Special Issue Article

International Journal of
Distributed
Sensor Networks

International Journal of Distributed Sensor Networks
2017, Vol. 13(12)
© The Author(s) 2017
DOI: 10.1177/1550147717741463
journals.sagepub.com/home/dsn
SAGE

A survey of distributed denial-of-service attack, prevention, and mitigation techniques

Tasnuva Mahjabin[1], Yang Xiao[1], Guang Sun[2] and Wangdong Jiang[2]

Abstract

Distributed denial-of-service is one kind of the most highlighted and most important attacks of today's cyberworld. With simple but extremely powerful attack mechanisms, it introduces an immense threat to current Internet community. In this article, we present a comprehensive survey of distributed denial-of-service attack, prevention, and mitigation techniques. We provide a systematic analysis of this type of attacks including motivations and evolution, analysis of different attacks so far, protection techniques and mitigation techniques, and possible limitations and challenges of existing research. Finally, some important research directions are outlined which require more attentions in near future to ensure successful defense against distributed denial-of-service attacks.

Keywords

Denial-of-service, distributed denial-of-service, Internet of Things, Internet of Things botnet, distributed denial-of-service attack defense, distributed denial-of-service prevention, distributed denial-of-service mitigation

Date received: 26 July 2017; accepted: 27 September 2017

Handling Editor: Shancang Li

Introduction

On 21 October 2016, a stream of distributed denial of service (DDoS) attacks involving tens of millions of Internet Protocol (IP) addresses had been noted and attacked dyn domain name system (DNS).[1] The magnitude of the attack was claimed to be 1.2 Tbps and it has involved Internet of Things (IoT) devices.[1] This significant incident of DDoS attacks has proven the immense danger inherent with DDoS attacks and has taken the attention of today's cyberworld. This attack has opened up an essential discussion about cyber security and its unpredictability. According to the 12th annual report of Arbor Network published in Waterman,[2] the size and growth of the DDoS attacks were the largest in the last year and it has also increased in its frequency over the past few years. In Figure 1, the attack volume sizes of the different DDoS attacks during the past 10 years are presented and the figure shows a tremendous growth in 2016 in terms of the volume size. Therefore, there is a strong need to provide an up-to-date and state-of-the-art survey of DDoS attacks, prevention techniques, and migration techniques, and this is the motivation behind this article.

The design of the Internet provides best-effort, packet switched services to the users.[3,4] The result of this is to share resources among different users. As a

[1]Department of Computer Science, The University of Alabama, Tuscaloosa, AL, USA
[2]Hunan University of Finance and Economics, Changsha, China

Corresponding author:
Guang Sun, Hunan University of Finance and Economics, No. 139, Fenglin 2nd Road, Changsha 410205, China.
Email: simon5115@163.com

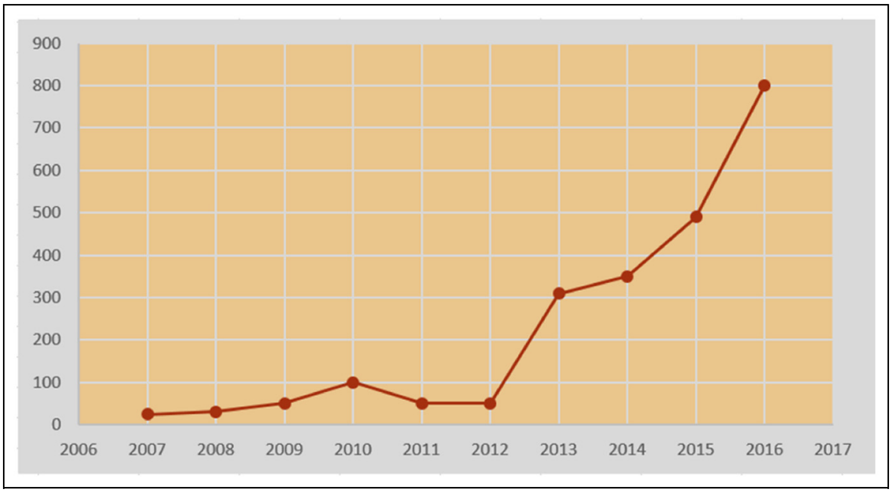

Figure 1. The volume sizes of DDoS attacks in gigabits per second, 2007–2016.[2]

consequence, the behaviors of one user may create disturbance to the services of the other users. One aim of DDoS attacks is to introduce this disturbance to the targeted users usually known as "victims." In general, a DDoS attack aims to hindering the access of legitimate users to a target system or services by overwhelming the resources.[5] For example, victims' networks or processing capacities are consumed by a huge stream of packets by attackers' exploiting Internet applications or network layer services and protocols. Consequently, the victims' networks or processing capabilities could not serve in a normal way and deny services to the victims. Furthermore, a DDoS victim may suffer a total or partial loss of its services and files if a careful mitigation method is not applied.[6] So far, the main threat of this attack comes from its distributed nature. Back in 1999, the first DDoS attack was reported by the computer incident advisory capability (CIAC)[7] and since then most of denial of service (DoS) attacks are found distributed in nature.

In general, DDoS attack packets do not show any obvious characteristics which can separate a malicious stream from a legitimate one. Also, the tools used in these attacks are easily accessible by the attackers and this increases the frequency and threats of the attacks. The simple structure of this kind of attacks follows many-to-one feature. Thus, if an attack is initiated; its complexity and impact become proportionally high. It takes the advantages of the less secure architecture of the Internet structure since the design of the Internet follows simple architecture which reduces the complexity of the core networks. Thus, the core routers have some limitations which help the DDoS attacks. The core routers cannot provide authentication services to the delivered IP packets. This limitation introduces a deep trouble known as IP spoofing which is one of the key powers of the DDoS attacks.[4] In IP spoofing, the attackers provide false information such as fake source IP addresses in the IP packets.[8] Furthermore, the routers also cannot provide packet tracing mechanisms because of the enormous traffic handled by them. These limitations provide the attackers the opportunity to remain hidden while performing DDoS attacks. The design of the Internet also ensures high-capacity traffic handling support to the core networks while low-capacity traffic handling support to the edge networks. This design ensures maximum utilization of the links with minimum costs. However, this also introduces the possibility to overwhelm an edge network if multiple sources talk to a single destination. This is what happens during a DDoS attack.[9] In the decentralized nature of the Internet management, all the networks are managed locally rather than with a central management authority or hierarchy.[10] This decentralized structure helped in the growth of the Internet. However, this mechanism also helps attackers as the implementation of a robust defense mechanism becomes extremely difficult for the decentralized structure. Without a central control, it is not possible to deploy a distributed solution to solve the problem of the DDoS attacks. Because DDoS attacks are totally distributed attacks, a single point solution does not improve the protection against this type of attacks. Thus, it is very obvious that even a highly secure system does not ensure protection from it. Also, the resource of a victim is always limited as compared to the resource of attackers as they can use a distributed attack scenario. All of these increase the complexity in detection, prevention, and mitigation of DDoS attacks.[11,12]

In this article, we provide an up-to-date and state-of-the-art survey of DDoS attacks, prevention techniques, and migration techniques. We present a systematic analysis of DDoS attacks which covers a taxonomy of DDoS attack types and their prevention and mitigation techniques. The contributions of this article include the following:

- We provide illustration of DDoS attack strategies which cover all of the phases involved in DDoS attacks.
- We present defense mechanisms against DDoS attacks which include important prevention and mitigation techniques.
- We include recent attack types as well as recent research on DDoS defense, presenting the current state of the art of DDoS research.
- We also enlist some challenges of the current research and future research directions.

Following this introduction, the rest of the article is organized as follows. Section "Attack targets and motivations" introduces targets and motivations of DDoS attacks. Section "Attack strategies" presents attack strategies used in DDoS attacks. Different types of DDoS attacks based on different attack mechanisms are presented in section "Attack mechanisms." Section "Prevention against DDoS attacks" covers DDoS prevention techniques, following DDoS mitigation techniques in section "DDoS mitigation." In section "DDoS attacks to other systems," we provide some discussions about DDoS attacks on non-traditional systems such as clouds, smart grids, smart homes, cyber-physical systems (CPSs), and IoT systems. We provide our discussions in section "Discussion" and conclude the article in section "Conclusion."

Attack targets and motivations

According to arbor network, every day above 1000 sizable different DDoS attacks are tracked by them around the world.[13] The targets of these DDoS attacks range from a very own home user to a government. In some attacks, a victim can be an e-commerce site, a bank, a commercial organization, or even an Internet service provider (ISP). One major motivation to attack these users is for some financial gains. However, an attractive target for a DDoS attack can be pornography or online gambling sites. Moreover, political organizations and governments are also major targets of DDoS attacks. Gaming sites or stock exchanges can also be targets of DDoS attacks, as shown in Figure 2.[14] This figure is published in a quarterly report from Kaspersky Lab and here we observe that mostly the e-commerce sites were the major targets of DDoS attacks in the second quarter of 2011.

Thus, the reasons or motivations behind DDoS attacks may vary. However, five different categories can be identified to characterize the motivations behind DDoS attacks:[15]

- *Financial or economic benefit*. The attacks that fall under this motivation are considered as the most dangerous attacks as they try to achieve some financial benefits from the attacks. The attackers in such a case are the highly experienced technicians. Thus, this type of attacks is hard to stop in the present scenario.
- *Revenge*. This is another motivation for DDoS attacks where some frustrated (possibly technically lower skilled) individuals perform the attacks as a repayment of some perceived oppressions.
- *Ideological belief*. Some attackers become motivated to attack a target because of their ideological belief. This has become an influential reason behind DDoS attacks. Although they are not large in frequency as compared to other motivations, their impacts and sizes are as large as seen in the recent years. The Estonia attack in 2007, the China and CNN attack in 2008,[16] the Iran attack in 2009,[17] and WikiLeaks in 2010[18] are some of the mostly highlighted DDoS attacks of the past 10 years where the motivations of the attacks are either ideological or political belief.
- *Intellectual challenge*. The attackers of this group are mainly motivated to conduct DDoS attacks to show off their capabilities and power. The availability of the easy-to-use attack tools and botnets motivates these attackers to conduct experiments of DDoS attacks.
- *Cyberwarfare*. This is another important attack motivation which incurs danger and significant economic impacts on its targets. Generally, some well-trained people of a military or terrorist organization conduct an attack of this type. Here, the attackers belong to some countries and perform their attacks on some other country's organizations. A significant amount of resource and time are used to do such attacks and this may paralyze a country's cyberworld and critical infrastructure through service disruptions.

Attack strategies

The basic structure of a DDoS attack is presented in Figure 3.[19] It comprises three different phases and four different components.[20] The components are known as an attacker, multiple control masters or handlers, multiple slaves, agents, or zombies, and a victim or target machine.

In the first phase, the attacker spends a lot of its time to create a significant amount of compromised machines which are called the masters or handlers as they appoint and control other machines in the attack army. The creation of the master army is usually an automated process where a continuous scanning is

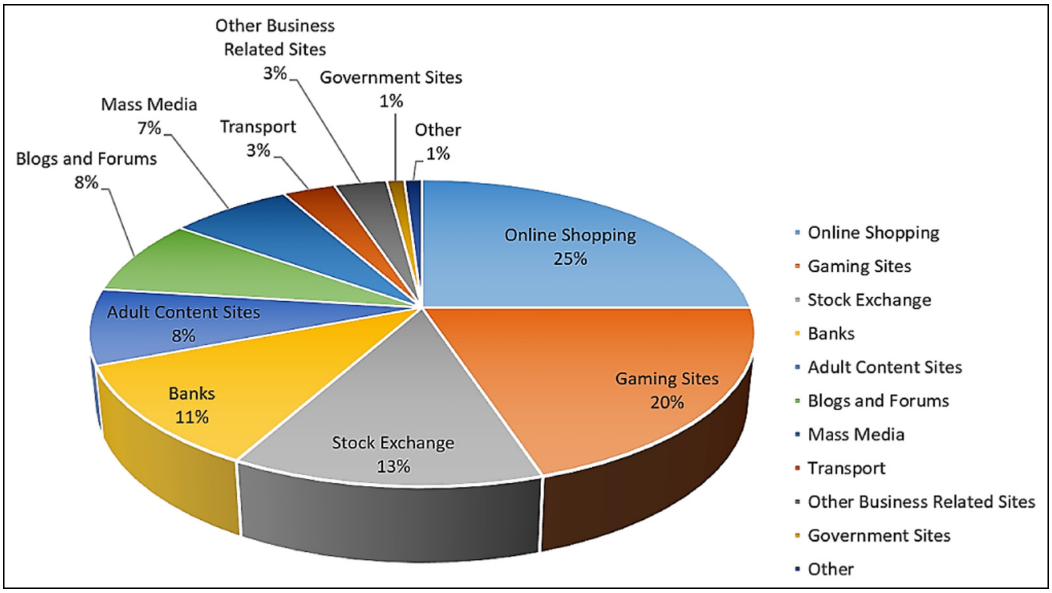

Figure 2. Breakdown of attacked sites in Q2 2011.[14]

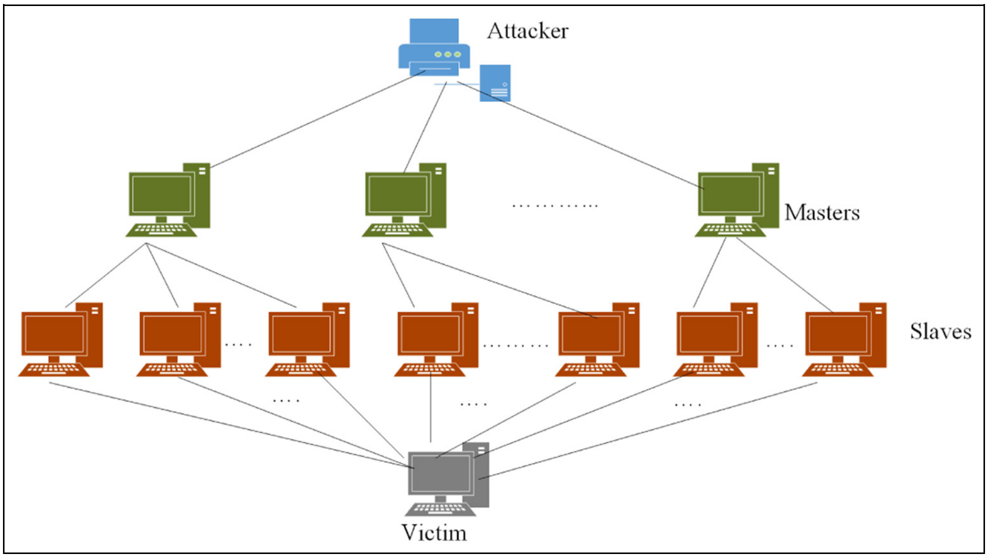

Figure 3. Structure of a DDoS attack.

performed to look for machines with security loopholes. The malicious codes installed by the attacker into these master armies work further to add more infected machines into the attack army. The slave machines are directly controlled by the masters and indirectly controlled by the attacker through these masters.

The second phase starts if a sufficient number of devices have joined as a compromised army. This compromised army is known as botnet.[19] In the second phase, the attacker transfers all necessary information such as codes and commands to the master armies which in turn send those to all slave armies to get ready for the attack.

