

B.1 Reflection/amplification based flooding attacks [2], [36]: These attacks use the same techniques as their network/transport-level peers (i.e., sending forged application-level protocol requests to the large number of reflectors). For instance, the DNS amplification attack employs both reflection and amplification techniques. The attackers (zombies) generate small DNS queries with forged source IP addresses which can generate a large volume of network traffic since DNS response messages may be substantially larger than DNS query messages. Then this large volume of network traffic is directed towards the targeted system to paralyze it. Another application-level attack example that employs reflection technique is *VoIP flooding* [35]. This attack is a variation of an application specific UDP flooding. Attackers usually send spoofed VoIP packets through SIP at a very high packet rate and with a very large source IP range. The victim VoIP server has to distinguish the proper VoIP connections from the forged ones that consume significant amount of resources. VoIP flooding can overwhelm a network with packets with randomized or fixed source IP addresses. If the source IP address has not been changed the VoIP flooding attack mimics traffic from large VoIP servers and can be very difficult to identify since it resembles good traffic.

B.2 HTTP flooding attacks [3], [35], [37], [39]: There are four types of attacks in this category:

B.2.1 Session flooding attacks: In this type of attack, session connection request rates from the attackers are higher than the requests from the legitimate users; hence, this exhausts the server resources and leads to DDoS flooding attack on the server. One of the famous attacks in this category is the HTTP get/post flooding attack (a.k.a., excessive VERB) [35] in which attackers generate a large number of valid HTTP requests (get/post) to a victim web server. Attackers usually employ botnets to launch these attacks. Since each of the bots can generate a large number of valid requests (usually more than 10 requests a second) there is no need for a large number of bots to launch a successful attack. HTTP get/post flooding attacks are non-spoofed attacks.

B.2.2 Request flooding attacks: In this type of attack, attackers send sessions that contain more number of requests than usual and leads to a DDoS flooding attack on the server. One of the well-known attacks in this category is the single-session HTTP get/post flooding (a.k.a., excessive VERB single session) [35]. This attack is a variation of HTTP get/post flooding attack which employs the feature of HTTP 1.1 to allow multiple requests within a single HTTP session. Hence, the attacker can limit the session rate of an HTTP attack and bypass session rate limitation defense mechanisms of many security systems.

B.2.3 Asymmetric attacks: In this type of attack, attackers send sessions that contain high-workload requests. Here, we enumerate some of the famous attacks in this category.

B.2.3.a Multiple HTTP get/post flood (a.k.a., multiple VERB single request) [35]: This attack is also a variation of HTTP get/post flood attack. Here, an attacker creates multiple HTTP requests by forming a single packet embedded with multiple requests and without issuing them one after another within a single HTTP session [35]. This way attacker can still maintain high loads on the victim server with a low attack packet rate which makes the attacker nearly invisible to netflow anomaly detection techniques. Also, attackers can easily bypass deep packet inspection techniques if they carefully select the HTTP VERB.

B.2.3.b Faulty Application [35]: In this attack, attackers take advantage of websites with poor designs or improper integration with databases. For instance, they can employ SQL-like injections to generate requests to lock up database queries. These attacks are highly specific and effective because they consume server resources (memory, CPU, etc.).

B.2.4 Slow request/response attacks: In this type of attack, attackers send sessions that contain high-workload requests. There are a number of famous attacks in this category that we describe in the following.

B.2.4.a Slowloris attack (a.k.a, slow headers attack) [40]: Slowloris is a HTTP get-based attack that can bring down a Web server using a limited number of machines or even a single machine. The attacker sends partial HTTP requests (not a complete set of request headers [41]) that continuously and rapidly grow, slowly update, and never close. The attack continues until all available sockets are taken up by these requests and the Web server becomes inaccessible. Attackers' source addresses are usually not spoofed.

B.2.4.b HTTP fragmentation attack [35]: Similar to Slowloris, the goal of this attack is to bring down a Web server by holding up the HTTP connections for a long time without raising any alarms. Attackers (bots) (non-spoofed) establish a valid HTTP connection with a web server. Then they fragment legitimate HTTP packets into tiny fragments and send each fragment as slow as the server time out allows. Using this approach, by opening multiple sessions on each bot, the attacker can silently bring down a Web server with just a handful of bots.

B.2.4.c Slowpost attack (a.k.a, slow request bodies or R-U-Dead-Yet (RUDY) attack) [42]: *Wong et al.* present a very similar attack to Slowloris that send HTTP post commands

slowly to bring down Web servers. The attacker sends a complete HTTP header that defines the "content-length" field of the post message body as it sends this request for benign traffic. Then it sends the data to fill the message body at a rate of one byte every two minutes. Hence, the server waits for each message body to be completed while Slowpost attack grows rapidly which causes the DDoS flooding attack on the Web server.

B.2.4.d Slowreading attack (a.k.a, slow response attack) [43]: Shekyan presents another type of attack in this category which works by slowly reading the response instead of slowly sending the requests. This attack achieves its purpose by setting a smaller receive window-size than the target server's send buffer. The TCP protocol maintains open connections even if there is no data communication; hence, the attacker can force the server to keep a large number of connections open and eventually causes the DDoS flooding attack on the server.

The message here is that DDoS, like most malicious security threats, is multidimensional. One must be prepared to detect and counter both the more well-known attacks that aggressively assault systems and the novel attacks that will slip in and undermine systems before you know what hit them.

IV. Botnet-based DDoS Attacks

As mentioned earlier, botnets are the dominant mechanisms that facilitate DDoS flooding attacks on computer networks or applications. Most of the recent and most problematic application layer DDoS flooding attacks have employed botnets. In this section, we present a comprehensive study of current botnet architectures and the tools that have been used to launch DDoS flooding attacks.

According to *Peng et al.* [32], there are two main reasons that make the development of an effective DDoS defense mechanism even more challenging when attackers employ zombies to launch DDoS flooding attacks. First, a large number of zombies involved in the attack facilitates attackers to make the attacks larger in scale and more disruptive. Second, zombies' IP addresses are usually spoofed under the control of the attacker, which makes it very difficult to traceback the attack traffic even to the zombies.

Usually a group of zombies that are controlled by an attacker (*a.k.a.* Master) form a botnet. Botnets consist of masters, handlers, and bots (*a.k.a.* Agents), as depicted in Figure 1. The handlers are means of communication that attackers (i.e., masters) use to communicate indirectly with their bots (i.e., to command and control their army). For instance, handlers can be programs installed on a set of compromised devices (e.g., network servers) that attackers communicate with to send commands. However, most of these installed programs leave unique footprints behind that are detectable with current antivirus software. Hence, currently attackers use other methods (e.g., Internet Relay Chat (IRC)) to communicate with their bots in order to send commands and control them. Bots are devices that have been compromised by the handlers. Bots are those systems that will eventually carry out the attack on the victim's system. Figure 1 shows all the elements of a botnet. Botnets can have hundreds of

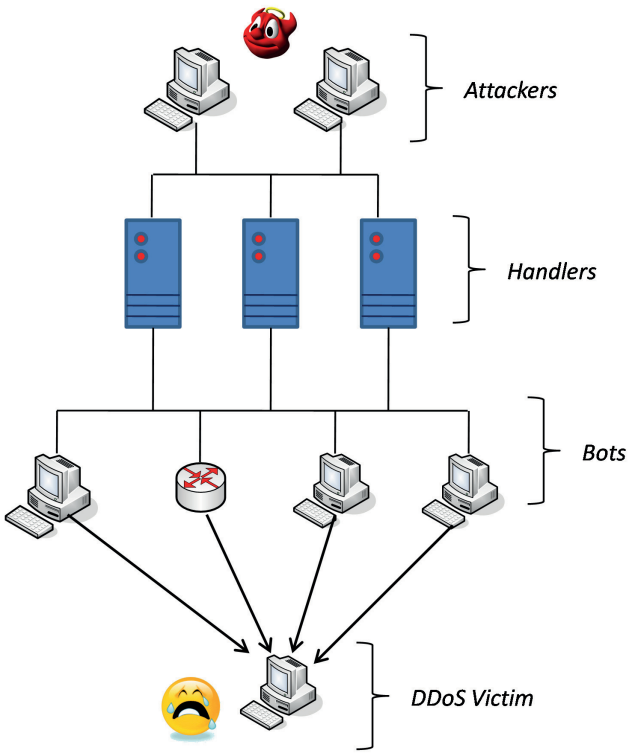

Fig. 1. Elements of a Botnet.

various implementations. Based on how bots are controlled by the masters, botnets are classified into three major categories [8], [44]: *IRC-based*, *Web-based*, and *P2P-based*. Since the first two categories have been widely used to launch DDoS flooding attacks, we briefly explain them and introduce some of the tools that have been used in each category.

1) **IRC-based** [45]: IRC is an on-line text-based instant messaging protocol in the Internet. It has client/server architecture with default channels to communicate between servers. IRC can connect hundreds of clients via multiple servers. Using IRC channels as handlers, attackers can use legitimate IRC ports to send commands to the bots making it much more difficult to track the DDoS command and control structure. Furthermore, an attacker can easily hide his presence because of the large volume of traffic that IRC servers usually have. Additionally, an attacker can easily share files to distribute the malicious code. Moreover, attackers can simply log on to the IRC server and see the list of all the available bots instead of maintaining their list locally at their site. The major limitation of botnets with a centralized command and control (C&C) infrastructure such as IRC-based botnets is that the servers are a potential central points of failure. That is, the entire botnet can be shutdown if the defender captures the C&C servers. Several well-known IRC-based botnet tools have been developed and used over the years for launching DDoS attacks such as: Trinity v3 [46] (conducts UDP, TCP SYN, TCP ACK, and TCP NUL flood attacks), and Kaiten [47] (conducts UDP, TCP, SYN, and PUSH+ACH flood attacks).

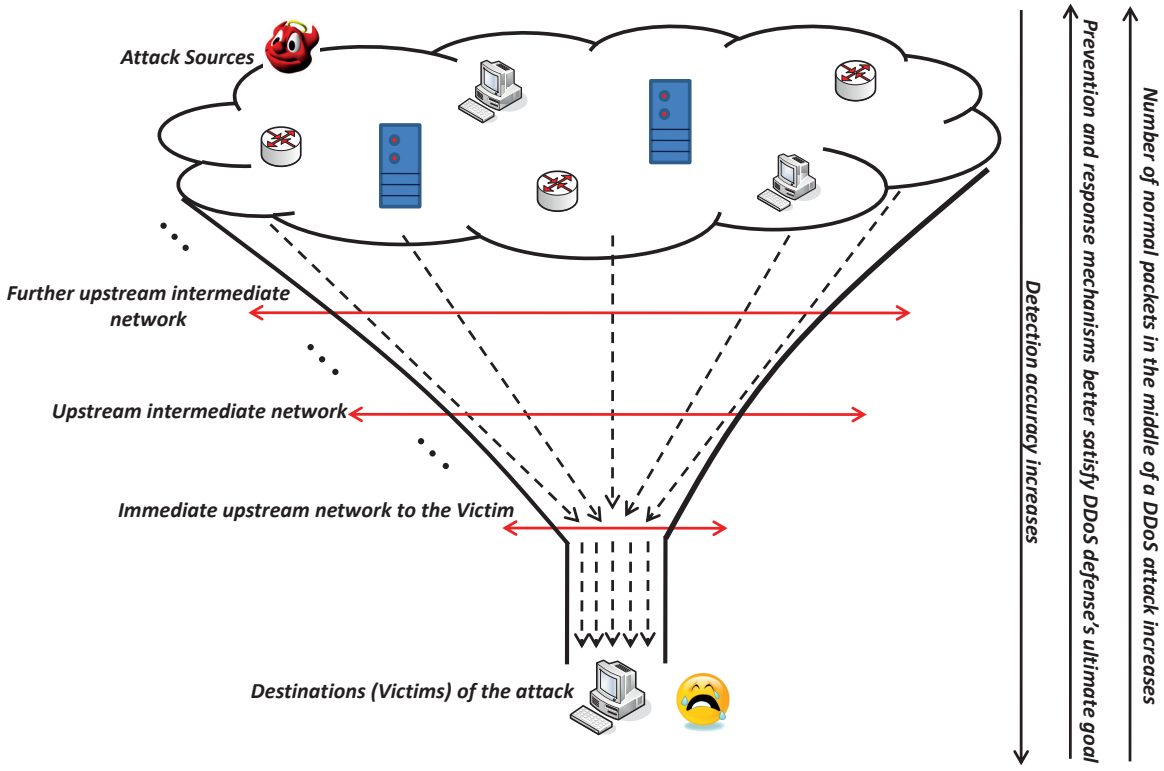

Fig. 2. Different locations for performing DDoS detection and response.

2) **Web-based (a.k.a., HTTP-based** [48]): More recently, botnets have started using HTTP as a communication protocol to send commands to the bots making it much more difficult to track the DDoS command and control structure. Web-based botnets do not maintain connections with a C&C server like IRC-based botnets do. Instead, each Web bot periodically downloads the instructions using web requests. Web-based botnets are stealthier since they hide themselves within legitimate HTTP traffic. Bots are configured and controlled through complex PHP scripts and they use encrypted communication over HTTP (port 80) or HTTPS (port 443) protocol. Three of the well-known and widely-used Web-based botnet tools are: BlackEnergy [49], Low-Orbit Ion Cannon (LOIC) [50] [4], and Aldi [52].

Some of the botnets could also provide their customers with some additional malicious services. For instance, McAfee reports that in a recent DDoS attack incident against South Korean government websites, the botnet that was used to launch the attack had employed resiliency techniques in order to evade its capture. The code also had destructive capabilities in its payload to destroy the compromised hosts, whenever required, by overwriting and deleting all the data on the hard drive [53].

V. DDoS Defense: Scope and Classification

Usually by the time a DDoS flooding attack is detected, there is nothing that can be done except to disconnect the victim from the network and manually fix the problem. DDoS flooding attacks waste a lot of resources (e.g., processing time, space, etc.) on the paths that lead to the targeted machine; hence, the ultimate goal of any DDoS defense mechanism is to detect them as soon as possible and stop them as near as possible to their sources. Figure 2 shows that detection and response can be performed in different places on the paths between the victim and the sources of the attack. As depicted in the diagram, a DDoS flooding attack resembles a funnel in which attack flows are generated in a dispersed area (i.e., sources), forming the top of the funnel. The victim, at the narrow end of a funnel, receives all the attack flows generated. Thus, it is not difficult to see that detecting a DDoS flooding attack is relatively easier at the destination (victim), since all the flows can be observed at the destination. On the contrary, it is difficult for an individual source network of the attack to detect the attack unless a large number of attack flows are initiated from that source. Obviously, it is desirable to respond to the attack flows closer to the sources of the attacks, but there is always a trade-off between accuracy of the detection and how close to the source of attack the prevention and response mechanism can stop or respond to the attack.

Moreover, the number of normal packets that reach the victims even when the victims are under a DDoS attack (i.e., in the middle of a DDoS attack) increases when response mechanisms (e.g., packet filtering) drop the attack packets closer to the sources of the attack. Otherwise, as attack flows reach closer to the victims, packet filtering mechanisms drop more legitimate packets that are destined to the victims[5].

[4]LOIC Web-based botnet has been recently used to launch DDoS flooding attack against Department of Justice (DOJ) and the Federal Bureau of investigation (FBI) [51]

[5]During large scale DDoS attacks, victims or their immediate upstream networks drop all the packets destined to the victims.

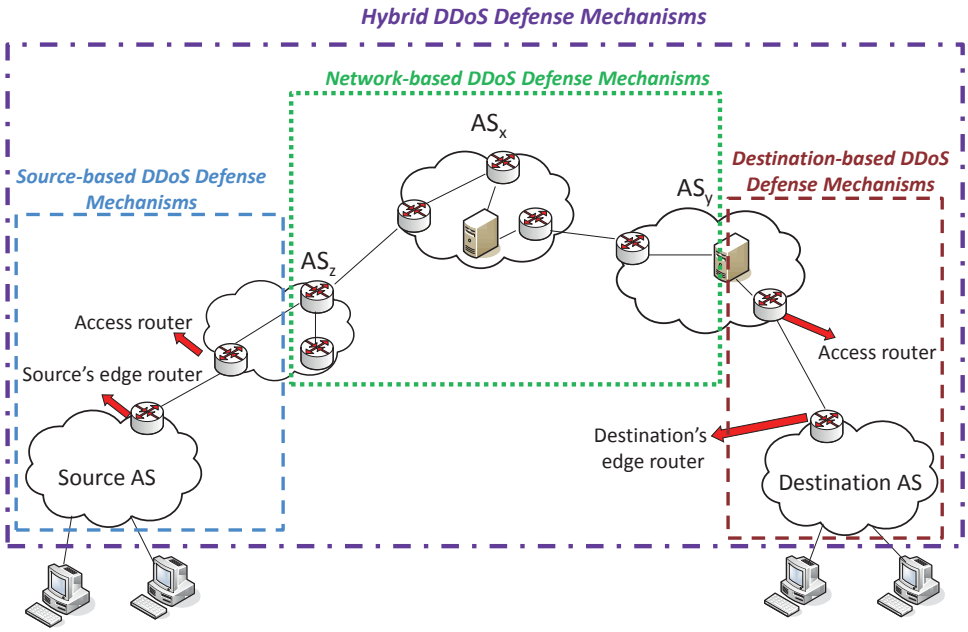

Fig. 3. A classification of the defense mechanisms against network/transport-level DDoS flooding attacks based on their deployment location in a simple network of Autonomous Systems (ASs).

Several mechanisms to combat DDoS flooding attacks have been proposed to date in the literature [2], [31]–[34], [36]. In this section we classify the defense mechanisms against two types of DDoS flooding attacks that we presented in section III using two criteria. We believe that these classification criteria are important in devising robust defense solutions. The first criterion for classification is the location where the defense mechanism is implemented (i.e., Deployment location). We classify the defense mechanisms against network/transport-level DDoS flooding attacks into four categories: *source-based*, *destination-based*, *network-based*, and *hybrid (a.k.a. distributed)* and the defense mechanisms against application-level DDoS flooding attacks into two categories: *destination-based*, and *hybrid (a.k.a. distributed)* based on their deployment location. Figure 3 shows the classification of the defense mechanisms against network/transport-level DDoS flooding attacks based on their deployment location in a simple network of Autonomous Systems (AS). There is no network-based defense mechanism against application-level DDoS flooding attacks since the application-level DDoS flooding attack traffic is not accessible at the layer 2 (switches) and layer 3 (routers) devices. Classification of DDoS defense mechanisms based on their deployment location was first presented in [1] and it is used by some other surveys as one of their classification criteria [2], [4], [8], [36]. In this paper, we extend this classification criterion by adding a *hybrid* category and analyzing several recent DDoS defense mechanisms in each category.

The second criterion for classification is the point of time when the DDoS defense mechanisms should act in response to a possible DDoS flooding attack. Based on this criterion we classify both defense mechanisms against application-level and network/transport-level DDoS flooding attacks into three categories (i.e., three points of defense against the flooding attack): *before the attack* (attack prevention), *during the attack* (attack detection), and *after the attack* (attack source identification and response) [2]. However, a comprehensive DDoS defense mechanism should include all three defenses since there is no one-size-fits-all solution to the DDoS problem. Our contribution to the last classification criterion is to classify and enumerate most of the recent defense mechanisms against DDoS flooding attacks into the aforementioned categories. Figure 4 shows the above mentioned taxonomy of the defense mechanisms against DDoS flooding attacks.

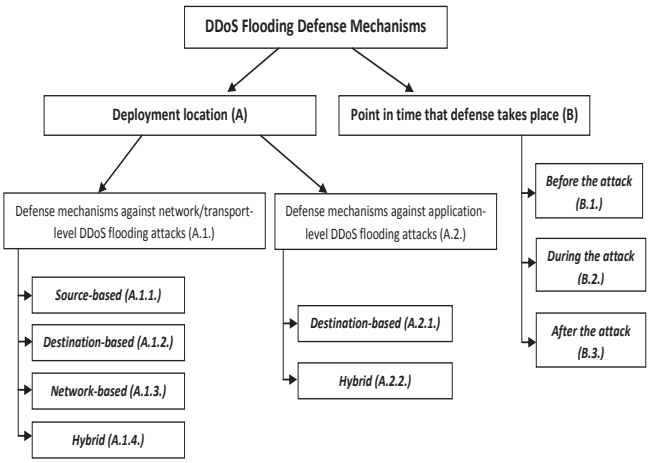

Fig. 4. A taxonomy of defense mechanisms against DDoS flooding attacks

A. *Classification based on the deployment location*

A.1. Defense mechanisms against network/transport-level DDoS flooding attacks

In the following, we discuss the defense mechanisms in each of the categories of the first classification criterion.

A.1.1. ***Source-based mechanisms:*** Source-based mechanisms are deployed near the sources of the attack to prevent network customers from generating DDoS flooding attacks. These mechanisms can take place either at the edge routers of the source's local network or at the access routers of an Autonomous System (AS) that connects to the sources' edge routers [1]. Various source-based mechanisms have been designed to defend against DDoS flooding attacks at the source; some of the major ones are as follows:

A.1.1.a. Ingress/Egress[6] *filtering at the sources' edge routers* [55]: The current IP protocol allows source hosts to alter source addresses in the IP packets. Packets with spoofed source IP addresses cause a huge problem in detecting DDOS flooding attacks. Victims cannot distinguish attack packets from legitimate ones based on source addresses. Although the IPSec protocol [56], [57] can address this problem by authenticating the source addresses of IP packets, this method is not widely deployed among service providers because of its increased overhead. Ingress/Egress filtering mechanisms have been proposed to detect and filter packets with spoofed IP addresses at the source's edge routers based on the valid IP address range internal to the network. However, the spoofed packets will not be detected if their addresses are still in the valid internal IP address range. For instance, if a packet is sent out from host i in the network M with the source address of host j, which is also valid in network M, the filtering will not detect it as a spoofed address. Furthermore, using Ingress/Egress filtering with mobile IP users, network traffic from a legitimate mobile IP (i.e., IPv.4) address has to be tunnelled in order to avoid filtering. Moreover, considering the current trend towards employing botnets to launch various attacks, attackers can still attack their targets by employing the pool of zombies with genuine IP addresses available through botnets.

A.1.1.b. D-WARD [58], [59]: This scheme aims to detect DDOS flooding attack traffic by monitoring both inbound and outbound traffic of a source network and comparing the network traffic information with predefined normal flow models. D-WARD attempts to stop attack traffic originating from a network at the border of the source network. Attack flows are identified and filtered if they mismatch the normal flow models. For instance, in the TCP protocol every packet will be acknowledged by the receiver. Hence, the normal traffic model for TCP could be defined by a maximum allowed ratio of the number of packets sent and received in the aggregate TCP flow to the peer. A normal model for all other traffic types could also be defined in a similar way. The calibration of the normal model for each kind of traffic enhances the false positive rates. Although D-WARD can generate filtering rules at the source, it consumes more memory space and CPU cycles than some of the network-based defense mechanisms. Furthermore, there is no strong incentives for the providers to employ D-WARD as it protects the others' network but not the providers' network from DDoS flooding attacks. Moreover, D-WARD can be easily bypassed by attackers who can control their traffic to be within a normal range.

[6]It is called either Ingress or Egress depending on where you stand in the network and apply the filters.

A.1.1.c. MUlti-Level Tree for Online Packet Statistics (MULTOPS) [60] *and Tabulated Online Packet Statistics (TOPS)* [61]: MULTOPS is a heuristic and a data-structure that network devices (e.g., routers) at the source subnet can use to detect and filter DDoS flooding attacks. Normally the rate of traffic in one direction is proportional to that in the opposite direction during normal operations on the Internet [60]. Hence, a significant difference between the rates of traffic going to and coming from a host or subnet can indicate that the network prefix is either the source or the destination of an attack. MULTOPS detects and filters DDoS flooding attacks based on this mechanism. One major drawback of MULTOPS is that it uses a dynamic tree structure for monitoring packet rates for each IP address which makes it a vulnerable target of a memory exhaustion attack [60]. An alternative approach called TOPS [61] provides an efficient method for detecting packet flow imbalances based on a hashing scheme that uses a small set of field length lookup tables. TOPS can improve the accuracy and reduce the false alarm rate of the system by monitoring traffic by protocol, and maintaining a probability distribution of traffic flow rates. Furthermore, TOPS's efficiency and accuracy makes it suited for implementation in the routers.

Both MULTOPS and TOPS are based on the assumption that incoming and outgoing traffic rates are proportional, which is not always the case. For instance, rates for multimedia streams are not proportional and usually the traffic rates from the servers are significantly higher than the ones from clients. Hence, MULTOPS and TOPS can have high false negative rates. Furthermore, attackers can increase the proportion of the incoming and outgoing traffic rates legitimately (e.g., downloading large files from different ftp servers through a number of genuine sources); hence, during the attack, the attack traffic will be undetected because of the similarity of the attack traffic rates to the normal traffic rates which have been legitimately increased.

A.1.1.d. MANAnet's Reverse Firewall [62]: As opposed to a traditional firewall, which protects a network from incoming packets, the reverse firewall protects the outside from packet flooding attacks that originate from within a network. A reverse firewall limits the rate at which it forwards packets that are not replies to other packets that recently were forwarded in the other direction. Of course, it must be possible to send some packets that are not replies, for instance, to start a new conversation. However, such packets must not be transmitted at a high rate. One of the main disadvantages of the reverse firewall is that it is manual and requires the administrators' involvement. Furthermore, the reverse firewall's configuration cannot be dynamically changed at runtime. Moreover, there is no benefit (e.g., financial gain) for the source networks to deploy costly reverse firewalls since there is no benefit for the source networks.

Source-based defense mechanisms aim to detect and filter the attack traffic at the sources of the attack; however, they are not entirely effective against DDoS flooding attacks. There are three main reasons which make these mechanisms a poor choice against DDoS flooding attacks. First, the sources of the attacks can be distributed in different domains making it difficult for each of the sources to detect and filter

attack flows accurately. Second, it is difficult to differentiate between legitimate and attack traffic near the sources, since the volume of the traffic may not be big enough as the traffic typically aggregates at points closer to the destinations. Finally, the motivation for deployment of the source-based mechanisms is low since it is unclear who (i.e., customers or service providers) would pay the expenses associated with these services. Hence, pure source-based mechanisms are not efficient and effective against DDoS flooding attacks.

A.1.2. ***Destination-based mechanisms:*** In the destination-based defense mechanisms, detection and response is mostly done at the destination of the attack (i.e., victim). There exist various destination-based mechanisms that can take place either at the edge routers or the access routers of the destinations' AS. These mechanisms can closely observe the victim, model its behavior and detect any anomalies. Some of the major destination-based DDoS defense mechanisms are as follows:

A.1.2.a. IP Traceback mechanisms [63]: The process of tracing back the forged IP packets to their true sources rather than the spoofed IP addresses that was used in the attack is called traceback. There are various IP traceback mechanisms that have been proposed to date [64]. These mechanisms can be classified into two main categories. The first category is ***packet marking*** mechanisms [64]–[66]. Usually routers in the path to the victim mark packets (i.e., add routers' identification to each packet) so that the victim can identify the path of attack traffic and distinguish it from legitimate traffic after the detection. However, storing the entire path in the IP identification field of each packet needs certain coding schemes and these schemes sometimes are not able to assign each mark to a unique path; hence, false positive rates of these mechanisms are still high. In other words, legitimate packets could be treated as attack packets. The second category is ***link testing*** mechanisms [67], [68] in which the traceback process usually starts from the router closest to the victim and iteratively tests its upstream links until it can be determined which link is used to carry the attacker's traffic (i.e., the traceback process is recursively repeated on the upstream router until the source is reached).

All of the traceback mechanisms have serious deployment and operational challenges [69]. One of the fundamental deployment and operational challenges is ensuring a sufficient number of routers that support traceback before it is effective. Moreover, attackers can also generate traceback messages; consequently, some form of authentication of traceback messages is necessary. Furthermore, most of the traceback mechanisms have heavy computational, network or management overheads [69].

A.1.2.b. Management Information Base (MIB) [70]: MIB data is comprised of parameters that indicate various packet and routing statistics. Continuously analyzing MIB can help victims to identify when a DDoS attack is occurring. During a DDoS attack, it is possible to map ICMP, UDP, and TCP packets' statistical abnormalities to a specific DDoS attack by identifying statistical patterns related to different parameters [71], [72]. Furthermore, this mechanism can also provide ways to adjust network parameters to compensate for the unwanted traffic (e.g., adding more resources to the target network) [70]. Although analysing MIB data sounds promising in detecting DDoS attacks, its effectiveness needs to be further evaluated in a real network environment.

A.1.2.c. Packet marking and filtering mechanisms [73], [74], [76]: These mechanisms aim to mark legitimate packets at each router along their path to the destination so that victims' edge routers can filter the attack traffic. These mechanisms let the receivers install dynamic network filters to block the undesirable traffic. Packet filtering mechanisms are dependent in part on the strength of the attackers, and when it increases, filters become ineffective and they cannot properly be installed. Several destination-based packet filtering mechanisms have been proposed in the literature so far. Here we briefly describe some of these proposed mechanisms:

History-based IP filtering [73]: By employing this mechanism, victims can filter bandwidth (flooding) attack traffic according to the history they have maintained while they were not under attack. In particular, the target destination can use an IP address database to keep all the IP addresses that frequently appear at the target. During a bandwidth attack, the target only admits the packets whose source IP addresses belong to the IP address database. This technique helps destination hosts in resource management when their links to the upstream network becomes a bottleneck during a DDoS flooding attack. However, any large-scale DDoS attack that simulates normal traffic behaviour will defeat such mechanisms.