In the final phase, the attacker commands its army to initiate and execute attacks. Thus, it attacks the victim in a distributed way and sends a large stream of packets which in turn flood the victim's system or major resources. In these attacks, the attacker usually uses spoofed IP addresses which helps him to hide the identity of the compromised devices. In most cases, this use of spoofed IP addresses also discourages the victims to filter out malicious traffic to find out the attacker.

Now we are going to illustrate the attack strategies in a detailed manner which would help to achieve a better understanding of the DDoS attack scenarios. In this illustration, we are going to describe the attack synopsis based on the degree of automation. We will also list and analyze the techniques involved in different phases of the DDoS attacks.

Attack synopsis based on degree of automation

Different phases and characteristics of the DDoS attacks can be manually set, controlled by the attacker, or it can be automated. Therefore, there are three different types of attack scenarios: manual, semi-automatic, and automatic which are introduced in the following.

Manual. In the manual attack scenario, the attacker does all of the works of the attack manually. The scanning of the machines to find security loopholes and controlling the compromised machines are performed in a manual way. This is a scenario seen in early DDoS days. Today, all of the actions have become automated and made DDoS attacks more easy and frequent.

Semi-automatic. The agent–handler attack mechanism is basically semi-automatic in nature. Here, the communications between the handler and the agents are somewhat manual as they communicate to know each other. Based on the instructions received from the attacker, these communications also set the type, the duration, and the victim of the attack. However, scanning, recruiting, and compromising the handler's machines are automatic in this attack scenario. Again, based on the type of the communications between the handler and agents, semi-automatic attacks are classified into two different types, as follows:

- *Direct communication.* In the direct communication mechanism, the handler and an agent know each other's identities for later communications. Therefore, the handler's IP address is hard coded to the attack code. The agent receives this IP address when the attack code is installed in the compromised agent machine. The motivation behind such a communication is to inform the handlers about the readiness of the agents. However, this type of communication is recognizable and also it is possible to uncover a DDoS attack through backtracking.
- *Indirect communication.* The direct communications between handlers and agents are excluded in this attack scenario. Rather, some Internet-based communication services, such as Internet chat program–Internet relay chat (IRC) channel, are used to regulate agents' work. It overcomes the drawback of the direct communication as the legitimate service of the IRC makes it difficult to identify malicious communications. Also, the distributed nature of the IRC hinders detection and investigation of the communication as well as exposure of the root of the attack.

Automatic. In this attack scenario, all of the phases and requirements to implement an attack is automated. Here, the attacker attacks a victim without any communication with handlers or agents. All the requirements of the attack army are coded in the attack code which is installed in the compromised machines and later executed to perform an attack. However, the back-doors created during the propagation mechanism remain open which can be further exploited to modify the existing codes.

Attack phases

In this section, different mechanisms involved in different attack phases are analyzed. There are three different phases of DDoS attacks which can be named as Phase I—recruiting attack armies, Phase II—propagation, and Phase III—attack. The details of these phases are listed in the following.

Phase I: recruiting attack armies. The first phase of the DDoS attacks is to generate the attack army or botnet as mentioned before. For this purpose, the attacker uses worms (self-propagating programs)[21] that infect the devices of the users by taking the advantages of their security flaws.[22] Many different techniques are used to generate this army as mentioned in different research papers.[22–26] The main theme is to scan through networks to find machines with flaws. These major techniques to infect a machine are explained as follows:

- *Random scanning.* In the random scanning strategy, already infected machines probe with random IP addresses from the IP address space to infect new machines. For example, the well-known worm Code-Red (CRv2) can be used to perform this scanning.[27] This scanning produces a huge traffic since it is very likely for the machines to be situated in different networks. Also, the lack of synchronization among different compromised machines creates high number of duplicate probes from the infected machines. As the number of infected machines increases, it increases the possibility to infect more machines. However, the extreme traffic produced by them also creates the possibility to detect the attack.

- *Hitlist scanning*. Hitlist scanning tries to reduce the initial infection time to infect a significant number of machines.[22] In this technique, the attacker creates an initial list of machines which are considered as potentially vulnerable. When a worm is released, it scans through the list and when it infects a machine, it propagates half of the list to the infected machine. In this way, only in a few seconds an active worm can infect all risky machines in its list. For example, a flash worm could infect all the venerable machines in only tens of seconds. The main overhead of this method is to generate the "Hitlist" required, but there are many different ways to produce it. For example, stealthy scanning (where an attacker applies portscans over a long period of time) and a distributed scanning (where already compromised machines are used to generate the list) are some well-known scanning strategies for the creation of the hitlist. The attackers can also use Web-crawling techniques or public surveys such as Netcraft Survey[28] in producing the list. Also, there are some cases where worms broadcast the list of their infected machines for future attacks. All of these can be applied to produce an efficient hitlist in advance. However, in the hitlist scanning, the transmission of an extremely large sized list as well duplicated transmissions may cause detection of the attack.
- *Permutation scanning*. The permutation scanning is a smart scanning technique where self-coordination is introduced to stop multiple probings of the same IP address. Also, it can determine when any further scanning will provide a limited benefit. It can also decide when to stop the process by examining progress of new infection. In this technique, a commonly shared list of pseudo-random permutation of the IP address space is used. Here, an index of the permutation is used to map the IP addresses. In the permutation scanning, for an already infected device through hitlist scanning, the new infection starts at the next point in the permutation list. However, if a device got infected from permutation scanning, it starts to infect new devices from a random point in the list. In this way, if an already infected machine is found during the scanning process, it is not infected again rather a new random starting point is selected for further infections. Also, if a worm finds that the further scanning is not beneficial anymore, it can stop its scanning and change the permutation key to produce a new permutation list. This increases the rate of the infection while reduces any duplicate infections.
- *Topological scanning*. The topological scanning is considered as an alternative to the hitlist scanning. In this scanning process, when a device got infected, a worm selects its new target from the information contained in the compromised machine. In peer-to-peer application, if a worm can infect an application, it can have a list of peers which are very attractive targets for the next infection. Thus, the topological scanning does not require a pre-produced list of devices and it can create its own list which makes this scanning attractive for the initial spread of the worm. Also, in the web server–based infection, a worm could spread itself and infect different clients and servers in a fashion similar to the spreading of a contagious disease.[22] In this case, the worms in a vulnerable web browser spread if the clients visit to the server as well as click to a particular content. The clients carry the worm to other servers where they visit and increase the infection through its propagation.[29] This type of propagation does not produce heavy traffic since it is not easy to detect such an attack mechanism. However, overall, the topological scanning depends on the activities of the compromised machine rather to the activities of the attacker. Thus, it is possible to end up with a slower and incomplete recruitment as compared to the other types of scanning.[30]
- *Local subnet scanning*. In the local subnet scanning, an already compromised host searches new targets in its own local subnet and tries to break those machines. In order to increase the number of the compromised machines, it may use jointly other already mentioned techniques. The main purpose of this scanning method is to infect as many of the local machines as possible in a subnetwork which are generally found protected by firewalls.

Phase II: propagation. The next phase after the generation of an attack army is to propagate the attack code to the compromised devices. This attack code includes information of the victim, time, and duration of the attack and so on. This section outlines this phase of the attack based on the research in Mirkovic and Reiher,[23] Patrikakis et al.,[24] and Long and Thomas[31]:

- *Central source propagation*. In the central source propagation mechanism, the attack code propagates from a central server to the compromised machine as shown in Figure 4(a). An example of this is a 1i0n worm which propagates in this manner.[31] Since every compromised device communicates and downloads the copy of the attack

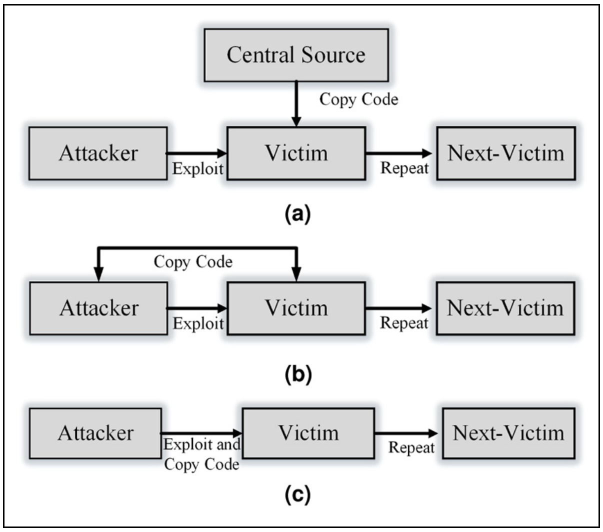

Figure 4. Attack code propagation: (a) central source propagation technique, (b) back-chaining propagation technique,[31] and (c) autonomous propagation technique.[31]

code from some central points, it introduces a huge amount of traffic in the network which may eventually lead to the attack discovery. Also, this technique has the potential to single point failure. Thus, if an attack is detected, removal of the central point may stop further infection and inclusion of the attack army.

- *Back-chaining propagation*. In the back-chaining propagation, the compromised machine downloads the attack code from the infected machine. This scenario is shown in Figure 4(b). Basically, during the infection phase, the attacker manages a method which establishes a connection between the attacker and the compromised machine. The Trivial File Transfer Protocol (TFTP) can be used in this mechanism.[24] A very well-known worm Ramen worm uses this propagation technique in the attack army creation process.
- *Autonomous propagation*. In the autonomous propagation, as its name implies, all the attack codes are transferred automatically from the attacker to the infected system during the time of the exploitation, as shown in Figure 4(c). It does not require any further communication with any more systems to transfer the attack codes. Code-Red is an example worm which uses this propagation mechanism.[31] In the autonomous propagation, the traffic required to propagate the code is significantly limited as compared to the other two methods, and thus, it poses a limited chance to discover the attack.

The third phase of the DDoS attacks is to attempt the attack. There exists a lot of techniques to initialize and execute DDoS attacks which fall into different categories. As our analysis aims to understand the attacks in an elaborated and easy way, we dedicate an entire next section to describe the third phase of the DDoS attacks.

Attack mechanisms

In order to understand research on DDoS attacks, it is important to understand DDoS attack classification mechanisms. In past decades, different classifications of DDoS attacks have been seen in the literature.[4,15,20,23,32–34] In this survey, our goal is to analyze all those attack taxonomies and to introduce a fully covered, easy to understand classification mechanism. Figure 5 introduces our classification mechanism which covers all aspects of DDoS attacks.[23] This classification is based on the impacts of the attacks in victims' networks or resources. In general, a web server or proxy server is the main victim for a DDoS attack and manages limited resources to provide its service. Thus, a general scenario to manage excess network traffic is to drop the packets which exceed some threshold limits. Dropping packets also convey the message to the senders of the packets to reduce their rates of sending. A legitimate sender responds this message by limiting its sending rate. However, the attacker takes this as a success of its initial attack execution and increases its rate as a response to the packet dropping. Consequently, the resources assigned to the victims' systems such as memory and CPU get overwhelmed and cannot continue their normal operations. Therefore, the victim eventually rejects requests from the genuine users. Also, another and more vulnerable impact of the attack is the depletion of the network bandwidths. In this case, an excessive amount of malicious flows overwhelm the network bandwidth, and this not only affects the victim but also the other systems that are dependent on this attack path. Thus, it puts a massive effect to the network and the systems connected to that network. Therefore, our classification of DDoS attacks considers these two impacts and categorizes DDoS attacks into two broad groups: bandwidth depletion attacks and resource depletion attacks. However, in reality, an attack can have both of the impacts and can impose the highest possible affects to the whole Internet. This type is named as infrastructure attack. We will cover this type of attacks in our analysis. We will also introduce another type of attacks known as the zero-day attack where the impacts of the attack are unknown.

Resource depletion attacks

The goal of resource depletion attacks is to overflow or crash all of the major resources of the system such as

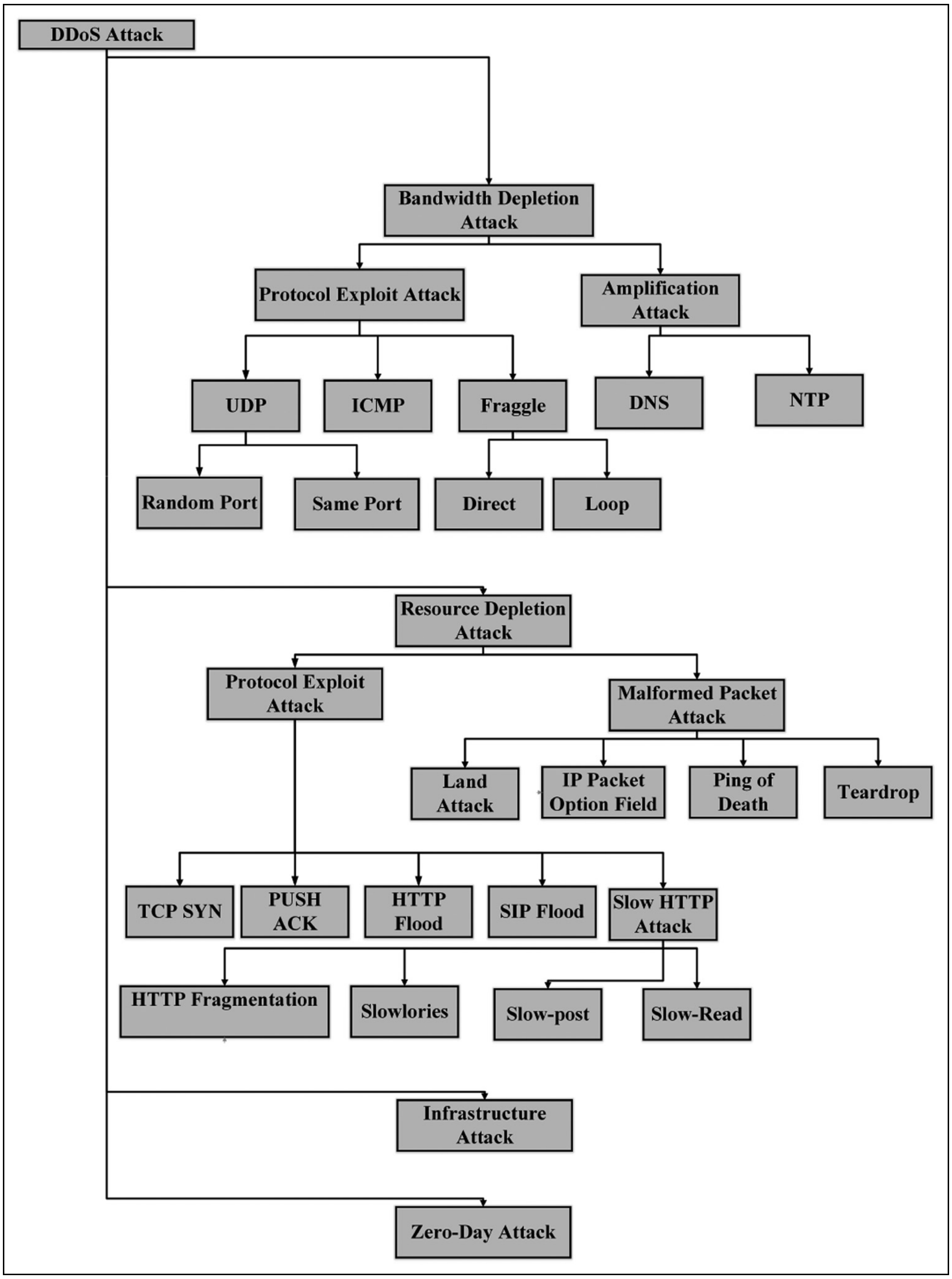

Figure 5. Different types of DDoS attacks.

memory, sockets, and CPU. There are two different ways to implement this type of attacks. In the first way, the attacker exploits some networks, transport and application layer protocols to achieve their goals. In the second way, malformed packets are used to perform the attacks.