Hop-count filtering [74]: In this mechanism, information about a source IP address and its corresponding hops from a destination are recorded in a table at the destination side when the destination is not under attack. Once an attack alarm is raised, the victim inspects the incoming packets' source IP addresses and their corresponding hops to differentiate the spoofed packets. It is not necessary for routers to collaborate mutually in this mechanism; however, it is difficult to ensure the integrity and accuracy of the source IP addresses and their corresponding hops from the victim. In other words, attackers can spoof IP addresses with the same hop-count as their machines do. Moreover, legitimate packets can be identified as spoofed ones if their IP to hop-count mappings are inaccurate or if the hop-count updates has a delay [75].

Path Identifier (Pi) [76]: In this mechanism, a path fingerprint is embedded in each packet which enables a victim to identify packets traversing the same paths through the Internet on a per packet basis, despite of the source IP address spoofing. Pi is a per-packet deterministic mechanism which means that each packet that is travelling along the same path carries the same identifier. This feature allows the victim to employ the Pi mark to filter out packets matching the attackers' identifiers on a per packet basis which is considered as a proactive role in defending against a DDoS attack. Pi is effective if about half of the routers in the Internet participate in packet marking [76]. The main disadvantage of Pi is that the limitation on the size of identification field may result in the same path information representing different paths. This can decrease the performance of Pi mechanism (i.e., increased false positive/false negative rates).

A.1.2.d. Packet dropping based on the level of congestion: These destination-based DDoS defense mechanisms drop

suspicious packets when the network links are congested to a certain level. Packetscore [77] is an example of this type.

Packetscore, proposed by *Kim et al.* in [77], is an automated attack characterization, selective packet discarding and overload control mechanism. The key idea is to prioritize packets based on per packet score which estimates the legitimacy of a packet given the attribute values it carries. Then, once the score of a packet is computed at Detecting-Differentiating-Discarding routers (3D-R) by employing a Bayesian-theoretic metric, a score-based selective packet discarding method at the destination is performed. The dropping threshold for the packet discarding method is dynamically adjusted based on (1) the score distribution of recent incoming packets and (2) the current level of overload of the system. However, *Kim et al.* did not provide any results to show how the time-scale of updates of the scorebooks, score cumulative distribution function (CDF), and dynamic discarding threshold could impact the response time and the decision of their proposed selective packet discarding scheme when subjected to more orchestrated synchronized DDoS attacks.

Most of the destination-based mechanisms cannot accurately detect and respond to the attack before it reaches the victims and wastes resources on the paths to the victims; hence, they are not capable of detecting and responding to the DDoS attack traffic properly. Therefore, network-based DDoS defense mechanisms have been proposed to address this problem and to help both source and destination based mechanisms to carry out their duties more accurately.

A.1.3. ***Network-based mechanisms:*** These mechanisms are deployed inside networks and mainly on the routers of the ASs [78]. Detecting attack traffic and creating a proper response to stop it at intermediate networks is an ideal goal of this category of defense mechanisms. Some of the main network-based DDoS defense mechanisms are as follows:

A.1.3.a. Route-based packet filtering [79], [80]: Route-based packet filtering extends ingress filtering to the routers at the core of the Internet. The traffic on each link in the core of the Internet usually originates from a limited set of source addresses. Hence, if an unexpected source address appears in an IP packet on a link, then it is assumed that the source address has been spoofed, and hence the packet can be filtered. However, this mechanism is ineffective against DDoS attacks if attackers either use genuine IP addresses instead of spoofed ones or spoof with carefully chosen source IP addresses that are not going to be filtered.

A.1.3.b. Detecting and filtering malicious routers [81]: Routers are continuously targeted and compromised. They can be leveraged to empower DDoS attacks. A range of specialized anomaly detection protocols have been proposed to detect malicious routers involved in packet forwarding between routers. For instance, *Watchers* [82] detects misbehaving routers that launch DDoS attacks by absorbing, discarding or misrouting packets. It uses the conservation of flow principle to examine flows between neighbors and endpoints. In [83], some of the implicit assumptions of *Watchers* that do not hold have been modified so the cost and complexity of *Watchers* algorithm is increased. Even with these modifications, *Watchers* requires explicit communication among the routers. Furthermore, it cannot detect spoofed packets. Even worse, such packets could be used by the attacker to misidentify a target as a bad router. *Watchers* can only detect compromised routers and it is vulnerable to misbehaving hosts. It also assumes that every router knows the topology of the network, which is not a feasible assumption for large networks.

There are other mechanisms in which the detection and filtering of malicious routers is based on routers' trustworthiness [84], [85]. For instance in [84], a Bayesian inference model has been employed to evaluate the trustworthiness of an access router with regards to forwarding packets without modifying their source IP addresses. In this approach, the trust values for the access routers are computed by a router (judge) that samples all the traffic being forwarded by the access routers. Employing trust calculations, decision making, and trust negotiations among the routers in order to efficiently and effectively detect and filter the malicious routers is the ultimate goal of these mechanisms.

Network-based mechanisms usually lead to high storage and processing overheads at the routers. These overheads get even worse if each router does redundant detection and response through the path to the destination [86], which can present a significant burden. Various researchers have proposed different approaches to reduce the amount of storage and consumption of CPU cycles for detection and response at the routers such as Bloom filters [78] [87], Packet sampling [88], etc. But these approaches are not sufficient when routers still do redundant jobs. Moreover, reducing the amount of redundant detection and response between the routers requires coordination among them [86]. Different communication protocols have been proposed to coordinate attack detection and response among the routers [1]. However, network-based defense mechanisms that have been proposed thus far are not effective and efficient because of their large overhead of network communication. For instance, the lack of bandwidth during DDoS attacks may limit the protocol for communication and cause network-based mechanisms to fail.

A.1.4. ***Hybrid (Distributed) mechanisms:*** In most of the previously discussed categories of DDoS flooding defense mechanisms (source-based, destination-based, and network-based), there is no strong cooperation among the deployment points. Furthermore, detection and response is mostly done centrally either by each of the deployment points (e.g., source-based mechanisms) or by some responsible points within the group of deployment points (e.g., network-based mechanisms). Hence, we call these categories of DDoS defense mechanisms *centralized*. As opposed to centralized defense mechanisms, hybrid defense mechanisms are deployed at (or their components are distributed over) multiple locations such as source, destination or intermediate networks and there is usually cooperation among the deployment points. For instance, detection can be done at the victim side and the response can be initiated and distributed to other nodes by the victim. Some of the hybrid DDoS defense mechanisms are as follows:

A.1.4.a. Hybrid packet marking and throttling/filtering mechanisms: All of the previously presented marking and filtering mechanisms place the attack detection module and the

packet filtering module at the same location. Hybrid packet throttling mechanisms usually place the attack detection modules near the victims and execute packet filtering close to the attack sources. In some of these mechanisms, victims under attack install a router throttle at upstream routers several hops away in order to limit the forwarding rate of the packets destined to those victims. Basically these mechanisms are packet filtering infrastructures that are leveraging the routers' support to filter out DDoS flows. It is important to note that these mechanisms only limit the rate of malicious packets and do not harm legitimate flows. Here, we briefly describe some of these mechanisms:

Aggregate-based Congestion Control (ACC) [89] *and Pushback* [89], [90]: ACC rate limits the aggregates rather than IP sources. Aggregates are subsets of traffic defined by some characteristics such as specific destination port or source IP address. In ACC, routers detect aggregates that are overwhelming them by using samples of packet drops in their queues. Then they send a pushback message to the upstream routers along with the information about the aggregates to request a rate limit by presenting a rate limit value. If the aggregate packets respect the rate limit they can still go through their path to their destinations, otherwise they are dropped to limit their rate and pushback messages are propagated to the upstream routers [89] [90]. Both ACC and Pushback are not effective against uniformly distributed attack sources because of voluminous traffic.

Attack Diagnosis (AD) and parallel-AD [91]: AD combines the concepts of pushback and packet marking. A victim host activates AD, after an attack has been detected, by sending AD-related commands to its upstream routers. Upon receiving such commands, the AD-enabled upstream routers deterministically mark each packet destined for the victim with the information about the input interface that processed that packet. The victim can then traceback the attack traffic to its source by employing the router interface information which is recorded in the packet marking process. Then, the victim issues messages that command AD-enabled routers to filter attack packets close to the source after the traceback is completed. The AD commands can be authenticated by the TTL field of the IP header without relying on any global key distribution infrastructure in the Internet. AD is not effective against large-scale attacks. An extension to AD, called Parallel AD (PAD), has been proposed to address AD's shortcomings against large-scale attacks. The main difference between PAD and AD is that PAD can diagnose and stop the traffic from more than one router at the same time. Hence, PAD is capable of throttling traffic coming from a large number of attack sources at the same time [91].

TRACK [92]: This mechanism also combines IP traceback, packet marking, and packet filtering. TRACK is composed of two components: *router port marking module* and *packet filtering module*. The router port marking module marks packets by probabilistically writing a router interface's port number, a locally unique 6-digit identifier, to the packets it transmits. Upon receiving the packets marked by each router in an attacking path, a victim machine can then use the information contained in those packets to trace the attack back to its source. Then, the packet filtering component employs the information contained in the same packets to filter the malicious packets at the upstream routers, thus effectively mitigating attacks. One of the main advantages of TRACK over previously discussed throttling mechanisms is its low communication and computation overhead [92]. TRACK has some limitations to be addressed in the future:

(i) Attackers can modify the marking fields of the packets in order to avoid being located; hence, TRACK is still vulnerable to these attacks; and

(ii) TRACK is claimed to be an effective approach for IP traceback but not effective for attack traceback in which the objective is to identify the real attacker that ordered the zombies to launch the DDoS attack. In other words, TRACK cannot find the IP addresses of the attackers that actually launch the attack.

A.1.4.b. DEFensive Cooperative Overlay Mesh (DEFCOM) [93]: DEFCOM is a distributed framework to enable information and service exchange among all of the defense nodes. It attempts to shift from isolated defense architectures towards a distributed framework of heterogeneous defense nodes in which all the nodes collaborate and cooperate to achieve an effective defense . For instance, since attack detection is best done near the victim and response is most effective at the source of the attack, defense nodes should be specialized for different aspects of the defense; hence, different nodes in their proposed distributed architecture are responsible for their specialized defense ability and *they must be able to communicate with others* in order to successfully detect and respond to the attacks. Each node in their proposed framework must at least support the followings: (i) *Attack alerts* generated from the alert generators should be sent to the rest of the network; (ii) *Rate-Limit requests* should be sent upstream; (iii) *Resource requests* that each node issues should be sent to its downstream neighbors; and (iv) *Traffic Classification* nodes must communicate with their downstream neighbors to ensure that the bulk of legitimate traffic will not be dropped.

DEFCOM is comprised of heterogeneous and distributed defense nodes organized into a P2P network where the nodes are communicating with each other to achieve cooperative defense. The P2P structure and topology construction of the DEFCOM defense nodes allows for the approximation of underlying topology; hence, when an attack alert is raised, it will be possible to discover victim-rooted traffic tree. Then, upstream and downstream relationships among peers is identified and proper rate limits to control the attack traffic is created and placed as close as possible to the sources of the attack. At the same time, classifier nodes will differentiate legitimate traffic from attack traffic.

Classifier nodes in the DEFCOM require an in-line deployment and their malfunction deters a wide deployment of classifier functionality in the network. Therefore, there will be no means to verify if the traffic which has been received by a node is legitimate or attack and rate limiters severely rate-limit all the traffic coming from these sources. If it is assumed that a significant portion of networks in the Internet will be legacy networks, this results in a DoS attack to the legitimate clients from legacy networks during attacks; this is the main disadvantage of the DEFCOM framework.

A.1.4.c. COSSACK [94]: This mechanism is built on all of the border routers of the edge networks, with the core software system called *watchdog*. This mechanism is based on a number of assumptions. First, the border routers are assumed to have ingress/egress filtering mechanisms implemented. Second, border routers can prevent IP spoofing by employing ingress/egress filtering. The final critical set of assumptions *Papadopoulos et al.* made in [94] consist of the existence of an attack signature, the capability of border routers to filter packets based on the signature, and the connection availability between *watchdogs* [94]. Therefore, with all the supports from different implemented components that they assume, the application layer *watchdog* could multicast attack notifications from the victim side to the source side and stop DDoS attack flows around the source employing the ingress/egress filtering mechanisms implemented on the border routers. COSSACK is, however, unable to handle attacks from legacy networks that do not deploy COSSACK defense mechanism.

A.1.4.d. Capability-based mechanisms [95]: These mechanisms let the destination explicitly authorize the traffic it desires to receive (e.g., *Portcullis* [96], *Traffic Validation Architecture (TVA)* [97], [98], and *Stateless Internet Flow Filter (SIFF)* [99]). In most of these mechanisms, senders obtain the capabilities, which are short-term authorizations, from the receivers and put them as stamps on their packets. Then, the verification points along the path check if the traffic is certified as legitimate or not. For instance, in the capability-based architecture presented in [95], sources must first obtain ”permission to send” from the destination. Basically, destination provides tokens, or capabilities, to sources whose traffic it agrees to accept. The sources then include these tokens in their packets. The verification points are distributed around the network to check if the traffic has been certified as legitimate by destinations and the path in between, and to discard unauthorized traffic. Privileged packets are prioritized at the routers and most of the bandwidth is allocated to them so that they are never dropped by unprivileged packet flooding. Privileged packets flooding rarely happens, since receivers can stop any undesirable flow by not sending the capability back to the senders.

SIFF architecture, which includes a handshake protocol that is used by sender to obtain capabilities in order to send privileged traffic, is similar to the capability-based architecture presented in [95]. However, both the approach in [95] and SIFF are vulnerable to flooding attacks against capability setup channel and attacks that flood the receiver using already acquired capabilities.

TVA addresses the limitations of capability mechanisms such as SIFF. TVA has the same process for acquiring a capability as SIFF but different format of capabilities are used in TVA [97], [98]. There are several disadvantages for the TVA mechanism:

(i) TVA, like previous capability-based mechanisms, assumes that the receiver can distinguish the attack traffic from the legitimate traffic. Hence, its effectiveness depends on the accuracy of the attack detection mechanism that a receiver is employing.