Protocol exploit attacks. There are some major protocol-based attacks which exploit the weakness of the different network layer's protocols. This forces the victim to use all of its CPU and memory to do some memory-intensive operations. For example, attacks of this group exploit transport layer protocols such as Transmission Control Protocol (TCP)[4,33] and some application layer protocols such as Hypertext Transfer Protocol (HTTP) and Session Initiation Protocol (SIP) in execution of the attacks.[4]

TCP SYN attack. In a TCP SYN attack, the attacker exploits the three-way handshaking mechanism of TCP's connection establishment process. During the

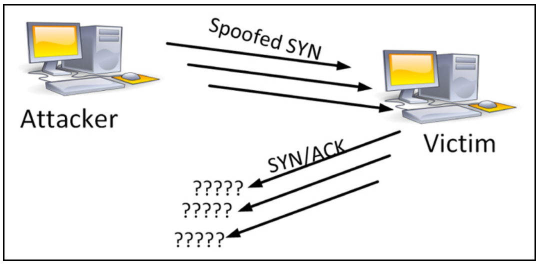

Figure 6. TCP SYN attack.[35]

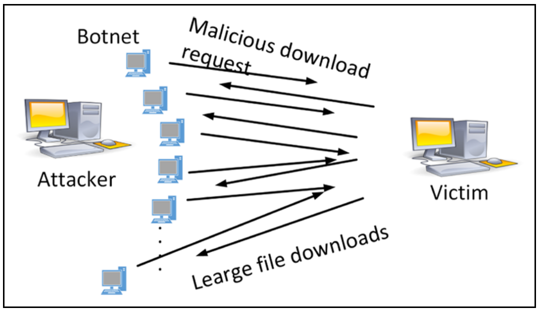

Figure 7. HTTP flood attack (exploits HTTP GET request).[36]

connection establishment, TCP requires consecutive acknowledgements between the two parties who want to create a TCP connection. This is accomplished by the three-way handshaking. In three-way handshaking, at first, the SYN packet is sent from a client to a server to begin the handshaking. Upon receiving this SYN packet, the server acknowledges the client by sending a SYN + ACK packet. Finally, as a response to this packet, the client sends back the final ACK packet which completes the handshaking and establishes the TCP connection. During this process, the server stores all of the intermediate states in the memory stacks until the connection establishes or the time-out occurs due to look up and confirmation of the identity of the client. The attacker exploits this feature and floods server's memory which eventually rejects connection requests from valid and legitimate users. In order to flood victim's memory, the attacker does not complete the handshaking process and thus creates a huge number of incomplete connections. To establish this incomplete connection, the attacker spoofs the source with non-existing IP addresses and sends the SYN packets with these spoofed IP addresses. Upon receiving the packets, the server replies with the SYN + ACK packets, but since the source IPs do not exist, it never receives the ACK packets from the sources as shown in Figure 6. However, as the server waits for the ACK packets, eventually all of its connection tables become full. Thus, the attacker floods victims memory and successfully deprives the legitimate users. It is also possible to do this attack using the genuine IP of the compromised machines. In this case, the compromised source machines ignore the SYN + ACK messages received from the victim and thus could perform a successful SYN attack.

TCP PUSH + ACK attack. The goal of this type of attacks is to run out memory and CPU processing power to hinder the legitimate users from their usual service. In this attack, the compromised botnet agents send a large number of TCP packets and set "1" to the PUSH and ACK bits of the header.[33] This forces the targeted victim to clear its memory stack and to send an acknowledgement to the client. Since the attacker floods the victim with this type of messages, eventually the victim's processing power and memory overload and run out. As a result, it cannot process the requests from the legitimate clients and thus fail to establish communications with them.

HTTP flood attack. HTTP flood attack is another example of resource depletion attacks where the application layer protocol HTTP is exploited to attack a victim. Specifically, in this type of attacks, an attacker manipulates the HTTP GET and HTTP POST requests while talking to a server or a specific application.[36] Here the concept is same as before, that is, overwhelming the resources to make the web server denies its legitimate users. In order to conduct this attack, it is required to set up a TCP connection with valid IP address. The attacker uses its botnets' IP addresses to establish the connections. Figure 7 shows an example of HTTP flood attack which exploits the HTTP GET request. Here, the attacker executes an HTTP GET request to download a very large file. It sends multiple requests from its botnets. In response to this request, the victim performs a series of actions. As a response, it is required to read the file from the back-end storage, store it into working memory, as well as cut it into multiple packets to send it to the botnets. Thus, the response includes the use of the memory and processing power of the victim. Therefore, a large number of this type of requests flood the whole resources of the victim and make it unable to respond to the valid users. In order to make the detection of this type of attacks more difficult or even impossible, the attacker could also resolve the links attached to the response and follow those links. This makes the traffic generated by the attacker looks like a normal traffic.

SIP flood attack. This is another example that exploits another application layer protocol named SIP, used in voice over IP (VOIP) call setup. An attack can be made using different types of SIP request messages (such as SIP REQUEST, SIP INVITE) or the SIP call control messages (SIP INFO, SIP NOTIFY, SIP

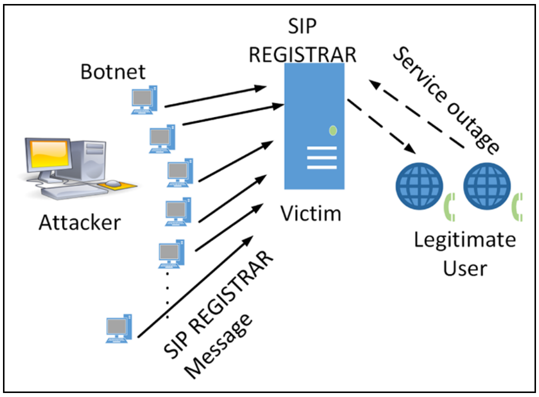

Figure 8. SIP flood attack.[37]

RE-INVITE).[37] The goal of this attack is to flood the proxy server or the SIP registration server (SIP REGISTRAR) and to consume all of its resources (CPU, memory, and network bandwidth) with the help of the attack army, as shown in Figure 8. Here, any army from the botnet sends thousands of messages to the SIP registrar server which is responsible to accept REGISTRAR requests as well as to keep records of the addresses and parameters of the user agents, unless it receives any server error message. As a result, the server overwhelms; the legitimate users experience service outage and cannot reach the server.

Slow request/response attack. In this type of attacks, it slowly consumes all of the resources of the victim. Slowlories,[38] HTTP fragmentation attack,[39] Slowpost or RUDY (R.U. Dead Yet) attack,[40] Slowreading attack[41] are some examples of this type of attacks. In Slowlories, the idea is to shut down the victim's system using only one machine. It starts the attack with a partial HTTP request. Then it sends other header information at a regular interval so that the sockets remain open. This way it consumes all the sockets, and as a result, the web server rejects its legitimate clients. So far this type of attacks cannot be prevented but can be mitigated by setting restrictions on the number of connections or the minimum transfer rate of a client. The HTTP fragmentation attack partitions a HTTP packet into tiny possible fragments and sends those through a valid HTTP connection at the minimum possible rate. Thus, it holds the connection as long as possible and makes a web site down by multiple connections from a botnet. Slowpost or RUDY or slow request attack is also an application layer attack which exploits the form submission field of the websites. In this attack, the attacker opens multiple parallel HTTP POST connections. The attacker submits the information in the form filed in a very small sized packet (1 byte) in a very slow rate. Thus, the connections remain open for a very long time and use up all the connections in the server's connection table. As a result, the server crashes and the attack becomes successful. The designer of Slow Read DDoS attack is Shekyan[42] who designed this attack to see the reaction of the HTTP server in case of slow response. The main idea behind this attack is to establish a connection using a valid HTTP request. But the attacker reads the response very slowly to make the connection open as long as possible. Thus, multiple such requests consume all the connections and make the server unavailable for the legitimate clients. In general, these slow HTTP attacks can be protected if a set of observations and rate limiting are applied to the flow. For example, rejecting unsupported HTTP connections or applying limit on the rate of the incoming data or limiting the length of the header or the body of the message and so on, could be a good protective step for slow HTTP attacks.[43]

Malformed packet attack. The main idea behind a malformed packet attack is to attack a victim using a deformed packet which may confuse the victim and as a result cause crash to the system. We outline some of these attacks as follows:

1. *Land attack*. In this type of attacks, it sets the victim's IP address to the packet's source and destination IP addresses.[33,34] Whenever a system receives this type of packet, it replies back to itself which in turn creates an infinite loop. As a result of this, the system crashes eventually.
2. *IP packet option field attack*. This attack targets the optional fields of an IP packet and randomizes its value. For example, if in an IP packet, all qualities of service bits are set to 1, it causes the victim to apply some additional time to analyze the packet. Thus, it can inundate the processing ability of the victim if a flood of packets arrives with such deformation.
3. *Ping of death attack*. In ping of death attack, an attacker intentionally forms a data packet that exceeds the maximum packet size which causes the victims to freeze or crash.[34,44] This attack can be initiated only by the attacker without the need of a botnet. Fortunately, current host systems are protected from this type of attacks.
4. *Teardrop attack*. This attack involves manipulation of the offset value which in turn generates errors in fragmentation and reassembly of packets. Basically, the attacker sends fragmented packets with overlapping offset numbers. Thus, during the time of the packet re-build, invalid packets are created and crash or reboot the target machine.

Bandwidth depletion attack

The bandwidth depletion attack is another important type of attacks in the DDoS world. The attacker's goal is to consume all of the network bandwidths of the victim's system using the attack army. As a result, the victim denies service to the legitimate users for a small to large amounts of time until the attack is mitigated. The bandwidth depletion attack can also be protocol exploited,[45] and it can be done by amplification where a reflector or amplifier is involved to increase the attack density as well as damage to the network.[46] We will introduce this attack mechanism with related examples.

Protocol exploited attack. Protocol exploited attacks can use a transport layer protocol such as User Datagram Protocol (UDP) or a network layer protocol such as Internet Control Message Protocol (ICMP). Next, we will outline major types of attacks found in the research.[4,15,20,23,32–34]

UDP flood attack. The UDP flood attack is a very common DDoS attack where an attacker sends a large stream of UDP packets from its attack army. Here the attacker can target a specific or a random port of the victim to inundate it. Generally, when a UDP packet is received to a system, it tries to identify the type of the application that is waiting on the destination port. When it becomes sure that no application is waiting, it responds with an ICMP packet as shown in Figure 9. The attacker uses spoofed IP addresses and continues sending the packets until all the bandwidths are consumed and the victim surrenders from its normal operation.

ICMP flood attack. The ICMP flood attack, also known as the ping flood attack, exploits the IP layer protocol ICMP's ICMP_ECHO_REQUEST packets. This packet (ping) is used to check whether a remote host is alive or not. In DDoS attacks, the attacker sends this packet using the broadcast IP address. Thus, it is delivered to all of the machines in the victim's network. The machines will reply to the spoofed source address that targets the victim with ICMP_ECHO_REPLY packet. Also, the attacker may use an intermediary network to inundate the victim. As a result, the bandwidth in victim's network becomes saturated, and consequently, it rejects requests from the legitimate users. An example of this type is the "Smarf" attack which uses some intermediary networks, also known as the reflectors to intensify the attack. This scenario is shown in Figure 10.

Fraggle attack. The fraggle attack is similar to the Smarf attack and is also obsolete nowadays.[47] It sends UDP_ECHO packets to the network amplifiers to flood victims bandwidth or can send it to a particular port to create an infinite loop. Smarf and fraggle attacks are also known as the amplification attack which uses reflector(s) as their attack launchers.[20] Any IP host who returns a packet in response to a received packet is known as the reflector. Therefore, routers, DNS servers, or web servers are the examples of reflectors. These reflectors trigger the attack by sending replies to the victim as responses to the received packets which contain spoofed source IP address as the IP address of the victim. The reflectors use their own legitimate IP addresses so they are detectable. However, the attacker who involves the reflectors in the attack remains hidden as it has spoofed its source IP to the IP of the victim.

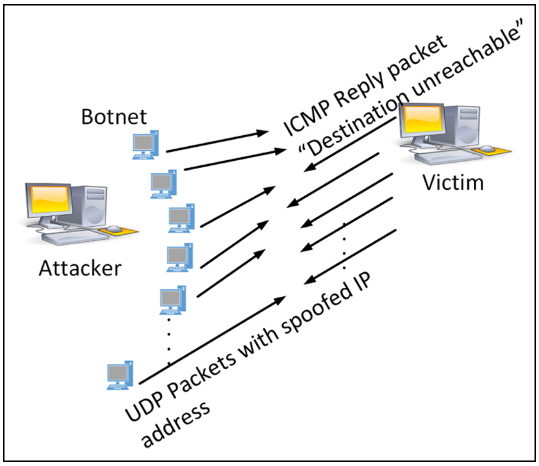
Botnet
ICMP Reply packet "Destination unreachable"
Victim
Attacker
UDP Packets with spoofed IP address

Figure 9. UDP flood attack.

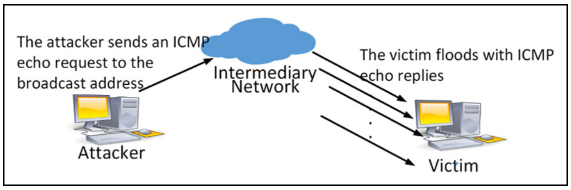
The attacker sends an ICMP echo request to the broadcast address
Intermediary Network
The victim floods with ICMP echo replies
Attacker
Victim

Figure 10. ICMP flood attack.[4]

Amplification attack. In this section, we are going to outline two very similar and common amplification attacks: DNS amplification attack and Network Time Protocol (NTP) amplification attack. The main idea behind these types of attacks is to generate a large response for a very small request and directing those responses to the victim which eventually consumes all bandwidths of the victim's network.

DNS amplification attack. It is one of the most common attacks of today's world which targets victim's network bandwidth. In one example, the motivation of

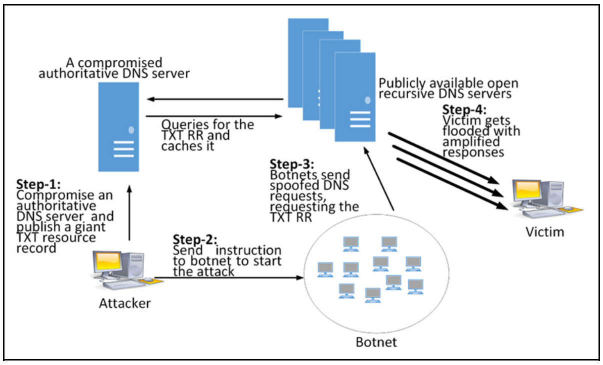
A compromised authoritative DNS server
Publicly available open recursive DNS servers
Queries for the TXT RR and caches it
Step-1:
Compromise an authoritative DNS server and publish a giant TXT resource record
Step-2:
Send instruction to botnet to start the attack
Step-3:
Botnets send spoofed DNS requests, requesting the TXT RR
Step-4:
Victim gets flooded with amplified responses
Attacker
Botnet
Victim

Figure 11. DNS amplification attack.[4]

an attacker is to exploit the vulnerable features of a DNS to amplify an attack in a much larger scale.[48–50] This attack is also an example of a reflection attack which uses multiple open recursive DNS servers to send a huge number of UDP packets to flood a victim. Using different types of techniques involved in amplification, an attacker can raise the volume of the attack traffic which could result a catastrophic affect to the most secure victim's system. The general scenario of the attack process is same as all of the other DDoS attacks: sending reflected responses to the spoofed source IP (victim's IP) to consume network bandwidth. In general, the response made by a DNS is large as compared to the query sent to it. Therefore, by increasing the size of this response message, more bandwidth is consumed than the normal situation. An attacker can compromise an authoritative DNS server to achieve this.[4] Figure 11 illustrates this scenario. Here, at first, the attacker exploits authoritative DNS server and using that server, it discloses a sizable resource record (RR) of TXT type. This TXT type RR is used in DNS to make it possible to store information that have not been classified previously.[51] This can include host name, server name, or some information about a server or a data center. The attacker manipulates this record to increase the message size. Then the attacker launches the attack by the botnet and instructs them to send requests to the open DNS recursive servers and ask for the large messages. After resolving the requests, the open DNS servers send the amplified responses to the spoofed source address which is actually the address of the victim. Theoretically, in order to generate a 10 Gbps flood to the victim, only 140 Mbps initial traffic is sent from the botnet.[4] Also, to amplify the attack, the attacker may send DNS request using the extended DNS protocol (EDNSO). This extension supports large DNS messages. To increase the size of the message, the attacker can also use Domain Name System Security Extension (DNSSEC). Here, they can apply some cryptographic features which increases the message size. However, according to US-CERT,[52] most of the attacks observed by them use DNS "ANY" request. In a single request, it returns all possible known information regarding a DNS zone. Thus, the response gets larger which boosts up the traffic toward the victim.

NTP amplification attack. This is very similar to the DNS amplification attack and it exploits NTP.[53] NTP is used to synchronize the clocks of the machines connected to the Internet.[54] In the DDoS attack, the attacker sends MON_GETLIST command to the NTP server which responds with the last 600 queries that have been made to the server. Thus, the response message is huge (approximately 19X bigger) as compared to the request message.[53] The attacker spoofs the source IP address and directs all those responses to the victim's machine. Thus, these amplified responses overwhelm victim's network bandwidth and inhibits legitimate users to get into the server.

Infrastructure attack

The most catastrophic type of DDoS attacks is the infrastructure attack. The aim of this attack is to damage significant crucial elements of the Internet. Thus, it not only targets the bandwidth of the network but also the resources (memory, CPU) of the targeted system. For example, infrastructure attack aims the DNS, especially the root DNSs as they are the top hierarchical service points and provide services to all users of the Internet around the world. Since the DNS maintains a hierarchical structure, an attack of this type which targets only the root name servers could not put very serious impacts on the Internet service of the whole world. Commonly, attackers use DNS flooding techniques to launch the attack. In the DNS flooding attack, compromised attack army or botnet sends normal UDP requests to the DNS server. However, the amount of such requests is so enormous that it floods the system and eventually all of the resources are consumed. Thus, the system denies all legitimate requests for a significant amount of time. On 21 October 2002, an infrastructure attack was launched which targeted all 13 root name servers.[55] This attack did not come out as a very successful one but introduced the attack of this type. Recently, exactly after 14 years, in 21 October 2016, the largest DDoS attack has targeted a major infrastructure known as the Dyn DNS.[1] This system provides DNS address resolution service to more than 3,500 enterprisers which also include some renowned organization like Netflix, Twitter, Linkedin, and so on.[56] It provides high-performance service with managed DNS infrastructure which includes numerous data centers located in multiple continents. It possesses the ability to analyze 3 billion data points per day. According to the source Dyn,[56] in each month, 50 considerable DDoS attacks are detected and mitigated by the experienced DNS experts of Dyn. However, the irony is that in October's DDoS attack, this high-performance system went down for a significant amount of time. The attackers involved millions of poorly secured IoT devices in this attack which are not powerful computers but some smart devices such as smart fridge, web cams, and digital video recorders (DVR).[57] Thus, it introduces a significant threat in the current cyber security world. This is because, currently the number of IoT devices is 28.4 billion and this number is increasing in a massive rate.[58] It was reported that in order to infect these huge number of devices, an open source botnet malware, "Mirai," is used. This malware is designed to infect IoT devices and launch DDoS attacks based on instruction from the attacker.[59] Finally, this attack has again bring up the necessity of a globally cooperated solution of the threats of the current world.

Zero-day attack

A zero-day attack happens in day 0 using some unknown security loopholes or vulnerabilities. It is called zero day because the vulnerabilities of the system are known at day one after the attack. Different security organizations or private software firms offer incentives and rewards to report the zero-day vulnerabilities.[60] The impact and signature of this type of attacks is also unknown until the attack is launched.

Prevention against DDoS attacks

Prevention against DDoS attacks is the most desirable defense technique to fight against the DDoS attacks. Basically, as mentioned in the previous section, DDoS attacks put an immense threat to the resources of the victim (CPU, memory) as well as to the network bandwidth and infrastructure. Therefore, if an attack has been already launched and become successful, it may cause significant compromise to the victim's system. Thus, protection against DDoS attacks is more effective against DDoS attacks since it ensures prevention of the DDoS attack traffic as well as manages large attack load before it may cause the attack to be successful. This ensures normal operation of the victim.

DDoS attacks have become very common in today's cyberworld. According to the quarterly report of Kaspersky Lab, the largest number of attacks that they have recorded in a day in the last quarter of 2016 is almost 2000.[61] Also, in this quarter, the longest attack was recorded as 292 h long. Moreover, when a victim gets attacked, it may experience more attacks in course of time. It was reported in the fourth quarter executive summary report of Akamai, on average a target may experience 30 DDoS attacks per quarter as well as three to five attacks a day.[62] Again, the research report published in 2012 by the security and data protection researcher of Ponemon Institute[63] shows that the average cost incurs for 1 min downtime as a result of DDoS attacks was approximately US$22,000 which may range from US$1 to US$100,000. These statistics reveal the immense need of the success of DDoS protection methods. In this section, we are going to outline all major protection/prevention techniques against DDoS attacks. Lots of researches are found which cover those techniques.[4,15,20,23,32–34,64–67] We will provide a comprehensive summary of all significant prevention methods to see their success and remedies. In our analysis, we will follow the classification shown in Figure 12.

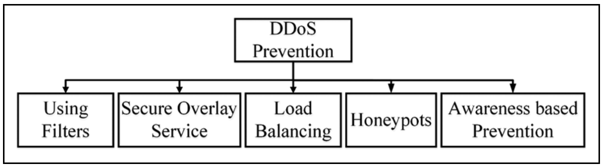

Figure 12. DDoS prevention.

Prevention using filters

In order to prevent the attack traffic, it is very important to filter them out. Filtering techniques mainly prevent a victim from the attacks as well as from being an unaware attacker. Basically, all filtering techniques are applied to the routers which ensure that only legitimate traffic can get access to a system. In this section, we are going to cover different filtering techniques found in the literature along with their success and failure in DDoS prevention.

Ingress/egress filtering. A very common and well-known filtering technique is the ingress/egress filtering. These techniques prevent traffic with spoofed IPs to enter into a protected network. Basically, ingress filtering filters the malicious traffic destined to a local network and egress filtering discards the malicious traffic leaving a local network. Ingress filtering defined in RFC 2267[68] allows those traffic to enter the network which matches with a predefined range of domain prefix of the network. Thus, if an attacker uses spoofed IP address which does not match with the prefix, it is discarded in the routers. Thus, these filtering techniques ensure prevention from a significant amount of DDoS attacks where spoofed IP is used. However, it is not a useful mechanism in the cases where the valid IP addresses of the botnets are used as a source IP during the attack. Also, the success of these filtering depends on the knowledge of the range of expected IPs for a port which is not always achievable for the complicated topologies used in different networks. Moreover, if an attacker uses some spoofed IP addresses which fall into the valid address range, the filters in the routers cannot detect the malicious traffic in such cases. Again, for these filterings, itunneling is required for mobile IP users so that they are not filtered by the routers using ingress/egress filtering. Finally, as it does not ensure incentives to the ISPs, it is partially deployed in the networks. In other words, many ISPs do not enforce this approach.

Martian address filtering and source address validation. Martian address filtering is defined in RFC 1812 and works for filtering spoofed IP addresses that are generated from a limited set of addresses.[69] This filtering ensures that a router must not forward any packet which has an invalid source or destination IP address. These invalid IPs range from the reserved or special IP addresses as well as the unallocated range of IP addresses. It also ensures that any packet with the destination IP address 255.255.255.255/32 must be discarded at the router.

Source address validation is also specified in RFC 1812, which is also implemented in the routers. In this filtering, the router compares the source address of a packet with the logical interface of the packet where it is received. If this interface does not match with the interface where the packet needs to be destined to reach the specified source address, the router discards the packet. Thus, it can filter packets with spoofed source IP addresses. However, the rate of the false positive may become high for the asymmetric routes of the Internet. This asymmetric nature does not guarantee the match of the interfaces upon receiving or return of a packet from a specific source address. Thus, this filtering may discard a large number of legitimate traffics. However, the essential issue here is that not all routers in the Internet implement these approaches.

Route-based packet filtering. Route-based packet filtering (RPF) filters packets with spoofed source IP addresses. This filtering technique increases the scope of the ingress filtering by providing service to the core routers.[70] It provides filtering based on the route information of a packet in each link of a core router. It depends on the principle that each link of the core router accepts traffic from only a limited number of source addresses. Thus, an IP packet which has a different source addresses than this set of addresses is discarded by the core routers as it appears to be spoofed to the router. In order to implement this technique, it requires information of the Border Gateway Protocol (BGP) routing topology. According to the simulation of Park and Lee,[70] a significant success of the technique will be achieved if 18% of the autonomous system (AS) implements this filtering technique. However, this number is found impractical in current Internet scenario. Also, in order to include the source addresses in the BGP message, it is required to modify the scheme used in BGP messaging. This increases the message size as well as the processing time of the BGP messages. Moreover, if the routers do not maintain updated information, this technique can discard legitimate packets for unwanted root change (for link failure, policy change, etc.). Also, as RPF filters packets based on BGP messaging information, an attacker can deceive root information as well as filtering rules by stealing BGP sessions. Finally, the attacker can carefully choose IP addresses which do not resemble the spoofed IPs. This can make the method ineffective in protecting DDoS from spoofed IP. Duan et al.[71] extend the idea of Park and Lee[70] by designing a packet filtering mechanism which considers update messages of local BGPs to filter out spoofed IPs. This method is easy to deploy on the current architecture which relies on the BGP routing protocol. It also reduces the rate of false positive and simplifies IP traceback process.

Source Address Validity Enforcement protocol. Source Address Validity Enforcement (SAVE) protocol is an improvement of the previously mentioned RPF.[72] It enforces the routers to send messages containing updated source information to each destination routers connected to a source. Then, each router updates its forwarding table with current information and uses it to filter the packets based on the methods of RPF. Thus, it overcomes the problem encountered in the RPF for the asymmetric and dynamic nature of the Internet. However, implementation of such protocol requires a change in the existing routing protocol which is a time-consuming process. Also, a partial deployment of the protocol does not ensure full success in filtering spoofed IP addresses.

Hop-count filtering. Hop-count filtering also filters packets with spoofed IP addresses.[73] In this method, the authors have used the concept that it is not possible to alter the number of hops of an IP packet when it travels from a source to a destination. Therefore, the authors have used this count to determine the validity of a packet. In this method, time to live (TTL) value is used to count the number of hops. This hop counts are stored in a mapping table against each source address. Upon receiving a packet, the number of hops required for this packet is calculated and matched against the mapping table. A packet is detected as spoofed packet if a mismatch is found in this comparison. If a flow of packets is identified as a flow of spoofed packets, the filter discards those packets as a prevention of an attack. The deployment of such a technique is easier as it requires implementation in the victim's system. However, it has a major limitation in the process of hop count. As this method counts the number of hops based on TTL, the number of false positive is larger in this method. This is because the initial TTL value is usually different for different operating systems. Also, the attacker can forge valid hop counts in their packets which allow the packets to pass the filter. Finally, for a flood of malicious packets, the system cannot perform the calculation and comparison and thus become the victim of the attack.

History-based filtering. This is another packet marking–based filtering mechanism where the history of the normal traffic is used to filter out the malicious traffic.[74] In this method, the destination of an attack maintains a database of IP addresses. This database contains the IP addresses which are commonly found in the destination. Therefore, when a bandwidth attack is targeted to this system, the system only allows those IP addresses that appear in their database and discards all other IP packets. However, if an attacker can simulate its attack traffic as a normal traffic, this filtering technique cannot successfully detect and discard the malicious flow.

Path identifier. Path identifier (Pi) method filter outs packets based on a path identifier that identifies the path of the attacker.[75] It is a deterministic approach where each packet is stamped with an identifier based on the path it has traveled. The packets that travel the same path contain the same identifier. Thus, if the victim can identify a packet traveling from the attacker, it can filter all the subsequent packets sent by the attacker. However, this mechanism works well when half of the routers get involved to mark the packets. Also, as it works with a small sized identification field, there remains the possibility that different paths will show the same path information. Thus, it increases the chance of false-positive and false-negative results.

Later, Yaar et al.[76] have proposed an improved version of the Pi method named StackPi that improves Pi's performance in terms of incremental deployment. Also the improved filtering mechanism is capable of detecting malicious flows based on just a single packet. According to the authors, this method can provide a reasonable amount of DDoS protection only if 20% of the routers implements this marking scheme. However, this method does not consider the detection of spoofed IP packets rather it marks and filters a malicious packet based on the deterministic packet marking mechanism. Also, a host or router who looks for the path information through this method may need to have some supportive software as well as expense of processing which are the barriers to deploy this method.[77]

PacketScore. It is a proactive filtering technique which uses Bayes' theorem to calculate conditional legitimate probability (CLP).[78] This CLP is used to determine the likelihood of a legitimate packet based on the baseline profile value and the attribute value of a packet. The packet filtering works based on this CLP value and a dynamic threshold. As the filtering takes into account the statistical analysis, this method works well for new attack signatures as well as non-spoofed attack traffics. However, this method requires a large amount of storage to deal with the increasing number of attack attributes. Later, a method named, ScoreForCore, has been proposed in Kalkan and Alagöz[79] which works based on the PacketScore mechanism but introduces efficient property to select appropriate attributes of the present attacks. However, this method requires collaboration of the peers to gather knowledge of the network.

Secure overlay

This is another preventive mechanism against DDoS which protects a subset of the networks.[80,81] The idea behind this method is to built up an overlay network on top of the IP network. This overlay network is the entry point for the outside network to establish a communication to the protected network. It is assumed that the isolation can be achieved if a protected network hides its IP addresses or uses a distributed firewall. This firewall ensures that only trusted traffic from the nodes of the overlay network can get entry to the protected network. Although this mechanism ensures prevention from the DDoS attack, it is applicable to a private network and is not appropriate for a public server.