(ii) TVA needs a significant amount of per flow state information to be maintained at each router, and

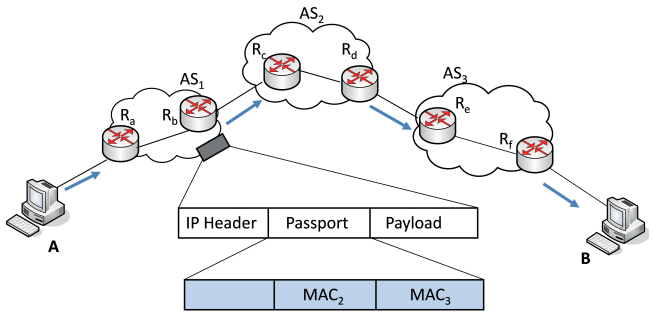

Fig. 5. A border router of a source AS (R_b) stamps source authentication information into the Passport header of an outbound packet. A border router of an intermediate or destination AS (R_c or R_e) verifies this information [100].

(iii) specific DDoS attacks are still possible even after all the routers implement TVA.

Capability-based mechanisms are always active and their processing and memory costs (overhead) are high. Furthermore, as we mentioned earlier, in order to prove the effectiveness of the capability-based mechanisms, one must first suggest a practical way to secure the capability setup channel, as well as a efficient algorithm for choosing what capabilities to offer to unknown sources; these are both challenging problems to address.

The issue of securely granting capabilities has been addressed by the source authentication systems that have been proposed in recent capability-based mechanisms such as *Passport* [100] and *TVA+* [101]. The Passport system uses efficient symmetric-key cryptography to put tokens on packets that allow each AS along the network path to independently verify that a source address is valid. Passport employs the routing system to efficiently distribute the symmetric keys that are used for verification. Figure 5 shows how the border router of a source AS (R2) stamps source authentication information into the Passport header of an outbound packet. By adopting Passport, Internet Service Providers (ISPs) can protect their own addresses from being spoofed at each other’s networks; hence, ISPs have stronger incentives to deploy Passport than alternatives such as ingress filtering. Nevertheless, when attackers can get capabilities from colluders, the capability-based mechanisms such as Passport become ineffective [102]. For instance, an off-path attacker who spoofs a legitimate sender’s address and who is in the same subnet as the sender can obtain the capabilities by eavesdropping and inject the attack packets using the capabilities. Another disadvantage of Passport is that it only prevents hosts in one AS from spoofing the IP addresses of other ASs. Hence, attackers can spoof the IP address of any host within the same AS [75]. Passport requires each border router to store AS paths for all the destination prefixes, and shared secret keys with all ASs; this approach would require a considerably large amount of memory for Internet-wide deployment [75].

A.1.4.e. Active Internet Traffic Filtering (AITF) as a filter-based (datagram) mechanism [103]: Capability-based mechanisms enable a receiver to deny by default all the traffic and explicitly accept only the traffic that belongs to

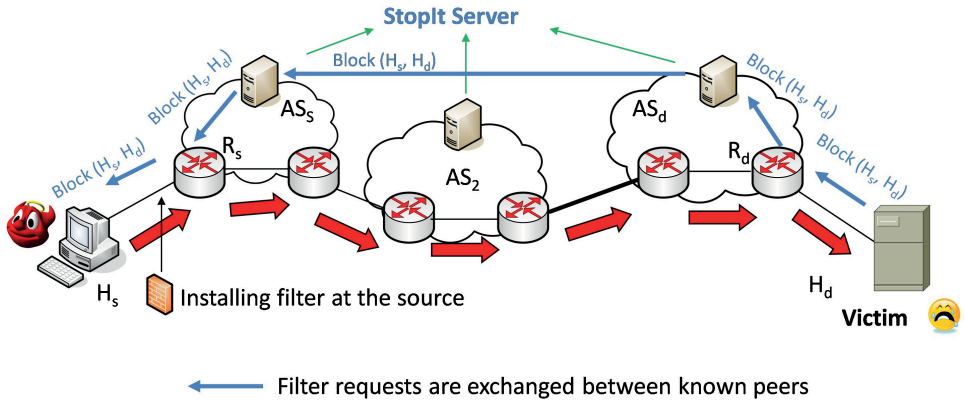

Fig. 6. StopIt architecture: How it installs filters on the sources of attack upon detecting the DDoS attack [95].

established network-layer connections. The alternative could be the datagram (*a.k.a.* filtering) mechanism in which a receiver accepts by default all the traffic and explicitly denies the traffic that has been identified as undesirable. The datagram mechanism requires a credible, bounded amount of filtering resources from participating ISPs, which offers incentives to ISPs to deploy it.

AITF is a hybrid DDoS defense mechanism which enables a receiver to contact misbehaving sources and ask them to stop sending it traffic. Each of the sources that have been asked to stop is policed by its own ISP, which ensures their compliances. Each ISP that hosts misbehaving sources must either support AITF mechanism (i.e., accept to police its misbehaving clients), or risk losing all of its access to the complaining receiver; this provides a strong incentive for all the ISPs to cooperate; especially when the receiver is a popular point of access. AITF preserves receiver's bandwidth in the face of DDoS flooding attacks at a per-client cost; thus, it is affordable for the ISPs to employ it. *Argyraki et al.* in [103] showed that even the first two networks that would deploy AITF could maintain their connectivity to each other in the face of DDoS flooding attack. AITF verifies the legitimacy of a filter request using a three-way handshake. However, if the flooded link is outside a victim's AS, the three-way handshake may not complete because the handshake packets traverse the same flooded link as the attack traffic does, and filters may not have been installed. Furthermore, AITF has several deployment problems since it relies on the routers, which are in the middle of the network (during initial deployment), to perform the actual filtering [104]. It also depends on various IP route records to determine where packets come from [104].

A.1.4.f. StopIt [101] is a hybrid filter-based DDoS defense mechanism that enables each receiver to install a network filter that blocks the undesirable traffic it receives. StopIt uses Passport as its secure source authentication system to prevent source address spoofing. Its design employs a novel closed-control and open-service architecture to battle strategic attacks that aim to prevent filters from being installed and to provide the StopIt service to any host in the Internet. Figure 6 shows the StopIt architecture and how a destination H_d installs a filter to block the attack flow (H_s,H_d) from a source H_s. Each cloud represents an AS boundary. Each AS has a StopIt server that sends and receives StoptIt requests, and hosts can only send StopIt requests to their access routers (e.g., R_s, R_d). Based on the studies in [101], StopIt outperforms filter-based designs such as AITF, and is effective in providing continuous non-interrupted communication under a wide range of DDoS attacks. However, StopIt does not always outperform capability-based mechanisms. For instance, if the attack traffic does not reach a victim, but congests a link shared by the victim, a capability-based mechanism (e.g., TVA) is more effective. Therefore, both filters (*a.k.a.* datagram) and capabilities are highly effective DDoS defense mechanisms, but neither is more effective than the other against DDoS flooding attacks.

StopIt mechanism is vulnerable to the attacks in which attackers flood the routers and StopIt servers with filter requests and packet floods. In order to prevent these attacks, the StopIt framework must ensure that a router or a StopIt server only receives StopIt requests from local nodes in the same AS, or another StopIt server. In doing so, network administrators must manually configure the routers and StopIt requests with the list of hosts, routers, and other StopIt servers. Such manual configuration for an AS with hundreds of thousands of nodes is a burdensome task [75]. Furthermore, StopIt needs complex verification/authentication mechanisms, and misbehaving StopIt server detection mechanisms to be implemented in both hosts and routers which makes it a challenging mechanism to deploy and manage in practice [75].

In [102], [105], TVA as a capability-based mechanism is compared to StopIt as a filter-based mechanism under similar assumptions and practical constraints. They compare six different DDoS flooding mitigation systems, including TVA and StopIt. They use simulations on realistic topologies and cover different attack strategies in their research. The outcome of their research is summarized in Figure 7 [102], [105]. In Figure 7, *attacks' power* is a generalized term which is defined in [105] based on: number of attackers (i.e., number of bots), and size of attack (i.e., packet size). As the number of attackers and their packet sizes increase the attack power increases. Effectiveness is also measured [105] based on the legitimate hosts' TCP transfer performance (i.e., percentage of completed TCP transfers). As the number of completed transfers increases the mechanisms that have been

TABLE I
SUMMARY OF FEATURES, ADVANTAGES, AND DISADVANTAGES OF DEFENSE MECHANISMS AGAINST NETWORK/TRANSPORT-LEVEL DDOS FLOODING ATTACKS BASED ON THEIR DEPLOYMENT LOCATION

		Features	Disadvantages	Advantages
Centralized	***Source-based***	Detection and response are deployed at the source hosts	Sources are distributed among different domains; hence, it is difficult for each of the sources to detect and filter attack flows accurately Difficult to differentiate legitimate and DDoS attack traffic at the sources, since the volume of the traffic is not big enough Low motivation for deployment; since, it is unclear who would pay the expenses associated with these services	Aims to detect and respond (i.e., filter) to the attack traffic at the source and before it wastes lots of resources
	Destination-based	Detection and response are deployed at the destination hosts (i.e., victims)	They cannot accurately detect and respond to the attack before it reaches the victims and wastes resources on the paths to the victim	Easier and cheaper than other mechanisms in detecting DDoS attacks because of their access to the aggregate traffic near the destination hosts
	Network-based	Detection and response are deployed at the intermediate networks (i.e., routers)	High storage and processing overhead at the routers Attack detection is difficult because of the lack of availability of sufficient aggregated traffic destined for the victims	Aims to detect and respond to (i.e., filter) the attack traffic at the intermediate networks and as close to source as possible
Distributed	***Hybrid (Distributed)***	Detection and response are deployed at various locations: detection usually occurs at destinations & intermediate networks, and response usually occurs at the sources & upstream routers near the sources There is a cooperation among various defense components	Complexity and overhead because of the cooperation and communication among distributed components scattered all over the Internet Lack of incentives for the service providers to cooperate/collaborate Need trusted communication among various distributed components in order to cooperate/collaborate	More robust against DDoS attacks More resources at various levels (e.g., destination, source, and network) are available to tackle DDoS attacks

employed are more effective. As *Yang et al.* showed in this figure, when the attackers' power is low, both filters and capabilities work well, although filters work slightly better. As the attackers' power increases, filters become ineffective when they cannot be properly installed, and then capabilities become ineffective when attackers can get capabilities from colluders. When the attackers' power is extremely high, both filters and capabilities become ineffective, and there should be some fail-safe mechanisms (e.g., fair queuing) in place to resolve the problem.

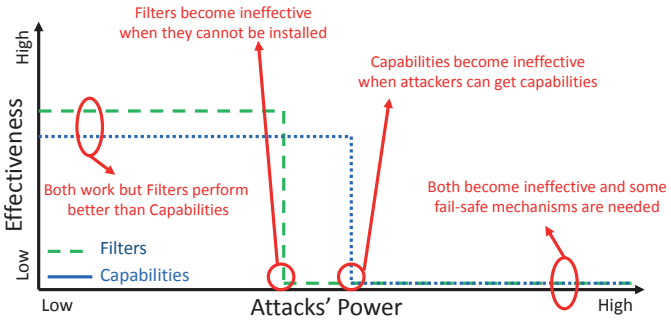

Fig. 7. Capability-based (capabilities) vs. datagram-based (Filters) mechanisms [102].

Discussion: Since attackers cooperate to perform successful attacks, defenders must also form alliances and collaborate with each other to defeat the DDoS attacks. The DDoS defense community is currently more involved in proposing novel hybrid DDoS defense mechanisms and most of the recently proposed mechanisms belong to the hybrid category. No single deployment point (centralized) can successfully defend against DDoS because of the fundamental challenges we enumerated for each of the deployment points. A hybrid (*Distributed*) defense mechanism is the best way to combat DDoS Attacks. We have enumerated various hybrid defense mechanisms in this section; they are comprised of multiple defense nodes deployed at various locations that cooperate with each other towards attack prevention, detection, and response. Detecting DDoS attack as soon as possible and before it reaches the victims, identifying the attack sources, and finally stopping the attack as close as possible to the attack sources is the ultimate goal of DDoS defense mechanisms; we strongly believe that this can be best achieved through hybrid (*Distributed*) DDoS defense mechanisms.

Combining source address authentication (to prevent IP spoofing), capabilities, and filtering would be the most effective and efficient solution because of the robustness of capabilities and the relative simplicity of a capability-based design. However, there will be a trade-off between performance and accuracy in any DDoS defense

solution and the goal is to minimize the gap between performance and accuracy. Table I summarizes the features of the four categories of defense mechanisms against network/transport-level DDoS flooding attacks that we classified in this section and enumerates the advantages and disadvantages of each category.

A.2. ***Defense mechanisms against application-level DDoS flooding attacks***

In the following, we discuss the defense mechanisms against application-level DDoS flooding attacks in each of the categories of the first classification criterion.

A.2.1. ***Destination-based (server-side) mechanisms:*** Most of the application layer protocols are organized in terms of *client-server* model. A *server* is a process that implements a specific service (e.g., DNS server, Web server). A *client* is a process that requests a service from a server. As we mentioned earlier, destination-based defense mechanisms are deployed at the destination of the attack (i.e., victim), which is the *server* of the application layer protocols' client-server model or the reverse proxy[7] when we consider a web cluster hosting different web applications. Most of these mechanisms closely observe the server and model its clients' behavior so that they can detect any anomalies and drop or rate limit the malicious requests. Some of these major mechanisms against application-level DDoS flooding attacks are as follows:

A.2.1.a. Defense against Reflection/Amplification attacks: [106], [108]: Defense mechanisms against reflection attacks (IP spoofing) have been already discussed in section *A.1.1.a*. Most of the defense mechanisms against amplification attacks are deployed at the server-side and their aim is to detect malicious traffic from different protocols such as DNS and SIP by employing various mechanisms such as machine learning techniques. Here, we review two mechanisms proposed to defend amplification attacks for the DNS and SIP application-level protocols. *Kambourakis et al.* in [106] propose a DNS Amplification Attacks Detector (DAAD) mechanism in which they collect the DNS requests and replies using IPtraf tool [107]. Then, their DAAD tool processes the captured network data, which are stored in the appropriate MySQL database, on-the-fly, classifies the requests/replies as suspicious or not and generates the corresponding alert to block the DNS requests/replies in the case of an undergoing attack [106]. In order to detect and block VoIP flooding attacks specifically and SIP protocol flooding attacks in general *Rahul et al.* employ a genetic algorithm to recognize the authorized users; then, their proposed VoIP Flood Detection System (VFDS) is used to detect TCP flooding attacks and SIP flooding attacks on SIP devices using their Jacobian Fast and Hellinger distance algorithms. In their mechanism the Jacobian Fast algorithm has been used to fix the threshold limit and Hellinger distance calculation, which is a statistical anomaly based algorithm, has been used to detect deviations in traffic [108].