Honeypots

A honeypot/honeynet is an interesting mechanism of DDoS prevention.[82,83] Here, honeypots/honeynets are some less secure systems which attract attackers to attack them. A honeynet mimics a legitimate network to trick an attacker so that the attacker thinks that it has attacked the actual system. Thus, the actual system remains protected. Not only that using a honeypot, it is also possible to extract important information (records of attack activity, tools, and software used for the attack) about an attacker. This information is further used to detect and prevent a DDoS attack and its attacker. However, this technique also suffers with some drawbacks. For example, the static and passive nature of the honeypots/honeynets does not ensure complete disguise from the attacker. Also, if the honeypot/honeynet cannot detect an attack using its signature-based detection tools, attack packets are forwarded to the actual destination.

Load balancing

This is an approach which tries to balance the loads of different systems so that no one system gets overloaded.[84] The result of the load balancing helps to gain the optimal productivity as well as the maximum uptime. In cases when a server faces a DDoS attack, a load balancer ensures resilience as it reroutes traffic to another active and un-attacked servers. In order to ensure the maximum load balancing, a bandwidth increase is required on all critical connections. A good number of replicated servers and data centers are also required to ensure elimination of single point failure. It also helps to reduce the surface of the attack and introduces difficulty to exhaust existing resources and link saturation.

Prevention based on awareness

Recent DDoS attacks mostly the IoT botnet–based DDoS attacks require awareness of the IoT users. This is because the IoT devices are very poor in terms of security. Also, some DDoS attacks can be prevented if the general user takes preventive measures in their own system. It not only ensures their security from being attacked but also from being a zombie of the attack army. Followings are some initiatives from users' side which can work to prevent a DDoS attack.

Changing IP addresses. In this technique, the computer system changes its IP address to invalidate an old address which may be the potential target of the DDoS attacks.[85] This process is successful if the attack is IP address based. However, this also incurs a lot of overheads such as updating different entry information. This method works successfully as long as the attacker does not get informed about the new IP address.

Disabling unusual services. This is a prevention approach for DDoS. Some services such as UDP echo, character generator services can cause threats of DDoS attacks. Therefore, disabling such services can protect a system from some type of DDoS attacks. Also, in order to prevent creation of IoT botnets, it is recommended to disable remote access options (Telnet and SSH) to the IoT systems.[59]

Applying security patches. It is also required to update all security patches regularly to ensure that the system is not affected by bugs or worms. Also, in order to prevent IoT botnet's generation, it is mandatory to change the default or generic passwords of the IoT devices.[86] This is an important awareness from the users side that can fight against the massive IoT–based DDoS attacks. However, the irony is that most of the users of the IoT devices are not aware of the threats or even they do not know the name of the DDoS attacks. When an attacker compromises a device for the DDoS army, the user of the device does not notice any change in their performance or behavior. Thus, stealthily they help in the attack without even knowing the name of the attack that they are participating.

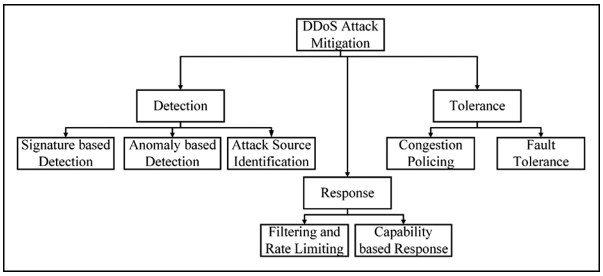
DDoS Attack Mitigation
Detection
Tolerance
Signature based Detection
Anomaly based Detection
Attack Source Identification
Congestion Policing
Fault Tolerance
Response
Filtering and Rate Limiting
Capability based Response

Figure 13. DDoS mitigation.

DDoS mitigation

Prevention of DDoS is one part of defense from DDoS. This part is a very important one to defend significant number of DDoS attacks. However, still there are remaining threats of the DDoS attacks with new attack signature and patches. Therefore, the next phase of the defense that is the field of the DDoS mitigation has a huge number of research activities. In this section, we will try to introduce significant works of this field. Mitigation of DDoS comprises three different mechanisms: detection mechanisms, response mechanisms, and tolerance mechanisms, as shown in Figure 13.

Detection

Detection of the attack is an important step to DDoS mitigation. Detection is very simple as the performance of the service or system degrades dramatically when an attack occurs. However, it is always challenging to differentiate the malicious flows from the legitimate flows. Sometimes, a response demands detection of the malicious events where another demands identification of the source of the attack. Two different techniques are found in the literature to detect malicious flows: signature-based detection and anomaly-based detection. On the other hand, attack source identification techniques are also important to identify the source of the attack.

Signature-based detection. Signature-based detection mechanisms rely on known DDoS attacks to identify the attack signatures which differentiate a normal traffic from a malicious one.[87] Thus, they are successful in detecting known DDoS attacks. However, any variations in already existing attacks remain unnoticed by these detection mechanisms. Also, as the techniques are unaware of the new signatures, they do not work for any new attack types that have not been seen previously. In this section, we will discuss about some well-known signature-based detection mechanisms.

Bro. It is a real-time network intrusion detection system which works by monitoring the network traffics of an intruder.[88] It comprises many powerful features such as a regular expression–based signature language, a robust attack detection approach, opportunities to extend for new events. However, it can suffer from DoS attacks which use algorithmic complexity. Also, it requires to create its attack signature as well as scripts to handle its event manually. Thus, it is important to engage an expert administrator in such cases.

Management information base traffic variable correlation. This method analyzes management information base (MIB) traffic variable to find out signatures of an attack.[89] Based on a three-step signature extraction method, a set of attack signatures are extracted. Based on this, the monitored traffics are identified as attack traffic or legitimate traffic. This technique can detect statistical irregularities specific to TCP, UDP, and ICMP packets that takes part in DDoS attacks. However, the detection accuracy of the method is found minimum if this method is evaluated in an environment with controlled traffic. Also, this method is not applicable to all over the Internet.

SNORT. SNORT is a very popular network intrusion detection tool.[90] It is a rule-based lightweight tool for detection of a broad range of attacks and probes. In order to increase its range of attack detection, it has combined signature-based detection and anomaly-based protection. However, as SNORT depends on exact pattern matching, it may introduce bottleneck to the performance of the system for the large traffic as well as speed of the Internet.

Spectral analysis. The techniques found here perform spectral analysis to differentiate between attack traffic and normal traffic. For example, in the work by Cheng et al.,[91] power spectral density of the packets is used to detect attack flow. Also, another approach is found in the work by Hussain et al.[92,93] where spectral analysis is performed to categorize the characteristics of an attack traffic.

Anomaly-based detection.

Anomaly-based detection mechanism can handle attacks with new signatures as well as newly appear attacks.[94] However, the selection of the threshold value to differentiate between attack traffic and normal traffic is an open challenge for these techniques.[95] In this section, we will analyze some popular anomaly-based detection methods.

MULTOPS. Multi-Level Tree for Online Packet Statistics (MULTOPS) is a tree-based attack detection technique especially designed to detect bandwidth flooding attacks.[96] This method detects attack traffic based on the disproportionality of the packet rates. Each time when the packet rate of a subnet reaches its threshold value, MULTOPS creates a new sub-node. This ensures firm packet tracking rates. However, it assumes that the rate of the incoming and outgoing traffic is proportional, which is not a practical assumption for today's multimedia message streams. Thus this technique can suffer with a high false-positive rate. Also, the attacker can target MULTOPS by memory exhaustion attack. Finally, Abdelsayed et al.[97] have proposed an improved method over MULTOPS. This method uses a data structure called tabulated online packet statistics (TOPS), which is more memory efficient than MULTOPS.

DWARD. DWARD is a DDoS attack detection and rate limiting technique which works near the source of the attack.[98] It also detects anomaly based on traffic statistics acquired from source edge routers during a monitoring phase and the specification of the network traffic in application and transport protocol. Based on these information, it applies a dynamic rate limiting on three different traffic categories: legitimate, suspicious, and attack traffic. However, this method also may wrongly classify a legitimate traffic as suspicious or attack traffic. Also as this method works near the source of the attack, it is required to deploy the method widely. However, this deployment does not ensure any incentive to the operators of the network.

Other methods. Also, there are a lot of other methods found in the literature[99–102] that try to differentiate attack traffic and legitimate traffic based on the analysis and detection of anomalies in traffic flows. Xie and Yu[103] have proposed a method to identify anomalies in the browsing behavior of the users based on hidden semi-Markov model. Their method works to detect application layer DDoS attacks based on the browsing behavior of the users. However, this method cannot handle attacks that is attempted by dynamic web page or it cannot properly detect an attack if the attack follows the behavior of the proxy servers. In the work by Xiang et al.,[104] a method is presented to detect low-rate attacks based on the information metrics. This is also an anomaly-based DDoS attack detection technique which works better than the Shannon metric or Kullback–Leibler's divergence methods. Also, the paper includes an IP traceback mechanism which is capable of detecting attack and the attackers and can also filter out the attack traffics. However, for these methods, it is found to be difficult to work with the network-wide nature of the attack.[105] Later, Jiang et al.[105] have implemented a method to detect abnormality in the network-wide traffic using the analysis of transform domain. Basically, they have detected the abnormality based on the analysis of the topological information of the network as well as network-wide traffic. Also, they have applied time window analysis which reduces computational overhead and increases detection performance. Also, in the work by Ma and Chen,[106] an entropy-based anomaly detection method is proposed which uses the variation of the Lyapunov exponent. In this method, the authors have considered the source and destination IPs consequence in the attack traffic. Bhuyan et al.[107] proposed a method based on extended-entropy metric which outperforms the previous method in terms of detection accuracy. This method works with less features from the IP traffic and uses less hop counts to traceback to the LAN of the zombie attack armies. Also, Jyothi et al.[108] have proposed a method that can adapt the threshold value to increase the complexity to the attacker who wants to adjust the traffic based on the threshold value. In order to do so, it monitors the behavior of the host or service systems. They have also taken into account the hardware performance counter (HPC) values and the statistics of correlating traffic of the network and machine learning approaches. All of these are used to detect DDoS attacks which achieves 99.8% accuracy in detection.

Attack source identification. Detection techniques work to identify malicious traffic of the network flow. Therefore, these techniques mainly work near the target of the attack. Thus, if filtering or response mechanisms are applied to the victim's side, it cannot ensure the minimum false-positive rate. That is, there exists the possibility to discard legitimate traffics for the methods that try to apply filtering in the victim's network. Therefore, it is more efficient if the attack source can be identified and filtering or rate limiting can be applied there. It will ensure the acceptance of more legitimate traffics in the system. The identification of the attack source as well as the network path of the attack traffic is known as IP traceback. In this section, we will introduce some popular methods of attack source identification.

Probabilistic packet marking. This technique probabilistically encodes information into the packet header for the receiver who uses this information to reconstruct the path of the packet traveled. Thus, without any extra traffic for the traceback or any external communication and support, this mechanism can identify the attack source by an attack path reconstruction. The first IP traceback method based on probabilistic marking of the packets was found in the work by Savage et al.[109] In this method, a router marks a packet with some partial information of the path that the packet has traveled. Here the identification field is used to store the mark. The method could also be used to do post-mortem of the attack. However, the computation complexity to reconstruct the path for multiple attackers is too large ($O(n^8)$ for n attacker) for this method.[110] Also, it produces a large number of false-positive results for multiple attackers. Another method similar to the previous one has been implemented in the work by Dean et al.[111] This method has used algebraic techniques to make it a robust one. Thus, their technique achieves more design flexibility. It is also possible to filter out the noise generated by the attacker as well as it can identify and separate more than one attack paths. However, this method is less efficient when many attackers get involved to the attacks since the number of packets required to reconstruct the attack path increases in a quadratic way as the number of attackers increases. Also, since this method does not use the distance table, an attacker can forge marking information which eventually produces a wrong attack path. Later, in the work by Song and Perrig,[112] two schemes have been proposed. One is the advanced and the other is the authenticated scheme for marking. The authors have considered the minimum routing and network overhead. The method has minimum computing complexity and higher precision for an extensive DDoS attack as compared to the previously mentioned work in the work by Savage et al.[109] Also, the authentication scheme ensures protection from the forgery (using a compromised router) of the attacker. The methods work depending on a map of the upstream routers. Although the construction or collection of this map is reasonable and practical, it requires some offline or peace time work for getting the map. Law et al.[113] have proposed an enhanced method based on probabilistic packet marking (PPM) which can determine the rate of the traffic. Thus, based on this traffic rate, the method can identify the attacker's node which generates the maximum traffic. In order to improve the scalability of the PPM mechanism in the work by Goodrich,[114] a method that utilizes big checksum cord is proposed. The author has mentioned some advantages of using this checksum cord. For example, usage of such checksum improves the efficiency of the reconstruction algorithm. Also, it serves as an associative address and integrity validator. It also helps to reduce the forgery that can be done by the attacker by inserting forge messages that cause collisions with the legitimate messages. In order to reduce the number of packets to reconstruct the attack path, an adaptive probabilistic marking scheme is proposed in the work by Tian et al.[115] Here, the TTL value is used to calculate the distance that a packet has traveled and also has used this distance to determine the probability of marking. This method reduces the reconstruction time to 20% as compared to the previous methods. It also defends spoofed packet marking. Peng et al.[116] introduce a PPM scheme with adjustable probability rather than a fixed one based on the routers' positions in an attack path. Another PPM scheme based on dynamic probability and Luby transform code (LT code)[117] is also proposed in the paper to reduce the number of packets to detect the source as well as packet collisions. Fadlallah[118] propose a method where the routers can adapt their packet marking probability. It is done based on the overwrite rate of the packet marking. This increases the probability of the packets from the faraway routers. Thus, the marking information from the distant routers reach to the victim with increased probability. As a result, it reduces the number of packets for the reconstruction of the traceback path.

Deterministic packet marking. Deterministic packet marking (DPM) uses some deterministic methods to mark packets of the routers which are nearer to the source of the attack. As compared to the PPM methods which can identify an attack source basically for flooding attacks (which involves a huge number of packets), a DPM method can work for the attacks including a small number of packets.[119] This scheme also reduces the storage and computation cost incurred by the PPM schemes in the victims site. The first DPM scheme is proposed by Belenky and Ansari.[120] The basic idea is to embed the IP address of the initial router into the packet. In their method, the authors have considered the interfaces as their traceback atomic units. Thus, it can separate the incoming and outgoing packets and thus can treat the packets differently for different directions. It also improves the cases of mark spoofing by ensuring marking of all packets traveling through a network. Thus, a spoofed mark gets overwritten by a correct mark. This initial method cannot identify multiple attackers. In order to solve this problem, later the authors have proposed a method to track multiple attackers.[121] Here, they have introduced a hash function which produces hash values for the ingress addresses. This hash value is used by the victim to identify correct mark information. In the work by Jin and Yang,[122] a method to improve the performance of the DPM method is proposed. Here, the marking field is divided into two sections: information and index. The information section comprises the address fragment or correlation function value. The authors have introduced effective techniques to store data in the information section. Xiang et al.[123] have proposed a flexible DPM scheme. The "flexible mark length" approach has made it appropriate for different network environments. It also comprises an overload prevention scheme for tracebacking IP addresses. The performance of the method is found to be better than the other DPM methods. Yu et al.[124,125] have proposed a DPM-based marking on demand (MOD) scheme which improves the scalability of the existing DPM methods. This method dynamically marks the traceback encodings and can detect all attack sources. However, the MOD server they have used in their method can be a target of the DDoS attack. Also, as the MOD database includes a large set of attack information, the performance to retrieve information from this database becomes a challenge for the users.

ICMP traceback message. In this method, an ICMP message (iTrace Packet) is generated in each router and send to the receiver which helps to traceback to the attack source.[126] In order to reduce the burden to the receiver, an adjustable low probability (not greater than 1/1000) traceback message is send to the receiver. Thus, this simple and easy method can help the receiver to traceback to the attack source. However, this method introduces an extra traffic and need of digital signature to ensure integrity of the traceback message. Lee et al.[127] have proposed an enhancement to the iTrace named (iTrace-CP) which carries the information of the entire path in its ICMP message. Thus, if the victim can identify an attack packet, he can construct the whole attack path/graph. However, this method requires some change in the router for the processing and also requires further consideration for the storage space in the packet. In order to improve the accuracy of the detection of the attack source when a legitimate user also uses the same path that the attacker has used, a method is proposed in the work of Izaddoost et al.[128] by modifying the intention-driven iTrace model.[129] Saurabh and Sairam[130] have proposed an enhanced version of the traceback methods. This method mainly works for the reflector attack and is capable of identifying the edge router directly that allows the transfer of the attack packets. This method also uses the powerful bloom filters to reduce the number of generated traceback messages. This highly scalable method can also detect the attack at an early stage. In the work by Yao et al.,[131] a method named passive IP traceback (PIT) is proposed. This method analyzes ICMP error messages which are actually generated by malicious traffic (spoofed traffic) and also uses the information that are available in public such as topological information to track down the spoofer. Although this method cannot work for all types of attacks or spoofers but it can eliminate the requirement of deployment for locating some of them.

Response mechanism

After the detection of the attack traffic or attack source, it is important to make a rapid response to mitigate the attack. A robust response mechanism can reduce or eliminate the impacts of a DDoS attack. In this section, we are going to identify some research activities in two different well-known response mechanisms: filtering or rate limiting and capability.

Filtering or rate limiting. Filtering or rate limiting is one of the mostly applicable way to respond to DDoS attacks. A system applies these techniques based on the results of the detection mechanisms. In general, if the results of the detection mechanism are found to be partially successful, that is, if it produces large false negatives or cannot identify a precise mark between the legitimate and malicious traffic, it is reasonable to apply rate limiting rather than filtering. On the other hand, if the detection mechanism can successfully distinguish an attack flow, it is more appropriate to filter that malicious traffic. Some common techniques found in the literature for filtering and rate limiting are found in these researches.[132,133]

Capability-based response. The main technique of DDoS that is flooding a victim with unwanted traffic comes from the fact that any sender can send as much traffic as he wants to a receiver. The receiver cannot control how much traffic she would receive. There exists flow control and congestion control mechanisms, but a misbehaving sender does not care about it. Therefore, the techniques involved in capability-based DDoS response work to find out a solution to control such misbehaving sender. Stateless inter flow filter (SIFF)[134] and traffic validation architecture (TVA)[135] are two example methods of this type.

Tolerance

When the results of the detection algorithm come out to be unsuccessful or when it becomes hard to apply a detection process, it is important to find out the alternatives. Tolerance mechanism is the alternative to this.[136] The techniques of tolerance mechanism work with a very little or even no knowledge about the result of the detection. Thus, it is the final stage of defense when the other stages (prevention and detection) fail. Thus, the goal of this mechanism is to provide maximum possible quality of service by minimizing the impacts of the attack. The research found in this field can be classified into congestion policing and fault tolerance.

Congestion policing. The main target of a bandwidth flooding attack is to congest the resource.[45,66] The impact of this type of attacks can be reduced or eliminated by applying congestion policing mechanism. Re-feedback[137] and NetFence[138] are two example mechanisms where congestion policing is applied to defend DDoS attacks.

Fault tolerance. Fault tolerance ensures high availability of the system. In fault tolerance system, the basic idea is to replicate or multiply the resource of a system such as software, hardware. There are some research found in this field which works to identify the applicability of the fault tolerance system and their impacts on DDoS attacks.[139–142]

DDoS attacks to other systems

In the previous sections, our survey mainly focuses on DDoS attacks on traditional systems. In this section, we provide some discussions about DDoS attacks on non-traditional systems such as clouds, smart grids, smart homes, CPSs, and IoT systems.

DDoS attacks on clouds

Cloud or cloud computing is an online platform which provides "pay-as-you-go" or subscription-based access to different services such as storage, software development, audio/video streaming, data analysis, file sharing.[143,144] It is a highly scalable, distributed infrastructure which makes it easier to get the best cloud services from anywhere in the world. It also eliminates the cost of the infrastructure setup and maintenance and it provides auto-scaling (shrinking or expanding the resources on demand) which ensures high speed and productivity. However, this platform could also be the target of all kinds of attacks including data breach, computation breach, flooding attacks, DoS/DDoS attacks, cloud integrity attacks, cross-virtual machine (VM) attacks, timing attacks, VM co-residence attacks, and so on.[145] For example, the difficulty to tackle cloud integrity attacks includes untrustworthy cloud servers and computational complexity of some approaches. Some recent counter approaches for cloud integrity attacks are listed as follows. In the work by Fu et al.,[146] a similarity search method for encrypted documents based on simhash is proposed to find similar encrypted documents stored in clouds. In the work by Xia et al.,[147] to prevent semi-honest cloud servers, image contents are encrypted via a stream cipher to add a watermark and indexed via a locality-sensitive hash. In the work by Fu et al.,[148] a multi-keyword fuzzy ranked search scheme is proposed for keyword-based search over encrypted outsourced data with keyword transformation and keyword weight. In the work by Fu et al.,[149] searchable encryption is based on content-aware search scheme using conceptual graphs with numerical vectors from linear forms and multi-keyword ranked search over encrypted cloud data. In the work by Fu et al.,[150] searchable encryption is based on conceptual graphs over encrypted outsourced data with sentence scoring in text summarization and Tregex to extract the topic sentences from documents to convert into conceptual graphs. In the work by Shen et al.,[151] to counter the semi-trusted due to the cloud provider as the third party, a framework for urban data sharing by exploiting the attribute-based cryptography to support dynamic operations is proposed.

Clouds could also be the target of DDoS attacks. Info Security reported that in the first quarter of 2015, one-third of the DDoS mitigation services were applied to IT services/Cloud/SaaS sectors.[152] Cloud computing provides auto-scaling, pay-as-you-go accounting multi-tenancy service which makes it an attractive choice for the people around the world. But at the same time, these features are also very useful for a successful DDoS attack.[144] In cloud infrastructure, cloud servers run many VMs to ensure services to the users. When an attacker attacks a server, the server may take this situation as a high resource utilization scenario. Then, the auto-scaling feature applies resource allocation, migration, or placement process to solve this overloaded situation. If this process continues, finally, the DDoS attack becomes successful and this attack imposes some direct and indirect effects on the services and revenues. IP spoofing, SYN flooding, buffer overflow, ping of death, and land attack are some example DDoS attacks in the cloud platforms.[153] Defense against DDoS attacks on clouds can also be achieved in three different ways: attack prevention, attack detection, and attack mitigation. Attack prevention methods include challenge response authentication,[154–156] hidden servers or ports,[157,158] resource limit[159,160] techniques where attack detection methods utilize anomaly detection,[161,162] BotCloud detection[163,164] techniques to prevent and detect a DDoS attack. Furthermore, some mitigation techniques are also found, including resource scaling,[165,166] software-defined networking (SDN),[167,168] and so on, to mitigate DDoS attacks on clouds.

DDoS attacks on smart grids

Smart Grid is the evolution of the traditional power grid to a smart infrastructure which incorporates the Information and Communication Technologies (ICT) to the traditional electric power supply systems.[169–171] It is a distributed power source with automated maintenance and operating capabilities which improve reliability and quality of the power as well as enhance the capacity and efficiency of the traditional power systems.[172] It requires a robust communication network for its reliable access, process, and delivery of the information. This delivery becomes damaged or disrupted when a DoS attack gets success in the smart grid, and as indicated in the work by Liu et al.,[173] there are significant amount of attacks in smart grids. Moreover, in smart grid, it is very important to ensure the timing constraints of the transferred messages because this timing constraint is the benchmark to the reliable monitoring and control of the systems. This constraint also imposes a threat to the smart grid since the main impact of the DoS attack is to make the system unavailable to reach and this introduces delay to the transfer of the messages. In Smart Grid, DoS attacks can target different communication layers: network, transport, medium access control (MAC), and physical layers through flooding or jamming in the substations.[171] Also, the intelligent electronic devices (IEDs) or the supervisory control and data acquisition centers can be the target of DoS attacks. Thus, a successful DoS attack can cause severe service degradation as the control and monitoring operations depend on timely message transfer. Attack detection and attack mitigation mechanisms are two main defense methodologies against DoS attacks in smart grid. There are many different attack detection methods that work to detect DoS attacks in smart grid. Signal-based detection[174,175] and packet-based detection[174,176] methods work to detect jamming attacks in the networks. These detection schemes are also applicable for passive detection of the DoS attacks in wireless-based smart grid applications as well as general detection of DoS attacks. Moreover, rate limiting or filtering techniques to the network layer work to mitigate the impacts of DoS attacks in smart grid. Also, the jamming-resilient schemes that have been used in the wireless network can also be used to mitigate the DoS attacks in the wireless communication–dependent smart grid.

DDoS attacks on smart homes

Through the emerging growth of the Internet and technology, our life is becoming smart day by day. As a consequence, a home of today is considered as a smart home if it is connected to a communication network. Using this connection, the resident of the home can control, monitor, and program all the smart home appliances (smart fridge, smart TV, security alarm, etc.) from a remote location. These smart home devices ease the life and increase security of a home owner. But at the same time, these devices may introduce an immense threat if they get attacked by any cyber attack. According to the work by Mantas et al.,[177] the security of a smart home depends on six main properties: confidentiality, integrity, authentication, authorization, non-repudiation, and availability. A DoS/DDoS attack targets availability of the smart devices by a partial or total interruption to the communication system. Auriemma[178] and Nunes[179] have reported that DoS attacks on smart TVs freeze the normal service and operation of the devices. To protect a smart device from the DoS attack, it is important to ensure authentication of the system.[180] In this way, it will be possible to block unauthorized devices and services.

Furthermore, more threatening scenario is the use of smart devices to create the botnets of the DDoS attacks. Smart devices are very vulnerable to become zombies since they are less secure devices. Most of the devices have factory built passwords. Therefore, it is very easy for attackers to take the control of such devices and make them the zombies of the attack army. As we have mentioned before, the attack on 21 October 2016 was mainly done by some "smart devices," specifically the CCTV security cameras.[181] This attack involves tens of millions of IP addresses, most of which are from these smart devices. This attack is the largest (in size) in history.[182] Since these threats are not very old, till now the only way to prevent the smart devices from being a part of the botnet is to increase the security of the devices itself and it can be ensured by the manufacturers or the end users.

DDoS attacks on IoT systems

The IoTs interconnect network entities of different types ranging from a toaster at a home to a heart monitoring device implanted in a human body, a connected automobile, a smart grid infrastructure, or a web cam of a computer. Thus, the entities of the IoTs are called "highly heterogeneous" networked entities.[183] In the work by Strategy and Unit,[184] the International Telecommunication Union (ITU) has mentioned its vision toward the IoT world as the "technology to provide connectivity to anything." DuBravac and Ratti[185] have mentioned that in 2015, the number of connected devices has become 10–20 billion in number, whereas in 2003, there were only 500 million connected devices and in 5 years, the number of connected devices will increase to 40–50 billion in number.[183] This scenario shows the tremendous growth of the IoT world which is good and bad at the same time. This is because, according to Trend Micro, Inc,[186] too many of these IoT devices are not secured at all since the vendors or manufacturers of those devices are not concerned about the security issues of these devices. Moreover, some manufacturers hard coded the less secure credentials of the IoT devices which cannot be changed but they are very easy to guess.[187] As a result, in the last few months of the 2016, the cyberworld has faced some of the largest (in size) DDoS attacks of its history on https://KrebsOnSecurity.com (620 Gbps),[188] OVH attack (990 Gbps),[189] and Dyn DNS (1.2 Tbps).[190] All of these attacks have involved millions of IPs from all of these smart devices. As we have already mentioned in section "Prevention based on awareness," it is very important to ensure device security through the actions from the device manufacturers and end users. On the other hand, as the IoT devices are working in different sophisticated areas such as health care, rescue service, connected vehicles, if these devices face a DoS attack, a catastrophic event may occur if the devices deny to communicate and provide services in critical situations. So far authentication is the solution to the DoS attacks in such devices. Thus, it is very important to put more attentions to this massively growing field of the Internet.

DDoS attacks on CPSs

According to the work by Baheti and Gill,[191] CPS refers to a new generation of systems with integrated computational and physical capabilities. Thus, it is a future system that will work as a bridge between the cyber and the physical worlds of computing. The main idea is to introduce enhanced control, monitoring, and coordination in different critical systems such as aerospace, health care, urban automobile.[192] As this system is also dependent on trusted communication, a DoS attack may incur significant threats to the availability and service of the system. This is an ongoing research issue to consider for a robust CPS in general.

Discussion

There are many researches fighting against DDoS attacks. However, still, the deployment or development of effective methods and mechanisms of these researches cannot resist DDoS attacks in present days. Rather, it is increasing in its frequency and size. Thus, it is also important to identify why these researches to defend DDoS attacks are not becoming successful in preventing and discouraging these threats.

A number of statements can be made as an answer to this question. The most important one is the lack of distributed cooperation among many points in the network. It is because of the distributed administration of the Internet which cannot enforce global cooperation. Also some socioeconomic factors are involved and they make it difficult to deploy the methods globally. As DDoS attacks are distributed attacks, a single point deployment cannot ensure best defense against the attacks.

Throughout our analysis, we outline some other challenges where further researches are needed in future. First, the increasing growth of the Internet and the availability of the insecure IoT devices are a very big threat of the current cyberworld. All recent major DDoS attacks are based on IoT botnets. The users of these enormous systems are not aware of the security of their devices. However, the main prevention so far against the creation of this huge botnet is to ensure device security from user's side. Ensuring defense against the IoT-based DDoS attacks is an extremely important field of research where there are many different unsolved issues that require special attention. Preventing the creation of the IoT botnets, detection,

Table 1. Summary of DDoS attacks.