A.2.1.b. DDoS-Shield [3], [109]: This mechanism uses statistical methods to detect characteristics of HTTP sessions and employs rate-limiting as the primary defense mechanism. DDoS-Shield consists of a suspicion assignment mechanism and a DDoS-resilient scheduler. The suspicion assignment mechanism assigns a continuous value as opposed to a binary measure, which have been assigned by previous mechanisms, to each client session. The DDoS-resilient scheduler acts as a rate-limiter and utilizes continues values, assigned by the suspicion assignment mechanism, to determine if and when to schedule a session's requests. However, it is not clear if a legitimate session is given another chance to receive the service if it is dropped by the DDoS-resilient scheduler.

A.2.1.c. Anomaly detector based on hidden semi-Markov model [110]: *Xie et al.* propose an anomaly detector based on hidden semi-Markov model to describe the dynamics of the access matrix and to detect the attacks. They use the entropy of document popularity fitting to the model to detect the potential application-layer DDoS attacks. The main disadvantage of this mechanism is the high complexity of its algorithm.

A.2.1.d. DAT (Defense Against Tilt DDoS attacks) [111]: This mechanism monitors users' features (e.g., instant traffic volume, session behavior, etc.) throughout a connection session to determine whether users are malicious or not. For different users' bahaviors, DAT provides differentiated services.

A.2.2. ***Hybrid (Distributed) mechanisms:*** Hybrid defense mechanisms are those mechanisms that employ collaboration/cooperation between clients and servers to detect and respond to the attacks. For instance, detection is done at the victim (web server/reverse proxy) and the response is initiated and distributed to the client-sides by the victim. Some of the hybrid mechanisms against application-level DDoS flooding attacks are as follows:

A.2.2.a. Speak-up [112]: The way this mechanism tries to decrease the number of malicious requests is to encourage all the clients to automatically send higher volumes of traffic. The reasoning behind this approach is that attackers are already using most of their upload bandwidth so cannot react to the encouragement. On the other hand, good clients have spare upload bandwidth and will react to the encouragement by increasing their traffic volumes drastically. The goal of this mechanism is that the good clients crowd out the bad ones, thereby capturing a much larger fraction of the server's resources than before. Speak-up is applicable mainly against session flooding attacks and it is not applicable in case of request flooding attacks and asymmetric attacks. It is also assumed that server will somehow detect whether or not it is under attack.

A.2.2.b. DOW (Defense and Offense Wall) [113]: This mechanism employs the encouragement method which is presented in the *Speak-up* mechanism with an anomaly detection method based on K-means clustering to detect and filter session flooding attacks, request flooding attacks, and asymmetric attacks. Like *Speak-up*, DOW's encouragement model (currency model) encourages legitimate clients to increase their session rates so that they get a chance to be served. In other words, their detection model drops suspicious sessions while their currency model encourages more legitimate sessions. They believe that if these two models collaborate with each other, normal clients could gain higher service rates and lower delays in their response times.

[7]The reverse proxy is a type of proxy server that retrieves resources on behalf of a client from one or more servers

However, the main question that remains to be answered is with regards to the complexity and performance of this approach. This mechanism at this stage is too resource consuming to be implemented.

A.2.2.c. Differentiate DDoS flooding bots from human [114], [115]: Mechanisms of this category try to differentiate between the traffic from clients with the legitimate users (Human) and the malicious users (bots). For instance, *Kandula et al.* [114] propose a system to protect web clusters from application-level DDoS attacks by employing Completely Automated Public Turing test to tell Computers and Humans Apart (CAPTCHA [116]). Their proposed framework optimally divides the time spent in authenticating new clients and serving the authenticated ones. One of the main disadvantages of their approach or any similar approach is that requiring users to solve puzzles in order to authenticate themselves may become annoying for the users and introduce more delay for legitimate users. Furthermore, this policy disables web crawlers's access to the web sites; hence, search engines may not be able to index the content. In order to address the drawbacks of CAPTCHA-based mechanisms, *Oikonomou et al.* [115] propose three defenses against flash-crowd attacks which differentiate humans from bots based on the uniqueness of human behavior with regard to (1) request dynamics, (2) choice of content to access and (3) ability to ignore invisible content.

A.2.2.d. Admission control and congestion control [117]: In this mechanism, *Srivatsa et al.* propose admission control to limit the number of concurrent clients served by the online service. Admission control works based on port hiding that renders the online service invisible to unauthorised clients by hiding the port number on which the service accepts incoming requests. Then, they perform congestion control on admitted clients to allocate more resources to good clients by adaptively setting the client's priority level in response to the client's requests in a way that incorporates application-level semantics. However, this mechanism requires a challenge server, which can be the target of DDoS attacks.

A.2.2.e. TMH (Trust Management Helmet) [118]: This mechanism uses trust to differentiate legitimate users and attackers. The key idea is that servers should give priority to protecting the connectivity of good users during the application layer DDoS attacks instead of identifying all the attack requests. In doing so, each user is assigned a license, which is cryptographically secured against forgery or replay attacks, and a trust value, which is based on users' history. Detection of attackers is possible by considering these two features provided by each user.

A.2.2.f. Hybrid detection based on trust and information theory based metrics [119]: This mechanism proposes a hybrid detection scheme based on the trust information and information theory based metrics. This mechanism initially filters suspicious flows based on the trust value scored by the client. Then an entropy, which is the information based metric, is applied for final filtering of suspicious flows. Trust value for each client is assigned by the server based on the access pattern of the client and is updated every time the client communicates with the server. Each request from the client always includes the trust value to identify itself to the server. Entropy of requests per session is calculated based on the Web user browsing behaviour (HTTP request rate, page viewing time and sequence of the requested objects) of the client which is captured from the system log during non-attack cases. Entropy is then used for further rate limiting the flows. There is a scheduler in the architecture of this mechanism which schedules the sessions based on the trust value of the users and the system workload.

Discussion: Detection of and responding to the application-level DDoS flooding attacks at the servers or reverse proxies is not effective enough since attack traffic could have already affected the victims. As we pointed out in section *A.1.5*, hybrid defense mechanisms are the best way to combat DDoS flooding attacks since all of the defense nodes collaborate with each other to defeat coordinated DDoS flooding attacks. We enumerate some of the recent state-of-the-art hybrid defense mechanisms against application-level DDoS flooding attacks in this section and since recent attack incidents have proved that current mechanisms have not been fully successful, advanced defense mechanisms with novel features are yet to be deployed. Here, we briefly discuss some of those required features:

(i) Defense mechanisms must be capable of detecting the attacks independent of the attack's exact nature of operation since predicting and detecting all possible attacks by the attackers is hard.

(ii) Enhanced detection mechanisms should be in place to better distinguish between the legitimate and malicious requests. Using metrics such as the request rate, the packet headers, or the contents of the request may not be sufficient enough.

(iii) Response mechanisms should be more adaptive in the sense that legitimate users can claim their fair share of resources. In other words, a more request throttling mechanism which assign more server resources to the legitimate clients should be in place rather than the request blocking mechanisms.

B. ***Classification by the point in time*** *(i.e., between the start and end of a DDoS attack)* ***which defense takes place***

B.1. ***Before the attack (attack prevention):*** The best point in time to stop a DDoS attack is at its launching stage. In other words, attack prevention is the best DDoS defense solution. The prevention mechanisms can be deployed at the attack sources, intermediate networks, destinations or a combination of them. Most of the prevention mechanisms aim to fix security vulnerabilities (e.g., insecure protocols, weak authentication schemes, and vulnerable computer systems) that can be exploited to launch DDoS attacks. Several prevention mechanisms have been proposed in the literature [120]. There are some general prevention mechanisms that should be employed almost everywhere (e.g., servers, hosts, and intermediate networks) and in as many places as possible by both end hosts and service providers. Some of these general prevention mechanisms are as follows:

B.1.1. System & Protocol security mechanisms to increase the overall security of the systems: For instance, by preventing illegitimate accesses to the machines, removing bugs, updating

installed protocols, installing software patches, removing unused software, etc [2].

B.1.2. Fail-safe protection: Possible anticipations in case something goes wrong (e.g., replication of services and applications in diverse locations in case DDoS attack occurs successfully, business continuity and disaster management plans, etc.).

B.1.3. Resource allocation & accounting [2]: Providing resources to counter DDoS attacks and control users' access based on their privileges and behaviors [121]–[123].

B.1.4. Reconfiguration mechanisms: These mechanisms alter the topology of either the victim network to add more resources to tolerate the DDoS attack (e.g., resource replication services [124]) or the intermediate network to isolate the attack sources (e.g., attack isolation strategies) [2].

B.1.5. Installing firewalls and improved Intrusion Detection & Prevention Systems (IDPSs): All of the end hosts are encouraged to install IDPSs to prevent them from being compromised by the adversaries.

B.1.6. Employing *local filters* (e.g., Ingress/Egress [55], History-based IP filtering [73], hop-count filtering [74], Pi [76], route-based packet filtering [79], [80], etc.) and *globally coordinated filters* (e.g., ACC [89], Pushback [89], [90], AD and parallel-AD [91], TRACK [92], etc.) to block attack flows before their bombardment is another important category of the prevention mechanisms against DDoS attacks.

B.1.7. Load balancing [120] *and Flow control* are two other mechanisms to prevent DDoS attacks. The former improves both the performance and mitigation against DDoS attack, and the latter prevents servers from going down.

B.1.8. Server-side specific security considerations: One of the main problems regarding application-level flooding attacks is that there is a lack of security mechanisms or security policies in place to address the servers vulnerabilities against application-level DDoS flooding attacks. Such security mechanisms or policies can protect servers from various attacks. For instance, *Shekyan* in [43] suggest the following policies as the best protections for the servers in handling the write readiness for active sockets:

- Do not accept connections with abnormally small advertised window sizes.
- Do not enable persistent connections and HTTP pipelining unless performance really benefits from it.
- Limit the absolute connection lifetime to some reasonable value.

As another example, disabling open recursion[8] on name servers from external sources and only accepting recursive DNS queries originating from trusted sources has been proposed as an effective mechanism to diminish the amplification vector of DNS amplification attacks [125]. Similar security mechanisms or security policies for different servers, such as, Web servers, application servers, database servers, etc., should be defined and should be used by considering current vulnerabilities of the servers against various application-level DDoS flooding attacks.

B.1.9. Finally, service providers can have *strategies* in place to better identify their legitimate users. For instance, they can put dynamic pricing to network resource usages and charge their customers differently for the use of different resources [34]. Another effective service provider strategy was recently employed by Cisco in their IPS 7.0 code upgrade [54] . IPS 7.0 upgrade has *global correlation* feature that can be configured on every service provider IPS sensors so that they are aware of the network devices with a reputation for malicious activity, and can take action against them. This feature is useful when service providers' network is under attack from a botnet DDoS attack since sensors can drop all the traffic coming from bad reputation sources. Furthermore, this whole process is very inexpensive since it occurs before the signatures are used.

Prevention mechanisms aim to provide systems with increased security. However, these mechanisms can never completely remove the threat of DDoS attacks since they are always vulnerable to novel attacks for which signatures and patches are not available.

B.2. ***During the attack (attack detection):*** The next step in defending against DDoS attacks is *attack detection*, which happens during the attack. The detection mechanisms can also be deployed at sources, intermediate networks, destinations or a combination of them.

There are various mechanisms to detect DDoS attacks. Some of the detection mechanisms detect attack flows when the network links are congested to a certain level [77] [126]. Other mechanisms detect DDoS flooding attack traffic (*not vulnerability attacks*[9]) when anomalous patterns are discovered in both the network/transport-level traffic and application-level traffic (e.g., MIB *information analysis* [70], D-WARD [58], [59], MULTOPS [60], TOPS [61], [127]–[129], DDoS-Shield [3], [109], DAT [111]. There are many IDPSs that are based on these detection mechanisms. They employ data mining and artificial intelligence techniques for more accurate detection. These mechanisms monitor some features/headers of the traffic flows at various locations and points in time. Basically, they learn the normal behavior of either the network/transport-level or application-level traffic. Then, based on the information they have monitored and collected they can detect any changes on the traffic patterns and usage patterns of the resources. Based on the analysis in [130], anomaly detection algorithms to detect a DDoS flooding attacks can be classified depending on either the monitored parameters [131] or statistical techniques used (e.g., change point detection [127], wavelet analysis [128]) or granularity level of the analysis [129].

As we discussed earlier, the most practical place to detect DDoS flooding attack is at the victims' side since abnormal deviations cannot be easily found until the attack turns to its final stage. Even after the attack is detected, it is difficult for the victims to launch an efficient response mechanism because of the numerous malicious packets that have been aggregated at the victims' side. Therefore, defending against DDoS flooding attacks should be initiated at earlier points in time and as near as possible to the sources of the attacks. Detecting

[8]Name servers on the Internet that have recursion enabled provide recursive DNS responses to anyone (a.k.a. open resolvers).

[9]Vulnerability attacks are mostly detected by employing databases of known signatures.

(defending) at either intermediate networks or sources of the attacks have two main advantages: (1) the detection is more concealed since it is deployed in a separate location from attack target and (2) the detection mechanism is less vulnerable to DDoS attacks. However, accurate detection is not easy or it is even impossible to achieve since there is not enough evidence to detect attacks at these stages (e.g., source and upstream routers). Two fundamental challenges to detect DDoS flooding attacks in time and as near as possible to the attack sources are: (1) the lack of a wide deployment of DDoS defense mechanisms at different points of the Internet, and (2) the lack of collaboration and cooperation among distributed deployed defense mechanisms in order to increase the detection accuracy, decrease unnecessary redundant tasks (because of the lack of coordination), and, finally, to increase the performance efficiency of DDoS defense mechanisms. In case of application-level DDoS flooding attacks, all of the current detection mechanisms are deployed at the destination (servers) since it is not possible to perform detection at the layer 2/layer 3 intermediate networks. However, it will be possible to stop application-level DDoS flooding attacks at the intermediate networks if some layer 2/layer 3 extractable features of these attacks are found by careful analysis of these attacks and in-depth architecture.

B.3. ***After the attack (attack source identification and response)***: After a DDoS attack is detected, the defense system should identify the source of the attack and block the attack traffic. Today, most of the DDoS response mechanisms cannot completely prevent or stop DDoS attacks. Therefore, minimizing the attack impact and maximizing the availability of services is the main focus of all after the attack mechanisms. Moreover, law enforcement agencies must collaborate and cooperate with each other in order to gather and submit evidences that could be used to prosecute attackers. It is necessary for all the Internet providers to understand that even if a particular provider would be able to secure its own assets, it does not secure itself against DDoS attacks as other compromised hosts of other providers could still be used to launch attacks on it. Therefore, without collaborating with others to make sure their assets are also secured, defending against DDoS attack is almost impossible.

There are two main categories for most of the after the attack mechanisms:

B.3.1. Attack source identification: The first category of after the attack mechanisms is responsible to identify the source of the attack. For instance, an attacker uses host x to launch an attack by representing the spoofed source address of host y, IP traceback mechanism must find out the real source address of the attacker which is host x. This can be accomplished if there is a way of traversing all the routers from x to the victim in the reverse order or marking the legitimate paths or packets so that spoofed or illegitimate ones are identifiable. Towards this, traceback mechanisms [63]–[68] have been proposed in the literature.