DDoS attack	Type of attack	IP spoofing	Attack features	Attack layer	Impact
TCP SYN attack	Protocol exploiting resource depletion	Spoofed	Exploits TCP's three-way handshaking	Transport layer	Consumes server's resources
TCP PUSH + ACK	Protocol exploiting resource depletion	Spoofed and non-spoofed	Sets 1 to the PUSH and ACK field of a TCP header	Transport layer	Floods victim's memory and CPU
HTTP flood	Protocol exploiting resource depletion	Non-spoofed	Exploits HTTP GET and HTTP POST request	Application layer	Consumes server's all resources
SIP flood	Protocol exploiting resource depletion	Non-spoofed	Exploits HTTP GET and HTTP POST request	Application layer	Consumes server's all resources
Slowlories	Protocol exploiting resource depletion	Non-spoofed	Keeps HTTP connection open as long as possible	Application layer	Consumes all sockets
HTTP fragmentation	Protocol exploiting resource depletion	Non-spoofed	Fragments a HTTP packet into smaller chunks and send those at minimum possible rates	Application layer	Consumes all sockets
R.U.D.Y	Protocol exploiting resource depletion	Non-spoofed	Exploits form submission field by sending form information in minimum possible packet size and rate	Application layer	Consumes all connections in the connection table
Slow read	Protocol exploiting resource depletion	Non-spoofed	Reads response at the slowest possible rate	Application layer	Consumes all connections in the connection table
Land attack	Resource depletion using malformed packets	Spoofed	Sets source and destination IP to victim's IP	Network layer	Creates infinite loop for the target which crashes victim's system
IP packet option field	Resource depletion using malformed packets	Spoofed	Sets 1 to all quality of service bits	Network layer	Inundates processing ability of the victim
Ping of death	Resource depletion using malformed packets	Spoofed	Forms a data packet that exceeds maximum packet size	Network layer	Causes buffer overflow and system crash
Teardrop attack	Resource depletion using malformed packets	Spoofed	Sends fragmented packets with overlapping offset numbers	Network layer	Generates error in fragmentation and reassembly of packets
UDP flood	Protocol exploited bandwidth depletion attack	Spoofed	Sends a large stream of UDP packets to a specific or random port of a target	Transport layer	Consumes network bandwidth
ICMP flood	Protocol exploited bandwidth depletion attack	Spoofed	Exploits ICMP's ECHO_REQUEST packet	IP layer	Saturates victim's network bandwidth
Fraggle	Protocol exploited bandwidth depletion attack	Spoofed	Sends UDP_ECHO packets to the network amplifier	IP layer	Saturates victim's network bandwidth
DNS amplification	Amplification based bandwidth depletion attack	Spoofed	Exploits and amplifies DNS response message	Application layer	Saturates victim's network bandwidth
NTP amplification	Amplification based bandwidth depletion attack	Spoofed	Exploits NTP using MON_GETLIST command	Application layer	Saturates victim's network bandwidth
DNS flooding	Flooding based infrastructure attack	Spoofed	Exploits and amplifies DNS response message	Application layer	Saturates victim's network bandwidth

DDoS: distributed denial of service; IP: Internet Protocol; UDP: User Datagram Protocol; HTTP: Hypertext Transfer Protocol; ICMP: Internet Control Message Protocol; DNS: domain name system; NTP: Network Time Protocol.

Table 2. Summary of DDoS prevention using filters.

Method	Point of action	Key features	Advantages	Limitations
Ingress/egress filtering[68]	Edge router of the victim's network	Filters traffic based on a predefined range of domain prefix	1. Filters spoofed IP packets 2. Easy to deploy	1. Not effective for the spoofed IP from valid address range 2. Challenging to obtain range of expected IP for complex topology 3. Tunneling required for mobile IP users
Martian address filtering[69]	All inbound routers	Prevents delivery of packets with IPs from the reserved or special IP addresses as well as the unallocated range of IP addresses	Filters spoofed IP packets	1. Not effective for the spoofed IP from valid address range 2. Works well for only a limited address range
Source address validation[69]	All inbound routers	Prevents delivery of packets based on the interface match in source and destination route	Filters spoofed IP packets	1. False-positive rate is high for the asymmetric route of the Internet
Route-based packet filtering[70]	Core routers	1. Filters packet based on route information 2. Require BGP routing topology	1. Filters spoofed IP packets 2. Successful if 18% of the autonomous system (AS) implements the technique	1. Needs modification in BGP messaging 2. Needs router information update 3. Possible to deceive root information by stealing BGP session
SAVE[72]	Core routers	1. Improvement to RPF 2. Enforces the routers to send messages containing updated source information to each destination routers connected to a source	1. Filters spoofed IP packets 2. Overcomes the problem of RPF	1. Needs modification in existing routing protocol 2. Partial deployment does not ensure full success
Hop-count filtering[73]	Routers in victim's site	1. Works based on the hop counts of a packet 2. TTL value is used to count the hops	1. Filters spoofed IP packets 2. A simple and lightweight technique which requires low storage facility	1. Larger false positive for different TTL values in different operating systems 2. Attacker can forge valid hop count
History-based filtering[74]	Routers in victim's site	1. Works based on the history of the normal traffic	Works well for bandwidth attack	Not found to be successful when attacker simulates its traffic as a normal traffic
Path identifier–based filtering[75,76]	Routers in victim's site	1. Works based on the identified path of the attacker 2. Marks the packets and identifies attack path based on deterministic packet marking approach	1. Can filter subsequent attack traffic after detection of a single attack traffic	1. Larger false positive for small identification field 2. Needs a large number of routers involvement
Packet-score[78,79]	Routers in victim's site	1. A statistical method that assigns each packet a score based on some profile value analysis 2. Uses Bayes' theorem to calculate CLP	1. Filters non-spoofed packets and attack packets with new signature	1. Larger storage requirement 2. Needs an initial profile information which is somewhat difficult to achieve

IP: Internet Protocol; RPF: Route-based packet filtering; TTL: time to live; CLP: conditional legitimate probability.

Table 3. Summary of DDoS attack detection methods.

Method	Research papers covered	Key features	Advantages	Limitations
Signature-based detection	88–93	1. Works based on already defined attack signatures 2. Uses pattern matching to detect attacks	1. Low false-positive rate and faster attack detection for known attacks 2. Can work for the system just after its installation	1. Difficult to manage state information where attack spans on multiple packets 2. Only works for known attack signatures so frequent update is required
Anomaly-based detection	96–108	1. Creates a baseline profile for the initial and normal system 2. Any deviation from the baseline profile is considered as anomaly 3. Updates profile based on new signature, behavior, and critical events such as false alarms	1. Can detect insider attack 2. Its way to work based on personal profile makes it difficult for the attacker to identify the “normal” behavior 3. Can detect unknown attacks	1. Require a training phase in the “normal” situation 2. Get affected by false alarms 3. It is possible to train a system by the attacker to accept him as a normal user
Probabilistic packet marking–based attack source detection	109–116,118	1. Probabilistically encodes information into packet header 2. Uses identification field to store the marking 3. Reconstruct attack path based on marked information	1. Does not require any extra traffic or communication mechanism 2. Can reconstruct attack path without ISP cooperation	1. High false-positive rate 2. Require large number of packets
Deterministic packet marking–based attack source detection	119–125	1. Deterministically marks packets in the ingress routers near the source of the attack 2. Uses 16-bit identification field and one bit reserved flag field to store the marking information 3. Reconstruct attack path based on marked information	1. Reduces storage and computation overhead of PPM 2. Can work with small number of packets	1. High false-positive rate 2. It can traceback to the nearest ingress router to the attacker but not to the exact attack source
ICMP message–based attack source detection	126–131	1. An ICMP message (iTrace) is generated in each router and send to the victim 2. Partial path information are contained in the packets 3. Combination of these information helps to traceback to the source	1. Simple and compatible to existing protocols 2. Possible to apply incremental deployment 3. ISP cooperation is not required	1. Produces extra traffic 2. Authentication is required for the message

ISP: Internet service provider; PPM: probabilistic packet marking; ICMP: Internet Control Message Protocol.

and rejection of the flows from unsophisticated IoT devices (such as security camera, smart refrigerator, home routers) are some example issues where more researches are required.

Second, it is also challenging to maintain a trade-off between the performance of the online (real time) defense techniques and consumption of the victim's resources. Since DDoS attacks already put immense pressure on the resources (processing power, memory, bandwidth) of the victim's system, it is very important to ensure best performance of the DDoS defense methods. That is, it is important to ensure lowest possible consumption of the victim's resources by the defense mechanisms while fighting against DDoS. This is an extremely important research direction since the best performed defense mechanisms assure the minimum downtime as well as the minimum revenue loss from the victim's side.

Third, scalability is another challenge found in DDoS research. DDoS attacks involve diverse attack scenario and signatures. Therefore, the researchers are trying to provide the best possible defense considering different aspects of the attacks. It is very important to test the real-life performance of those researches. Also, it is essential to test the performance in real time when different methods collaborate with each other as different methods work on to solve the problem from different directions. The real-life attack scenario is very different from the testing environment which works based on some fixed dataset and fixed attack signatures. Therefore, it is important to ensure scalability of the defense mechanisms in real-time attack scenarios.

Fourth, ensuring defense against the zero-day attack is always an open research problem. The DDoS attackers are always working on to introduce new types of attacks with increased power and complexity. Therefore, the research to defend against zero-day attack is the most challenging one. Besides all the major technical skills, this research also requires understanding attackers' psychology and skills which bring out new types of DDoS attacks.

We summarize different types of attacks, filtering techniques, and attack detection methods in Tables 1–3, respectively. In those summaries, we identify the key features of the attacks as well as advantages and disadvantages of different defense mechanisms.

Another aspect to improve the detection of DDoS attacks is to include more accountability functions and auditing functions in the Internet. In the work by Xiao,[193] an accountable logging method, called flow-net, was introduced for better logging for computer and network systems. However, this approach requires all routers to implement the flow-net method and it is unlikely to achieve in reality. In the work by Shen et al.,[194] a public auditing protocol using a doubly linked info table and a location array with global verification to conduct both single and batch auditing was proposed. However, if applying this approach to the Internet, very expensive cryptograph functions will cause the systems slow down.

Conclusion

In this survey, we have presented a comprehensive and systematic analysis of the DDoS attacks. We have enlisted different attack types seen so far. In our work, we have analyzed well-known prevention and mitigation techniques based on their success and failures. We have summarized different types of attacks, filtering techniques, and attack detection methods. We have identified the key features of the attacks as well as advantages and disadvantages of different defense mechanisms. However, still there exists the chance to see new unseen attacks with new signatures and features. However, this survey will work as an easy to understand foundation of the DDoS attacks for its systematic explanation and analysis. As this survey has also included recent attacks and recent research against DDoS attacks, it also presents the current state of the art of the DDoS attacks. We also provided some discussions about DDoS attacks on non-traditional systems such as clouds, smart grids, smart homes, CPSs, and IoT systems. Finally, we have also enlisted the challenges involved to the research of the DDoS attacks. Thus, it outlines some extremely important future research directions deserving attentions.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: Project is partially supported by the open Foundation for University Innovation Platform from Hunan Province, China (grant no. 16K013).

References

1. York K. Dyn statement on 10/21/2016 DDoS attack, 2017, http://dyn.com/blog/dyn-statement-on-10212016-ddos-attack/ (accessed 10 March 2017).
2. Waterman S. DDoS attacks growing faster in size, complexity—arbor report, 2017, http://edscoop.com/ddos-attacks-growing-faster-in-size-complexity-arbor-report (accessed 10 March 2017).
3. Liu B. High performance simulation technology in the internet of things. *Int J Sens Netw* 2015; 17(3): 195–202.
4. Peng T, Leckie C and Ramamohanarao K. Survey of network-based defense mechanisms countering the DoS and DDoS problems. *ACM Comput Surv* 2007; 39(1): 3.

5. Guo Y and Perreau S. Detect DDoS flooding attacks in mobile ad hoc networks. *Int J Secur Network* 2010; 5(4): 259–269.
6. Deng J, Han R and Mishra S. Limiting DoS attacks during multihop data delivery in wireless sensor networks. *Int J Secur Network* 2006; 1(3–4): 167–178.
7. Criscuolo PJ. Distributed denial of service: Trin00, tribe flood network, tribe flood network 2000, and Stacheldraht CIAC-2319. Technical report, DTIC Document, 2000, https://e-reports-ext.llnl.gov/pdf/237595.pdf
8. Malliga S and Tamilarasi A. A backpressure technique for filtering spoofed traffic at upstream routers. *Int J Secur Network* 2010; 5(1): 3–14.
9. Mirkovic J, Dietrich S, Dittrich D, et al. *Internet Denial of service: attack and defense mechanisms* (Radia Perlman Computer Networking and Security). Upper Saddle River, NJ: Prentice Hall PTR, 2004.
10. Yue S, Xiao Y and Xie G. Fault tolerance experiments in 4D future internet architecture. *J Internet Technol* 2010; 11(4): 543–552.
11. Gao J and Xiao Y. ProtoGENI DoS/DDoS security tests and experiments. In: *Proceedings of the 1st GENI research and educational experiment workshop (GREE12), in conjunction with GENI GEC 13*, Los Angeles, CA, 13–15 March 2012.
12. Gao J, Xiao Y, Rao S, et al. Security tests and attack experimentations of protoGENI. *Int J Secur Network* 2015; 10(3): 151–169.
13. Nazario J. DDoS attack evolution. *Netw Secur* 2008; 2008(7): 7–10.
14. Namestnikov Y. DDoS attacks in Q2 2011, 2011, https://securelist.com/analysis/quarterly-malware-reports/36394/ddos-attacks-in-q2-2011/ (accessed 10 March 2017).
15. Zargar ST, Joshi J and Tipper D. A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Commun Surv Tut* 2013; 15(4): 2046–2069.
16. Nazario J. Politically motivated denial of service attacks. In: Czosseck C and Geers K (eds) *The virtual battlefield: perspectives on cyber warfare*. Amsterdam: IOS Press, 2009, pp.163–181.
17. Wikipedia. Internet activism during the 2009 Iranian election protests—Wikipedia, the free encyclopedia, 2017, https://en.wikipedia.org/w/index.php?title=Internet_activism_during_the_2009_Iranian_election_protests&oldid=769078109 (accessed 10 March 2017).
18. Schonfeld E. WikiLeaks reports it is under a denial of service attack, 2017, https://techcrunch.com/2010/11/28/wikileaks-ddos-attack/ (accessed 10 March 2017).
19. Liu J, Xiao Y, Ghaboosi K, et al. Botnet: classification, attacks, detection, tracing, and preventive measures. *EURASIP J Wirel Comm* 2009; 2009: 692654-1–692654-11.
20. Douligeris C and Mitrokotsa A. DDoS attacks and defense mechanisms: classification and state-of-the-art. *Comput Netw* 2004; 44(5): 643–666.
21. Sun B, Yan G, Xiao Y, et al. Self-propagating mal-packets in wireless sensor networks: dynamics and defense implications. *Ad Hoc Netw* 2009; 7: 1489–1500.
22. Staniford S, Paxson V, Weaver N, et al. How to own the Internet in your spare time. In: *Proceedings of the 11th USENIX security symposium*, San Francisco, CA, 5–9 August 2002, vol. 2. pp.14–15. Berkeley, CA: USENIX Association.
23. Mirkovic J and Reiher P. A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Comp Com* 2004; 34(2): 39–53.
24. Patrikakis C, Masikos M and Zouraraki O. Distributed denial of service attacks. *Internet Protocol J* 2004; 7(4): 13–35.
25. Weaver N. *Potential strategies for high speed active worms: a worst case analysis* (Whitepaper). Berkeley, CA: UC Berkeley, 2002.
26. Weaver NC. Warhol worms: the potential for very fast internet plagues, 2001, http://www.iwar.org.uk/comsec/resources/worms/warhol-worm.htm (accessed 1 April 2017).
27. Moore D, Shannon C and Claffy K. Code-Red: a case study on the spread and victims of an internet worm. In: *Proceedings of the 2nd ACM SIGCOMM workshop on Internet measurement*, Marseille, 6–8 November 2002, pp.273–284. New York: ACM.
28. Netcraft. Web server survey, 2017, https://news.netcraft.com/archives/category/web-server-survey/ (accessed 1 April 2017).
29. Chen Z and Ji C. Optimal worm-scanning method using vulnerable-host distributions. *Int J Secur Network* 2007; 2: 71–80.
30. Chen Z and Chen C. Characterising heterogeneity in vulnerable hosts on worm propagation. *Int J Secur Network* 2016; 11: 224–234.
31. Long N and Thomas R. *Trends in denial of service attack technology*. Pittsburgh, PA: CERT Coordination Center, 2001.
32. Mansfield-Devine S. DDoS: threats and mitigation. *Netw Secur* 2011; 2011(12): 5–12.
33. Specht SM and Lee RB. Distributed denial of service: taxonomies of attacks, tools, and countermeasures. In: *Proceedings of the ISCA 17th international conference on parallel and distributed computing systems (PDCS 2004), international workshop on security in parallel and distributed systems*, San Francisco, CA, 15–17 September 2004, pp.543–550. IEEE.
34. Srivastava A, Gupta B, Tyagi A, et al. A recent survey on DDoS attacks and defense mechanisms. In: Nagamalai D, Renault E and Dhanuskodi M (eds) *Advances in parallel distributed computing*. Berlin; Heidelberg: Springer, 2011, pp.570–580.
35. SecurityIQ. Cheating VoIP security by flooding the SIP, 2016, http://resources.infosecinstitute.com/cheating-voip-security-by-flooding-the-sip/#gref (accessed 13 April 2017).
36. VERISIGN. Security services: HTTP flood attack, 2017, https://www.verisign.com/en_US/security-services/ddos-protection/ddos-attack/index.xhtml (accessed 13 April 2017).
37. Patrick Park NW. Call flooding attack, 2009, http://www.networkworld.com/article/2234402/cisco-subnet/call-flooding-attack.html (accessed 13 April 2017).
38. RSnake JK and Lee R. Slowloris HTTP DoS, 2009, https://gist.github.com/steakknife/1865841 (accessed June 2009).
39. RioRey. Taxonomy of DDoS attacks, 2015, https://static1.squarespace.com/static/5548bab5e4b08ecb6652391c/