B.3.2. Initiating a proper response: The second category of after the attack mechanisms is responsible for initiating a proper response to the attack. Most of the DDoS defense mechanisms apply throttling (rate limit) or packet filtering on upstream routers and hosts for the traffic coming from those *identified attack flows* (e.g., spoofed IP addresses) after identifying the source of the attack. For instance, history-based IP filtering [73], hop-count filtering [74], Pi [76], AD [91], TRACK [92], and StopIt [101], [105] employ packet filtering upon detection of DDoS attacks and ACC [89], Pushback [89], [90], PAD [91], AITF [103], and DEFCOM [93] employ throttling upon detection of DDoS attacks. Other mechanisms specially in the case of application-level DDoS flooding attacks employ some encouragement models in which servers ask the legitimate clients to increase their session rates to crowd out the malicious clients (e.g., Speak-up [112], and DOW [113]).

VI. DDoS Defense: Performance Measurement Metrics

Many mitigation and defense mechanisms to address DDoS attacks have already been proposed in the literature. However, there is no unique set of consistent metrics to evaluate these mechanisms. Obviously, monetary cost (CAPEX and OPEX) can be used to compare DDoS defense mechanisms, however, this does not evaluate the effectiveness of the schemes. In this section, we review and discuss some of the metrics and attributes found in the literature that can be used to comparatively evaluate DDoS mitigation techniques [132]. Then, in Table III and Table IV, we qualitatively compare the defense mechanisms against network/transport-level DDoS flooding attacks and the defense mechanisms against application-level DDoS flooding attacks based on their deployment location using some of these performance measurement metrics such as: defense strength (accuracy), scalability, delay, system performance degradation, implementation complexity, and whether these categories of defense mechanisms are considered as holistic defense mechanisms or not.

The performance measurement metrics are as follows:

1. Defense Strength: The strength of a defense mechanism can be measured by various metrics depending on how well it can prevent, detect, and stop the attacks. These metrics could be defined based on the decision or prediction that each defense mechanism makes. Defense mechanisms either detect and respond to the attacks or miss them. Based on their responses, there are four possible outcomes as shown in Table II.

In Table II, outcome A is called ***true negative*** (i.e., the desired outcome was negative and the outcome of the defense mechanism was negative as well), B is called ***false negative*** (i.e., the desired outcome was positive and the outcome of the defense mechanism was negative), C is called ***false positive*** (i.e., the desired outcome was negative and the outcome of the defense mechanism was positive), and D is called ***true positive*** (i.e., the desired outcome was positive and the outcome of the defense mechanism was also positive).

Based on the outcomes presented in Table II, six metrics, which have been previously introduced in the artificial intelligence literature [133], can be employed to evaluate DDoS defense mechanisms. These metrics are as follows:

1.a. Accuracy ((A+D)/(A+B+C+D)): Ratio of the correct outcomes of the defense mechanism (true positives and true negatives) over the total outcomes of the defense mechanism.

TABLE II
A MATRIX SHOWING THE OUTCOMES OF DDoS MITIGATION/DEFENSE MECHANISMS COMPARED TO DESIRABLE OUTCOMES

		Desirable decision of the DDoS defense	
		Negative	Positive
DDoS defense decision	Negative	A	B
	Positive	C	D

1.b. Sensitivity (D/(B+D)): Ratio of true positives over total desired positive outcomes.

1.c. Specificity (A/(A+C)): Ratio of true negatives over total desired negative outcomes.

1.d. Precision (D/(C+D)): Ratio of true positives over the total positive outcomes of the defense mechanism.

1.e. Reliability or False positive rate (C/(C+D)): Ratio of false positive outcomes of the defense mechanism over total positive outcomes of the defense mechanism.

1.f. False negative rate (B/(A+B)): Ratio of false negative outcomes of the defense mechanism over total negative outcomes of the defense mechanism.

2. Compromise-ability: Could an attacker exploit a defense mechanism in order to launch attacks (e.g., DDoS) against the whole system?

3. Delay in detection/response: How long does it take to detect/react to the attack?

4. System performance degradation: Does a defense mechanism cause any performance issues (e.g., memory shortage, lack of CPU cycles) or demand any additional requirements to perform perfectly?

5. Passive, reactive or proactive: Does a defense mechanism defend attacks by proactively preventing them from happening? Does it only react to the existing attacks? or does it take actions only after the DDoS attacks are launched?

6. Holistic defense: A holistic defense mechanism by considering all the required tasks in order to stop the DDoS attacks (i.e., both detection and response).

7. Implementation complexity: One of the important metrics to compare defense mechanisms is their implementation complexity. The best defense mechanisms in this classification are those that are easy and feasible to implement.

8. Usability: The interface that defense mechanisms provide to their users should be as user-friendly as possible.

9. Deployment location: As we mentioned earlier, deployment location is another metric to compare various defense mechanisms. Each location has its own benefits and disadvantages which makes one mechanism better than the other.

10. Scalability: A scalable defense mechanism can effectively handle its attack detection and response duties even if both the number of attackers and the amount of attack traffic increases.

VII. CYBER-INSURANCE & DDOS FLOODING ATTACK

Prevention, protection, and mitigation of cyber attacks solely by a combination of technical and operational/procedural means is not a complete cyber defense strategy. A complete cyber defense strategy must include cyber risk management that accepts and manages the information and network security risks [134]. Various organizations and businesses incorporate cyber risk management as part of their overall risk management strategy. According to a recent survey sponsored by *Zurich financial services* [135], which was conducted to get an insight on current state of enterprise-wide information security and cyber liability risk management, more than two-thirds of the respondents claimed that information and network security risks were a specific focus within their organizations. The inability of technical solutions to provide absolute information or network security, the defectiveness of traditional insurance policies to address these kind of risks (i.e., cyber-threats), and the underdevelopment clarification in the cyber-liability laws has led to the appearance of cyber-insurance products, as part of the cyber risk management, to cover losses and liabilities from the network or information security breaches [136]–[138]. In other words, cyber-insurance is an insurance product that can be purchased by organizations to protect them from the risks that cannot be mitigated [134] and to transfer the financial consequences of those risks to the cyber-insurance companies [139]. However, as found in [135], cyber-insurance is not still part of the cyber risk management of most businesses and the organizations.

In all of the traditional insurance policies (e.g., earthquake/fire protection) offered by insurance companies, there are some requirements that the property owner should meet (e.g., policies, standards) before obtaining the insurance [138]. As with other insurance policies, the cyber-insurance policy also requires the organizations to take some initial steps in order to obtain the cyber-insurance. These steps may include employing various information/network security standards, privacy policies, and information/network security assessment frameworks (e.g., Bell Labs security framework [140], ITU X.805 standard [141], ISO 27002 standard [142]) that most of the time requires significant investments by the IT organizations. However, most of the private individuals and organizations are reluctant towards these investments because they believe that their investments will not be entirely effective since most of their systems are somehow connected to the outside systems by either the Internet or other networked environments and those systems may be insecure; hence, they may put their own systems at risk [138].

Coverage of DDoS flooding attacks is a common feature of most current cyber-insurance policies. As we mentioned earlier, there are some requirements that the property owner (e.g., service provider) should meet before obtaining the cyber-insurance. For instance, service providers can enforce specific policies to insure the security of their customers' received services. For example, traffic segregation via VPNs and cleaning via a number of scrubbing centers in the network that filter bad traffic (e.g., viruses) and allow only clean traffic to the customers' site based on the customers' Service Level Agreement (SLA [10]) is currently one of the popular policy practices associated with obtaining cyber-insurance policies

[10]A formal written agreement made between service provider and its customer, defining the understanding of service requirements between them.

TABLE III
QUALITATIVE COMPARISON OF DEFENSE MECHANISMS AGAINST NETWORK/TRANSPORT-LEVEL DDOS FLOODING ATTACKS BASED ON THEIR DEPLOYMENT LOCATION

		Defense strength (Accuracy)	Scalability	System performance	Implementation complexity	Holistic defense
Centralized	***Source-based***	Low	Low	Moderate	Low	No
	Destination-based	High	Low	Good	Low	No
	Network-based	Low	Medium	Moderate	Medium	No
Distributed	***Hybrid (Distributed)***	Medium	Medium-high	Poor-moderate	Medium-high	Yes

TABLE IV
QUALITATIVE COMPARISON OF DEFENSE MECHANISMS AGAINST APPLICATION-LEVEL DDOS FLOODING ATTACKS BASED ON THEIR DEPLOYMENT LOCATION

	Defense strength (Accuracy)	Scalability	System performance	Implementation complexity	Holistic defense
Destination-based (server-side)	High	Low	Moderate-good	Low	No
Hybrid (Distributed)	Medium	Medium	Moderate	Medium-high	Yes

(e.g., IntruGuard [143], Neustar SiteProtect [144], and Cisco service provider infrastructure security techniques [145]).

VIII. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper, we have presented a comprehensive classification of various DDoS defense mechanisms along with their advantages and disadvantages based on where and when they detect and respond to DDoS flooding attacks. An ideal comprehensive DDoS defense mechanism must have specific features to combat DDoS flooding attacks both in real-time and as close as possible to the attack sources. These features are as follows:

1) *More nodes in the Internet should be involved in preventing, detecting, and responding to DDoS flooding attacks (i.e., Hybrid (Distributed) defense)*. As we discussed earlier, the detection accuracy is high at the victim side but it is not robust; victims cannot tolerate high volume of DDoS traffic. Stopping the attacks at the source could be the best response option but it is very difficult since the volume of the traffic at the sources is not significant to differentiate between legitimate and malicious traffic. Furthermore, the collateral damage is high at intermediate networks because there is not enough memory and CPU cycles to profile the traffic. Therefore, centralized mechanisms in which, all the defense components (i.e., prevention, detection, and response) are deployed at the same place, are not practical against DDoS flooding attacks.
2) *There should be collaboration and cooperation among the key defensive points within and between service providers in the Internet*. The main challenge in order to achieve this goal is that there should be some economic incentives among different service providers in order to achieve highly cooperative defense mechanisms.
3) *More reliable mechanisms are required to authenticate the sources of the Internet traffic so that malicious users could be identified and held accountable for their activities (i.e., Anti-spoofing mechanisms).*
4) *Trusted communication mechanisms for cooperation and collaboration among various distributed components are needed*. For instance, in the pushback mechanism, rate limit requests to the upstream routers could be sent by a malicious point in the network.

We strongly believe that combining source address authentication, capability mechanisms, and filtering mechanisms could be the most effective and efficient way to address the DDoS flooding attacks in a distributed cooperative/collaborative DDoS defense mechanism. More development and deployment of distributed defense mechanisms from researchers and service providers respectively is what we expect to see in the near future (short to medium term). In a longer term we expect to see:

- The inevitable cooperation and collaboration among service providers to detect and stop the DDoS flooding attacks closer to their sources. The rapid growth of collaborative environments such as Cloud Computing [146] and the Internet of Things (IoT) [147]–[149] leads to a large number of application developments both in and for such environments. This expands the threat landscape for DDoS flooding attacks and speeds up the transition to the era in which there is an inevitable cooperation and collaboration among various organizations and service providers for a stronger and faster defense against DDoS flooding attacks.
- The incorporation of the results of the attackers' incentives analysis into future defense strategies (i.e., this may lead to different strategies based on the attacker's motivations).
- The employment of the cross layer traffic analysis and defense mechanisms (i.e., looking at the information at multiple protocol layers simultaneously to detect and respond to the DDoS flooding attacks).
- The development of strict cyber-crime laws and multi-national enforcement mechanisms along with refined cyber-insurance policies that require implementation of DDoS detection and prevention mechanisms.

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers for their constructive comments and valuable suggestions. This research has been supported by Cisco systems' research award and the NSF award CCF-0720737.

REFERENCES

[1] P. J. Criscuolo, *Distributed Denial of Service*, Tribe Flood Network 2000, and Stacheldraht CIAC-2319, Department of Energy Computer Incident Advisory Capability (CIAC), UCRL-ID-136939, Rev. 1., Lawrence Livermore National Laboratory, February 14, 2000.

[2] J. Mirkovic and P. Reiher, *A taxonomy of DDoS attack and DDoS defense mechanisms*, ACM SIGCOMM Computer Communications Review, vol. 34, no. 2, pp. 39-53, April 2004.

[3] S. Ranjan, R. Swaminathan, M. Uysal, and E. Knightly, *DDoS-Resilient Scheduling to Counter Application Layer Attacks under Imperfect Detection*, IEEE INFOCOM'06, 2006.

[4] R. K. C. Chang, *Defending against flooding-based distributed denial of service attacks: A tutorial*, Computer J. IEEE Commun. Magazine, Vol. 40, no. 10, pp. 42-51, 2002.

[5] R. Puri, *Bots and Botnet – an overview*, Aug. 08, 2003, [online] http://www.giac.org/practical/GSEC/Ramneek_Puri_GSEC.eps

[6] B. Todd, *Distributed Denial of Service Attacks*, Feb. 18, 2000, [online] http://www.linuxsecurity.com/resource_files/intrusion_detection/ddos–whitepaper.html

[7] CERT, *Denial of Service Attacks*, June 4, 2001, [online] http://www.cert.org/tech_tips/denial_of_service.html

[8] J. Liu, Y. Xiao, K. Ghaboosi, H. Deng, and J. Zhang, *Botnet: Classification, Attacks, Detection, Tracing, and Preventive Measures*, EURASIP J. Wireless Communications and Networking, vol. 2009, Article ID 692654, 11 pages, 2009.

[9] *Yahoo on Trail of Site Hackers*, Wired.com, Feb. 8, 2000, [online] http://www.wired.com/news/business/0,1367,34221,00.html

[10] *Powerful Attack Cripples Internet*, Oct. 23, 2002, [online] http://www.greenspun.com/bboard/q–and–a–fetch–msg.tcl?msg_id=00A7G7

[11] *Mydoom lesson: Take proactive steps to prevent DDoS attacks*, Feb. 6, 2004, [online] http://www.computerworld.com/s/article/89932/Mydoom_lesson_Take_proactive_steps_to_prevent_DDoS_attacks?taxonomyId=017

[12] *Lazy Hacker and Little Worm Set Off Cyberwar Frenzy*, July 8, 2009, [online] http://www.wired.com/threatlevel/2009/07/mydoom/

[13] *New "cyber attacks" hit S Korea*, July 9, 2009, [online] http://news.bbc.co.uk/2/hi/asia-pacific/8142282.stm

[14] *Operation Payback cripples MasterCard site in revenge for WikiLeaks ban*, Dec. 8, 2010, [online] http://www.guardian.co.uk/media/2010/dec/08/operation–payback–mastercard–website–wikileaks

[15] T. Kitten, *DDoS: Lessons from Phase 2 Attacks*, Jan. 14, 2013, [online] http://www.bankinfosecurity.com/ddos-attacks-lessons-from-phase-2-a-5420/op-1

[16] Forrester Consulting, *The Trends And Changing Landscape Of DDoS Threats And Protection*, A commissioned study conducted by Forrester Consulting on behalf of VeriSign, Inc., July 2009.

[17] *Worldwide Infrastructure Security Report: Volume VI, 2011 Report*, Arbor Networks, Feb. 1st, 2011, [online] http://www.arbornetworks.com/report

[18] Prolexic Technologies, [online] http://www.prolexic.com/index.php_knowledge–center/frequently–asked–questions/index.html

[19] X. Geng, Y. Huang, and A. B. Whinston, *Defending wireless infrastructure against the challenge of DDoS attacks*, Mobile Networks and Applications, vol. 7, no. 3, pp. 213-223, 2002.

[20] A. D. Wood, and J. A. Stankovic, *A Taxonomy for Denial-of-Service Attacks in Wireless Sensor Networks*, Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems, CRC Press, 2004 (invited chapter).

[21] S. T. Zargar, M. B. H. Weiss, C. E. Caicedo, and J. B. D. Joshi, *Security in Dynamic Spectrum Access Systems: A Survey*, in Telecommunications Policy Research Conference, Arlington VA, 2009.

[22] N. Fultz, and J. Grossklags, *Blue versus Red: Towards a Model of Distributed Security Attacks*, In Financial Cryptography and Data Security, Roger Dingledine and Philippe Golle (Eds.). Lecture Notes in Computer Science, vol. 5628, Springer-Verlag, pp. 167-183, 2009, Berlin, Heidelberg.

[23] L. Greenemeier, *Estonian Attacks Raise Concern Over Cyber "Nuclear Winter"*, Information Week, May 24, 2007, [online] http://www.informationweek.com/news_showArticle.jhtml?articleID=199701774

[24] [online] http://isc.sans.edu/diary.html?storyid=6622

[25] [online] http://techcrunch.com/2010/11/28/wikileaks-ddos-attack/

[26] [online] http://www.sbsun.com/ci_21392063/us-general-we-hacked-enemy-afghanistan

[27] [online] http://www.gideonrasmussen.com/article-14.html

[28] [online] http://online.wsj.com/article/SB123914805204099085.html

[29] P. Liu, W. Zang, and M. Yu, *Incentive-based modeling and inference of attacker intent, objectives, and strategies*, ACM Trans. Inf. Syst. Secur. vol. 8, no. 1, pp. 78-118, February 2005.

[30] N. Chantler, *Profile of a Computer Hacker*, Interpact Press, FL, 1997.

[31] L. C. Chen, T. A. Longstaff, and K. M. Carley, *Characterization of defense mechanisms against distributed denial of service attacks*, Computers & Security, vol. 23, no. 8, pp. 665-678, December 2004.

[32] T. Peng, C. Leckie, and K. Ramamohanarao, *Survey of network-based defense mechanisms countering the DoS and DDoS problems*, ACM Comput. Surv. 39, 1, Article 3, April 2007.

[33] U. Tariq, M. Hong, and K. Lhee, *A Comprehensive Categorization of DDoS Attack and DDoS Defense Techniques*, ADMA LNAI 4093, pp. 1025-1036, 2006.

[34] S. M. Specht, and R. B. Lee, *Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures*, in Proc. 17th International Conference on Parallel and Distributed Computing Systems, pp.543-550, 2004.

[35] RioRey, Inc. 2009-2012, *RioRey Taxonomy of DDoS Attacks*, RioRey_Taxonomy_Rev_2.3_2012, 2012. [online] http://www.riorey.com/x-resources/2012/RioRey_Taxonomy_DDoS_Attacks_2012.eps

[36] C. Douligeris, and A. Mitrokotsa, *DDoS attacks and defense mechanisms: classification and state-of-the-art*, Computer Networks, Vol. 44, No. 5, pp. 643-666, April 2004.

[37] S. Ranjan, R. Swaminathan, M. Uysal, A. Nucci, and E. Knightly, *DDoS-Shield: DDoS-Resilient Scheduling to Counter Application Layer attacks*, IEEE/ACM Trans. Netw., Vol. 17, No. 1, pp. 2639, February 2009.

[38] *Arbor Application Brief: The Growing Threat of Application-Layer DDoS Attacks*, Arbor Networks, Feb. 28, 2011, [online] http://www.arbornetworks.com/component/docman/doc_download/467-the-growing-threat-of-application-layer-ddos-attacks?Itemid=442.

[39] BreakingPoint Labs, *Application-Layer DDoS Attacks Are Growing: Three to Watch Out For*, Oct. 4, 2011, [online] http://www.breakingpointsystems.com/resources/blog/application-layer-ddos-attacks-growing/

[40] ha.ckers.org, *Slowloris HTTP DoS*, Retrieved Oct. 19, 2012, [online] http://ha.ckers.org/slowloris/

[41] TrustWave SpiderLab, *(Updated) ModSecurity Advanced Topic of the Week: Mitigating Slow HTTP DoS Attacks*, Jul. 13, 2011, [online] http://blog.spiderlabs.com/2011/07/advanced-topic-of-the-week-mitigating-slow-http-dos-attacks.html

[42] K. J. Higgins, *Researchers To Demonstrate New Attack That Exploits HTTP*, Nov. 01, 2010, [online] http://www.darkreading.com/vulnerability-management/167901026/security/attacks-breaches/228000532/index.html

[43] S. Shekyan, *Are you ready for slow reading?*, Jan. 5, 2012, [online] https://community.qualys.com/blogs/securitylabs/2012/01/05/slow-read

[44] E. Alomari, S. Manickam, B. B. Gupta, S. Karuppayah, and R. Alfaris, *Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art*, International Journal of Computer Applications, Vol. 49, no. 7, pp. 24-32, Jul., 2012.

[45] J. Lo et al., *An IRC Tutorial*, April, 2003, irchelp.com 1997, [online] http://www.irchelp.org/irchelp/irctutorial.html#part1.

[46] B. Hancock, *Trinity v3, a DDoS tool, hits the streets*, Computers & Security, Vol. 19, no. 7, pp. 574-574, Nov., 2000.

[47] Bysin, *knight.c sourcecode*, 2001, [online] http://packetstormsecurity.org/distributed/ knight.c.

[48] team-cymru Inc., *A Taste of HTTP Botnets*, July, 2008, [online] http://www.team-cymru.com/ReadingRoom/Whitepapers/2008/http-botnets.eps

[49] J. Nazario, *BlackEnergy DDoS Bot Analysis*, Arbor Networks, 2007, [online] http://atlas-public.ec2.arbor.net/docs/BlackEnergy+DDoS+Bot+Analysis.eps

[50] Praetox Technologies *Low Orbit Ion Cannon*, 2010, [online] https://github.com/NewEraCracker/LOIC/

[51] E. Mills, *DOJ, FBI, entertainment industry sites attacked after piracy arrests*, 2012, [online] http://news.cnet.com/8301-27080_3-57362279-245/doj-fbi-entertainment-industry-sites-attacked-after-piracy-arrests.

[52] C. Wilson , *DDoS and Security Reports: The Arbor Networks Security Blog*, Arbor Networks, 2011, [online] http://ddos.arbornetworks.com/2012/02/ddos-tools/.

[53] [online] http://infosecisland.com/blogview/12395–DDoS–Attack–Utilizes–Self–Destructing–Botnet.html

[54] Cisco, *IPS 7.0 Global Correlation*, 2009, [online] http://www.cisco.com/en/US/docs/security/ips/7.0/configuration/guide/ime/ime_collaboration.html

[55] P. Ferguson, and D. Senie, *Network Ingress Filtering: Defeating Denial of Service Attacks that employ IP source address spoofing*, Internet RFC 2827, 2000.

[56] S. Kent, and R. Atkinson, *Security Architecture for the Internet Protocol*, IETF, RFC 2401, November 1998.

[57] S. Kent, and R. Atkinson, *IP Authentication Header*, IETF, RFC 2402, November 1998.

[58] J. Mirkovic, G. Prier, and P. Reiher, *Attacking DDoS at the Source*, in Proc. 10th IEEE International Conference on Network Protocols (ICNP '02), Washington DC, USA, 2002.

[59] J. Mirkovic, G. Prier, and P. Reihe, *Source-End DDoS Defense*, in Proc. 2nd IEEE International Symposium on Network Computing and Applications, April 2003.

[60] T. M. Gil, and M. Poleto, *MULTOPS: a data-structure for bandwidth attack detection*, in Proc. 10th Usenix Security Symposium, Washington, DC, pp. 2338, August 13–17, 2001.

[61] S. Abdelsayed, D. Glimsholt, C. Leckie, S. Ryan, and S. Shami, *An efficient filter for denial-of-service bandwidth attacks*, in Proc. 46th IEEE Global Telecommunications Conference (GLOBECOM'03), pp. 13531357, 2003.

[62] Mananet, *Reverse Firewall*, [online] http://www.cs3–inc.com/pubs/Reverse_FireWall.eps

[63] A. John, and T. Sivakumar, *DDoS: Survey of Traceback Methods*, International Journal of Recent Trends in Engineering ACEEE (Association of Computer Electronics & Electrical Engineers), vol. 1, no. 2, May 2009.

[64] R. Chen, J. M. Park, and R. Marchany, *RIM: Router interface marking for IP traceback*, in IEEE Global Telecommunications Conference (GLOBECOM'06), 2006.

[65] B. Al–Duwairi, and G. Manimaran, *Novel Hybrid Schemes Employing Packet Marking and Logging for IP Traceback*, IEEE Trans. Parallel Distrib. Syst., vol. 17, no. 5, pp. 403- 418, May 2006.

[66] S. Savage, D. Wetherall, A. Karlin, and T. Anderson, *Practical Network Support for IP Traceback*, Technical report, Department of Computer Science and Engineering, University of Washington, 2000.

[67] H. Burch, and B. Cheswick, *Tracing anonymous packets to their approximate source*, in Proc. USENIX Large Installation Systems Administration Conference, pages 319–327, New Orleans, USA, Decemeber 2000.

[68] J. Glave, *Smurfing cripples ISPs*, in Wired TechnologyNews, 1998, [online] http://www.wired.com/news /news/technology/story/9506.html

[69] Y. C. Wu, H. R. Tseng, W. Yang, and R. H. Jan, *DDoS detection and traceback with decision tree and grey relational analysis*, Int. J. Ad Hoc Ubiquitous Comput., vol. 7, no. 2, pp. 121-136, 2011.

[70] B. Joao, D. Cabrera, and et al., *Proactive Detection of Distributed Denial of Service Attacks Using MIB Traffic Variables — A Feasibility Study*, Integrated Network Management Proceedings, pp. 609-622, 2001.

[71] R. Jalili, F. ImaniMehr, *Detection of Distributed Denial of Service Attacks Using Statistical Pre-Prossesor and Unsupervised Neural Network*, ISPEC, Springer-Verlag Berlin Heidelberg, pp.192-203, 2005.

[72] M. Li, J. Liu, and D. Long, *Probability Principle of Reliable Approach to detect signs of DDOS Flood Attacks*, PDCAT, Springer-Verlag Berlin Heidelberg, pp.596-599, 2004.

[73] T. Peng, C. Leckie, and K. Ramamohanarao, *Protection from distributed denial of service attacks using history-based IP filtering*, ICC '03. May , Vol.1, pp: 482- 486, 2003.

[74] H. Wang, C. Jin, and K. G. Shin, *Defense Against Spoofed IP Traffic Using Hop-Count Filtering*, IEEE/ACM Trans. Netw., vol. 15, no. 1, pp.40-53, February 2007.

[75] M. Abliz, *Internet Denial of Service Attacks and Defense Mechanisms*, University of Pittsburgh, Department of Computer Science, Technical Report. TR-11-178, March 2011.

[76] A. Yaar, A. Perrig, and D. Song, *Pi: A Path Identification Mechanism to Defend against DDoS Attacks*, in IEEE Symposium on Security and Privacy, pp. 93, 2003.

[77] Y. Kim, W. C. Lau, M. C. Chuah, and H. J. Chao, *PacketScore: A Statistics-Based Packet Filtering Scheme against Distributed Denial-of-Service Attacks*, IEEE Trans. Dependable Secure Computing, vol. 3, no. 2, pp. 141-155, 2006.

[78] E. Y. K. Chan et al., *Intrusion Detection Routers: Design, Implementation and Evaluation Using an Experimental Testbed*, IEEE J. Sel. Areas Commun., vol. 24, no. 10, pp. 1889 - 1900, 2006.

[79] K. Park, and H. Lee, *On the effectiveness of probabilistic packet marking for IP traceback under denial of service attack*, in Proc. IEEE INFOCOM 2001, pp. 338347.

[80] K. Park, and H. Lee, *On the Effectiveness of Route-Based Packet Filtering for Distributed DoS Attack Prevention in Power-Law Internets*, in Proc. ACM SIGCOMM, August 2001.

[81] A. T. Mizrak, S. Savage, and K. Marzullo, *Detecting compromised routers via packet forwarding behavior*, IEEE Network, pp.34-39, 2008.

[82] K. A. Bradley, S. Cheung, N. Puketza, B. Mukherjee, and R. A. Olsson, *Detecting Disruptive Routers: A Distributed Network Monitoring Approach*, in Proc. 1998 IEEE Symposium on Security and Privacy, May 1998.

[83] J. R. Hughes, T. Aura, and M. Bishop, *Using Conservation of Flow as a Security Mechanism in Network Protocols*, in Proc. 2000 IEEE Symposium on Security and Privacy, May 2000.

[84] J. M. Gonzalez, M. Anwar, and J. B. D. Joshi, *A trust-based approach against IP-spoofing attacks*, in Proc. IEEE PST, pp. 63-70, 2011.

[85] P. Zhou, X. Luo, A. Chen, and R. K. C. Chang, *STor: Social Network based Anonymous Communication in Tor*, in The Computing Research Repository (CoRR), 2011.

[86] S. T. Zargar, and J. B. D. Joshi, *A Collaborative Approach to Facilitate Intrusion Detection and Response against DDoS Attacks*, the 6th Int'l Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2010), Chicago, IL, October 9-12, 2010.

[87] M. R. Sharma, and J. W. Byers, *Scalable Coordination Techniques for Distributed Network Monitoring*, in Proc. PAM, pp. 349-352, 2005.

[88] B. Claise, *Cisco Systems NetFlow Services Export Version 9*, RFC 3954, 2004.

[89] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker, *Controlling high bandwidth aggregates in the network*, presented at Computer Communication Review, pp.62-73, 2002.

[90] D. Yau, J. C. S. Lui, and F. Liang, *Defending against distributed denial of service attacks using max-min fair server centric router throttles*, IEEE international conference on Quality of Service. 2002.

[91] R. Chen, and J. M. Park, *Attack Diagnosis: Throttling distributed denial-of-service attacks close to the attack sources*, IEEE Int'l Conference on Computer Communications and Networks (ICCCN'05), Oct. 2005.

[92] R. Chen, J. M. Park, and R. Marchany, *TRACK: A novel approach for defending against distributed denial-of-service attacks*, Technical Report TR-ECE-06-02, Dept. of Electrical and Computer Engineering, Virginia Tech, Feb. 2006.

[93] J. Mirkovic, P. Reiher, and M. Robinson, *Forming Alliance for DDoS Defense*, in Proc. New Security Paradigms Workshop, Centro Stefano Francini, Ascona, Switzerland, 2003.

[94] C. Papadopoulos, R. Lindell, J. Mehringer, A. Hussain, and R. Govindan, *Cossack: Coordinated Suppression of Simultaneous Attacks*, in Proc. DARPA Information Survivability Conference and Exposition, Vol. 1, pp. 2 13, Apr. 2003.

[95] T. Anderson, T. Roscoe, and D. Wetherall, *Preventing Internet denial-of-service with capabilities*, SIGCOMM Comput. Commun. Rev., vol. 34, no. 1, pp. 39-44, 2004.

[96] B. Parno et al., *Portcullis: protecting connection setup from denial-of-capability attacks*, SIGCOMM Comput. Commun. Rev., vol. 37, no. 4, pp. 289-300, 2007.

[97] X. Yang, D. Wetherall, and T. Anderson, *TVA: a DoS-limiting network architecture*, IEEE/ACM Trans. Netw., vol. 16, no. 6, pp. 1267-1280, 2008.

[98] X. Yang, D. Wetherall, and T. Anderson, *A DoS-limiting Architecture*, in ACM SIGCOMM, Philadelphia, PA, USA, August 2005.

[99] A. Yaar, A. Perrig, and D. Song, *SIFF: a Stateless Internet Flow Filter to Mitigate DDoS Flooding Attacks*, in Proc. 2004 IEEE Symposium on Security and Privacy, pp. 130-143, May 2004.

[100] X. Liu, A. Li, X. Yang, and D. Wetherall, *Passport: secure and adoptable source authentication*, in Proc. 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI'08), San Francisco, CA, USA, pp. 365-378, 2008.

[101] X. Liu, X. Yang, and Y. Lu, *To filter or to authorize: network-layer DoS defense against multimillion-node botnets*, in Proc. ACM SIGCOMM

conference on Data communication (SIGCOMM '08), NY, USA, pp. 195-206, 2008.

[102] X. Yang, *A DoS Limiting Network Architecture*, [online] http://www.cs.duke.edu/nds/ddos/

[103] K. Argyraki, and D. R. Cheriton, *Scalable network-layer defense against internet bandwidth-flooding attacks*, in IEEE/ACM Trans. Netw., 17(4), pp. 1284-1297, August 2009.

[104] F. Huici, *Deployable Filtering Architectures Against Large Denial-of-Service Attacks*, Ph.D. dissertation, Department of Computer Science, University College London, December, 2009.

[105] X. Liu, X. Yang, and Y. Lu, *StopIt: Mitigating DoS Flooding Attacks from Multi-Million Botnets*, Technical Report 08-05, http://www.cs.duke.edu/ xinl/stopit-tr.eps

[106] G. Kambourakis, T. Moschos, D. Geneiatakis, and S. Gritzalis, *Detecting DNS Amplification Attacks*, in Critical Information Infrastructures Security Lecture Notes in Computer Science, Vol. 5141, pp. 185-196, 2008.

[107] IPTraf tool, *An IP Network Monitor*, [online] http://iptraf.seul.org/

[108] A. Rahul, S. K. Prashanth, B. S. kumarand , and G. Arun, *Detection of Intruders and Flooding In Voip Using IDS, Jacobson Fast And Hellinger Distance Algorithms*, IOSR Journal of Computer Engineering (IOSRJCE), Vol. 2, no. 2, pp. 30-36, July-Aug. 2012.

[109] S. Ranjan, R. Swaminathan, M. Uysal, A. Nucci, and E. Knightly, *DDoS-shield: DDoS-resilient scheduling to counter application layer attacks*, IEEE/ACM Trans. Netw., Vol. 17, no. 1, pp. 26-39, February 2009.

[110] Y. Xie, and S. Z. Yu, *A large-scale hidden semi-Markov model for anomaly detection on user browsing behaviors*, IEEE/ACM Trans. Netw. (TON), Vol. 17, no. 1, pp. 54-65, February 2009.

[111] H. I. Liu, and K. C. Chang, *Defending systems Against Tilt DDoS attacks*, Telecommunication Systems, Services, and Applications (TSSA), pp. 22-27, October 20-21, 2011.

[112] M. Walfish, M. Vutukuru, H. Balakrishnan, D. Karger, and S. Shenker, *DDoS defense by offense*, SIGCOMM Computer Communications Review, Vol. 36, no. 4, pp. 303-314, August 2006.

[113] J. Yu, Z. Li, H. Chen, and X. Chen, *A Detection and Offense Mechanism to Defend Against Application Layer DDoS Attacks*, the third International Conference on Networking and Services (ICNS'07), pp. 54, June 19-25, 2007.

[114] S. Kandula, D. Katabi, M. Jacob, and A. W. Berger*Botz-4-sale: Surviving organized ddos attacks that mimic flash crowds*, in Proc. Symposium on Networked Systems Design and Implementation (NSDI), Boston, May 2005.

[115] G. Oikonomou, and J. Mirkovic, *Modeling human behavior for defense against flash-crowd attacks*, in Proc. 2009 IEEE international conference on Communications (ICC'09), pp. 625-630, 2009.

[116] L. V. Ahn, M. Blum, N. J. Hopper, and J. Langford, *CAPTCHA: using hard AI problems for security*, in Proc. 22nd international conference on Theory and applications of cryptographic techniques (EUROCRYPT'03), Eli Biham (Ed.). Springer-Verlag, Berlin, Heidelberg, 294-311. 2003.

[117] M. Srivatsa, A. Iyengar, J. Yin, and L. Liu, *Mitigating application-level denial of service attacks on Web servers: A client-transparent approach*, ACM Trans. Web (TWEB), Vol. 2, no. 3, July 2008.

[118] J. Yu, C. Fang, L. Lu, and Z. Li, *A Lightweight Mechanism to Mitigate Application Layer DDoS Attacks*, in Proc. Infoscale 2009, LNICST 18, pp. 175191, 2009.

[119] S. R. Devi, and P. Yogesh, *A hybrid approach to counter application layer DDoS attacks*, International J. Cryptography and Information Security(IJCIS), Vol. 2, no.2, June 2012.

[120] X. Geng, and A. B. Whinston, *Defeating Distributed Denial of Service attacks*, IEEE IT Professional, 2(4), pp. 36-42, 2002.

[121] *Strategies to Protect Against Distributed Denial of Service (DDoS) Attacks*, Retrieved Oct. 19, 2012, [online] http://www.cisco.com/en/US/tech/tk59/technologies/_white_paper09186a0080174a5b.shtml

[122] A. D. Keromytis, V. Misra, and D. Rubenstein, *SOS: Secure Overlay Services*, in Proc. SIGCOMM'02, 2002.

[123] D. G. Andersen, V. Misra, and D. Rubenstein, *Mayday: Distributed filtering for internet services*, in Proc. USENIX'03, 2003.

[124] J. Yan, S. Early, and R. Anderson, *The XenoService - A Distributed Defeat for Distributed Denial of Service*, in Proc. ISW 2000, October 2000.

[125] ICANN Report, *DNS Distributed Denial of Service (DDoS) Attacks*, Security and Stability Advisory Committee (SSAC), March 2006.

[126] Y. Huang, and J. M. Pullen, *Countering Denial of Service attacks using congestion triggered packet sampling and filtering*, in Proc. 10th International Conference on Computer Communiations and Networks, 2001.

[127] T. Peng, C. Leckie, and K. Ramamohanarao, *Detecting distributed denial of service attacks using source ip address monitoring*, 2003, [Online] http://www.cs.mu.oz.au/tpeng/mudguard/research/detection.eps

[128] A. Dainotti, A. Pescape, and G. Ventre, *Wavelet-based detection of dos attacks*, in IEEE Global Telecommunications Conference, GLOBECOM, 2006.

[129] M. Kim, H. Kang, S. Hong, S. Chung, and J. W. Hong, *A flow-based method for abnormal network traffic detection*, in Network Operations and Management Symposium, vol. 1, pp. 599-612, April 2004.

[130] R. M. Mutebi, and I. A. Rai, *An Integrated Victim-based Approach against IP Packet Flooding Denial of Service*, International J. Computing and ICT Research, Special Issue Vol. 4, No. 1, pp. 70-80, October 2010.

[131] V. A. Siris, and F. Papaglou, *Application of anomaly detection algorithms for detecting syn flooding attacks*, in Proc. IEEE GLOBECOM, 2004.

[132] J. Mls, *Mitigating denial of service attacks: A tutorial*, J. Computer Security, Vol. 13, No. 6, pp. 807-837, 2005.

[133] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy". Remote Sensing of Environment, Vol. 62, No. 1, pp. 77–89, 1997.

[134] R. P. Majuca, W. Yurcik, and J. P. Kesan, *The evolution of cyberinsurance*, Information Systems Frontier, 2005.

[135] *A New Era In Information Security and Cyber Liability Risk Management: A Survey on Enterprise-wide Cyber Risk Management Practices*, Sponsored by Zurich Financial Services Group and administered by New York-based Advisen Ltd, October 2011, [online] http://corner.advisen.com/pdf_files/cyberliability_riskmanagement.eps

[136] J. Kesan, R. Majuca, and W. Yurcik, *Cyberinsurance as a market-based solution to the problem of cybersecurity:a case study*, SIFT Information Security Services, 2006.

[137] C. Lai, G. Medvinsky, and C. Neuman, *Endorsments, licensing, and insurance for distributed systems services*, in Proc. 2nd ACM Conf. Computer and Comm. Security (CCS), Fairfax, VA, November 1994.

[138] W. S. Baer, and A. Parkinson, *Cyberinsurance in IT Security Management*, IEEE Security & Privacy, Vol. 5, No. 3, pp. 50-56, May-June 2007.

[139] T. Bosco, *The Economic Viability of Cyber Insurance: Seeking Financial Certainty in IT Security*, in Proc. WEIS'05, Harvard, MA, June 2005.

[140] V. J. Gurbani, and A. R. McGee, *An early application of the Bell Labs Security framework to analyze vulnerabilities in the Internet telephony domain*, Bell Labs Technical Journal - Information Technology/Network Security, Vol. 12, No. 3, pp. 7-19, September 2007.

[141] *Security architecture for systems providing end-to-end communications*, ITU X.805 standard, International Telecommunication Union, 2003, [online] http://www.itu.int/rec/T-REC-X.805-200310-I/en

[142] *Information technology – Security techniques – Code of practice for information security management*, ISO 27002 standard, International Organization for Standardization, [online] http://www.iso.org/iso/catalogue_detail?csnumber=50297

[143] *IntruGuard: DDoS Mitigation Solution*, Juniper networks and IntruGuard, November 2007, [online] http://www.intruguard.com/documents/351267.eps

[144] *DDoS Survey: Q1 2012 When Businesses Go Dark*, Neustar Insights: DDoS Survey Q1 2012, [online] http://hello.neustar.biz/rs/neustarinc/images/neustar-insights-ddos-attack-survey-q1-2012.eps

[145] *Service Provider Infrastructure Security Techniques*, Service Provider Security, Cisco systems, [online] http://www.cisco.com/web/about/security/intelligence/sp_infrastruct_scty.html

[146] S. T. Zargar, H. Takabi, and J. B. D. Joshi, *DCDIDP: A Distributed, Collaborative, and Data-driven Intrusion Detection and Prevention Framework for Cloud Computing Environments*, the 7th Intl Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2011), October 15-18, 2011, Orlando, FL.

[147] J. Carter, *The Internet of Things: how it'll revolutionise your devices*, techradar, July 2012, [online] http://www.techradar.com/news/internet/the-internet-of-things-how-itll-revolutionise-your-devices-958669

[148] D. Evans, *The Internet of Things [INFOGRAPHIC]*, Cisco Blog, July 2011, [online] http://blogs.cisco.com/news/the-internet-of-things-infographic/

[149] M. Chui, M. Lffler, and R. Roberts, *The Internet of Things*, McKinsey Quarterly, March 2010, [online] http://www.mckinseyquarterly.com/The_Internet_of_Things_2538

Saman Taghavi Zargar is a PhD candidate in the Telecommunications and Networking Program and a member of the Laboratory of Education and Research on Security Assured Information Systems (LERSAIS) in the School of Information Sciences at the University of Pittsburgh. His research interests include network security; intrusion prevention, detection and response; security, privacy, and trust issues in the cloud computing environments; security, privacy, and trust issues in Dynamic Spectrum Access (DSA); distributed, mobile, and pervasive/ubiquitous computing. Zargar has an MS in computer engineering with the concentration in software engineering from Ferdowsi University of Mashhad, Iran. He is also a graduate student member of IEEE and the ACM.

James Joshi is an associate professor in the School of Information Sciences at the University of Pittsburgh. He is a founder and the director of the Laboratory of Education and Research on Security Assured Information Systems (LERSAIS). He received his MS in Computer Science and PhD in Computer Engineering degrees from Purdue University in 1998 and 2003, respectively. His research interests include Access Control Models, Security and Privacy of Distributed Systems, Trust Management and Information Survivability. He is a recipient of the NSF-CAREER award in 2006. He directs the NSF CyberCorp Scholarship for Service program at the University of Pittsburgh.

David Tipper is the Director of the Telecommunications and Networking Program and a Faculty member at the University of Pittsburgh, Pittsburgh, PA. He is a graduate of the University of Arizona (Ph.D. EE, MS SIE) and Virginia Tech (BS EE). His current research interests are survivable networks, performance analysis techniques, wireless/wired network design and information assurance techniques. Professor Tipper's research has been supported by grants from various government and corporate sources such as NSF, DARPA, NIST, IBM, ARO and AT&T. Professional activities include serving as the General Chair of the 7th Design of Reliable Communication Networks Workshop (DRCN2009), co-guest editor of two special issues of the Journal of Network and Systems Management one on Fault Management in Communication Networks which appeared in June, 1997 and one on Designing and Managing Optical Networks and Service Reliability, which appeared March, 2005 and co-guest editor of a special issue of the journal Telecommunication Systems on Reliable Networks Design and Modeling which will appear in 2013. He is the co-author of the textbook The Physical Layer of Communication Systems, which was published by Artech House in 2006. Also, he is the co-editor and a contributor to Information Assurance: Dependability and Security in Networked Systems, which was published by Morgan Kaufmann in 2008.