t/554923d6e4b03d6d83ac7c5e/1430856662717/RioRey_Taxonomy_DDoS_Attacks_2.6_2015-1.pdf (accessed 19 May 2017).

40. Incapsula DPCI. R.U.D.Y. (R-U-Dead-Yet?), 2015, https://www.incapsula.com/ddos/attack-glossary/rudy-r-u-dead-yet.html (accessed 19 May 2017).
41. Park J, Iwai K, Tanaka H, et al. Analysis of slow read DoS attack. In: *Proceedings of the 2014 international symposium on information theory and its applications (ISITA)*, Melbourne, VIC, Australia, 26–29 October 2014, pp.60–64. New York: IEEE.
42. Shekyan S. Are you ready for slow reading? Qualys blog, 2012, https://community.qualys.com/blogs/securitylabs/2012/01/05/slow-read (accessed 12 December 2014).
43. Shekyan S. How to protect against slow HTTP attacks, 2011, https://blog.qualys.com/securitylabs/2011/11/02/how-to-protect-against-slow-http-attacks (accessed 30 May 2017).
44. EUROCON. DoS attacks and countermeasures, 2017, https://sites.google.com/a/pccare.vn/it/security-pages/dos-attacks-and-countermeasures (accessed 13 April 2017).
45. Deng J, Meng K, Xiao Y, et al. Implementation of dos attack and mitigation strategies in IEEEE 802.11b/g WLAN. In: *Proceedings of SPIE defense security and sensing 2010*, Orlando, FL, 5–9 April 2010. Bellingham, WA: SPIE.
46. Du X, Guizani M, Xiao Y, et al. Defending DoS attacks on broadcast authentication in wireless sensor networks. In: *Proceedings of the IEEE international conference on communications 2008 (ICC '08)*, Beijing, China, 19–23 May 2008, pp.1653–1657. New York: IEEE.
47. Radware. DDoS attack definitions—DDoSPedia, 2017, https://security.radware.com/ddos-knowledge-center/ddospedia/fraggle-attack/ (accessed 19 April 2017).
48. Base IK. What is a DNS amplification attack? 2013, https://deepthought.isc.org/article/AA-00897/0/What-is-a-DNS-Amplification-Attack.html (accessed 19 April 2017).
49. Incapsula I. DNS amplification, 2017, https://www.incapsula.com/ddos/attack-glossary/dns-amplification.html (accessed 19 April 2017).
50. US-CERT. DNS amplification attacks, 2017, https://www.us-cert.gov/ncas/alerts/TA13-088A (accessed 20 April 2017).
51. Rosenbaum R. Using the domain name system to store arbitrary string attributes, 1993, https://datatracker.ietf.org/doc/rfc1464/
52. US-CERT. Alert (ta13-088a): DNS amplification attacks, 2013, https://www.us-cert.gov/ncas/alerts/TA13-088A (accessed 10 March 2017).
53. Graham-Cumming J. *Understanding and mitigating NTP-based DDoS attacks*, vol. 9. San Francisco, CA: Cloudflare, Inc., 2014.
54. Wikipedia. Network time protocol—Wikipedia, the free encyclopedia, 2017, https://en.wikipedia.org/w/index.php?title = Network_Time_Protocol&oldid = 782576449 (accessed 30 May 2017).
55. Wikipedia. Distributed denial-of-service attacks on root nameservers—Wikipedia, the free encyclopedia, 2017, https://en.wikipedia.org/w/index.php?title = Distributed_denial-of-service_attacks_on_root_nameservers (accessed 21 April 2017).
56. Dyn O. DNS: DNS products trusted by the world's most admired digital brands, 2017, http://dyn.com/dns/ (accessed 20 April 2017).
57. Limer E. How hackers wrecked the internet using DVRs and webcams, 2016, http://www.popularmechanics.com/technology/infrastructure/a23504/mirai-botnet-internet-of-things-ddos-attack/ (accessed 20 April 2017).
58. Statistica. Internet of Things (IoT): number of connected devices worldwide from 2012 to 2020, 2017, https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/ (accessed 20 April 2017).
59. Herzberg B, Bekerman D and Zeifman I. Breaking down Mirai: an IoT DDoS Botnet analysis, 2017, https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html (accessed 20 April 2017).
60. Radware DDoS Knowledge Center. Zero-day—zero-minute attack, 2017, https://security.radware.com/ddos-knowledge-center/ddospedia/zero-day-zero-minute-attack/ (accessed 30 May 2017).
61. Khalimonenko A, Strohschneider J and Kupreev O. Quarterly malware reports: DDoS attacks in Q4 2016, 2017, https://securelist.com/analysis/quarterly-malware-reports/77412/ddos-attacks-in-q4-2016/ (accessed 3 May 2017).
62. Akamai. Akamais [state of the internet]/security Q4 2016 executive summary, 2017, https://www.akamai.com/us/en/multimedia/documents/state-of-the-internet/q4-2016-state-of-the-internet-security-executive-summary.pdf (accessed 3 May 2017).
63. Ponemon Institute. Cyber security on the offense: a study of it security experts, 2012, https://security.radware.com/uploadedFiles/Resources_and_Content/Attack_Tools/CyberSecurityontheOffense.pdf (accessed 3 May 2017).
64. Abliz M. *Internet denial of service: attacks and defense mechanisms*. Technical report no. TR-11-178, March 2011, pp.1–50. Pittsburgh, PA: Department of Computer Science, University of Pittsburgh.
65. Kalkan K, Gür G and Alagöz F. Filtering-based defense mechanisms against DDoS attacks: a survey. *IEEE Syst J*. Epub ahead of print 27 September 2016. DOI: 10.1109/JSYST.2016.2602848.
66. Kolesnikov V and Lee W. MAC aggregation resilient to DoS attacks. *Int J Secur Network* 2012; 7: 122–132.
67. Mölsä J. Mitigating denial of service attacks: a tutorial. *J Comput Secur* 2005; 13(6): 807–837.
68. Senie D and Ferguson P. Network ingress filtering: defeating denial of service attacks which employ IP source address spoofing. *Network*, 1998, https://dl.acm.org/citation.cfm?id = RFC2267
69. Baker F. Requirements for IP version 4 routers, 1995, https://datatracker.ietf.org/doc/rfc1812/
70. Park K and Lee H. On the effectiveness of route-based packet filtering for distributed DoS attack prevention in power-law internets. *ACM SIGCOMM Comp Com* 2001; 31(4): 15–26.
71. Duan Z, Yuan X and Chandrashekar J. Controlling IP spoofing through interdomain packet filters. *IEEE T Depend Secure* 2008; 5(1): 22–36.

72. Li J, Mirkovic J, Wang M, et al. SAVE: source address validity enforcement protocol. In: *Proceedings of the 21st annual joint conference of the IEEE computer and communications societies (INFOCOM 2002)*, New York, 23–27 June 2002, vol. 3, pp.1557–1566. New York: IEEE.
73. Jin C, Wang H and Shin KG. Hop-count filtering: an effective defense against spoofed DDoS traffic. In: *Proceedings of the 10th ACM conference on computer and communications security*, Washington, DC, 27–30 October 2003, pp.30–41. New York: ACM.
74. Peng T, Leckie C and Ramamohanarao K. Protection from distributed denial of service attacks using history-based IP filtering. In: *Proceedings of the IEEE international conference on communications, 2003 (ICC '03)*, Anchorage, AK, 11–15 May 2003, vol. 1, pp.482–486. New York: IEEE.
75. Yaar A, Perrig A and Song D. Pi: a path identification mechanism to defend against DDoS attacks. In: *Proceedings of the 2003 symposium on security and privacy*, Berkeley, CA, 11–14 May 2003, pp.93–107. New York: IEEE.
76. Yaar A, Perrig A and Song D. StackPi: new packet marking and filtering mechanisms for DDoS and IP spoofing defense. *IEEE J Sel Area Comm* 2006; 24(10): 1853–1863.
77. Beverly R, Berger A, Hyun Y, et al. Understanding the efficacy of deployed internet source address validation filtering. In: *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*, Chicago, IL, 4–6 November 2009, pp.356–369. New York: ACM.
78. Kim Y, Lau WC, Chuah MC, et al. PacketScore: a statistics-based packet filtering scheme against distributed denial-of-service attacks. *IEEE T Depend Secure* 2006; 3(2): 141–155.
79. Kalkan K and Alagöz F. A distributed filtering mechanism against DDoS attacks: ScoreForCore. *Comput Netw* 2016; 108: 199–209.
80. Adkins D, Lakshminarayanan K, Perrig A, et al. *Towards a more functional and secure network infrastructure*. Technical report no. UCB/CSD-03-1242, 2003. EECS Department, University of California, Berkeley, http://www2.eecs.berkeley.edu/Pubs/TechRpts/2003/6241 .html
81. Keromytis AD, Misra V and Rubenstein D. SOS: secure overlay services. *ACM SIGCOMM Comp Com* 2002; 32(4): 61–72.
82. Krämer L, Krupp J, Makita D, et al. AmpPot: monitoring and defending against amplification DDoS attacks. In: *Proceedings of the international workshop on recent advances in intrusion detection*, Kyoto, Japan, 2–4 November 2015, pp.615–636. Cham: Springer.
83. Weiler N. Honeypots for distributed denial-of-service attacks. In: *Proceedings of the 11th IEEE international workshops on enabling technologies: infrastructure for collaborative enterprises, 2002 (WET ICE 2002)*, Pittsburgh, PA, 12 June 2002, pp.109–114. New York: IEEE.
84. McMullin M. DNS, load balancing and DDoS attacks, 2016, https://kemptechnologies.com/blog/load-balancing-and-ddos-attacks/ (accessed 5 May 2017).
85. Geng X and Whinston AB. Defeating distributed denial of service attacks. *IT Prof* 2000; 2(4): 36–42.
86. Liu J, Xiao Y and Chen CLP. Internet of things' authentication and access control. *Int J Secur Network* 2012; 7(4): 228–241.
87. Sun B, Osborne L, Xiao Y, et al. Intrusion detection techniques in mobile ad hoc and wireless sensor networks. *IEEE Wirel Commun* 2007; 14: 56–63.
88. Paxson V. Bro: a system for detecting network intruders in real-time. *Comput Netw* 1999; 31(23): 2435–2463.
89. Cabrera JB, Lewis L, Qin X, et al. Proactive detection of distributed denial of service attacks using MIB traffic variables—a feasibility study. In: *Proceedings of the 2001 IEEE/IFIP international symposium on integrated network management*, Seattle, WA, 14–18 May 2001, pp.609–622. New York: IEEE.
90. Roesch M. Snort: lightweight intrusion detection for networks. In: *Proceedings of the 13th USENIX conference on system administration*, Seattle, WA, 7–12 November 1999. Berkeley, CA: USENIX Association.
91. Cheng CM, Kung H and Tan KS. Use of spectral analysis in defense against DoS attacks. In: *Proceedings of the IEEE global telecommunications conference, 2002 (GLOBECOM'02)*, Taipei, Taiwan, 17–21 November 2002, vol. 3, pp.2143–2148. New York: IEEE.
92. Hussain A, Heidemann J and Papadopoulos C. A framework for classifying denial of service attacks. In: *Proceedings of the 2003 conference on applications, technologies, architectures, and protocols for computer communications*, Karlsruhe, 25–29 August 2003, pp.99–110. New York: ACM.
93. Hussain A, Heidemann J and Papadopoulos C. Identification of repeated denial of service attacks. In: *Proceedings of the 25th IEEE international conference on computer communications (INFOCOM 2006)*, Barcelona, 23–29 April 2006, pp.1–15. New York: IEEE.
94. Jow J, Xiao Y and Han W. A survey of intrusion detection systems in smart grid. *Int J Sens Netw* 2017; 23(3): 170–186.
95. Sun B, Xiao Y and Wang R. Detection of fraudulent usage in wireless networks. *IEEE T Veh Technol* 2007; 56(6): 3912–3923.
96. Gil TM and Poletto M. MULTOPS: a data-structure for bandwidth attack detection. In: *Proceedings of the 10th conference on USENIX security symposium*, Washington, DC, 13–17 August 2001, pp.23–38. Berkeley, CA: USENIX Association.
97. Abdelsayed S, Glimsholt D, Leckie C, et al. An efficient filter for denial-of-service bandwidth attacks. In: *Proceedings of the IEEE global telecommunications conference, 2003 (GLOBECOM'03)*, San Francisco, CA, 1–5 December 2003, vol. 3, pp.1353–1357. New York: IEEE.
98. Mirkovic J, Prier G and Reiher P. Attacking DDoS at the source. In: *Proceedings of the 10th IEEE international conference on network protocols*, Paris, 12–15 November 2002, pp.312–321. New York: IEEE.
99. Barford P, Kline J, Plonka D, et al. A signal analysis of network traffic anomalies. In: *Proceedings of the 2nd ACM SIGCOMM workshop on Internet measurement*, Marseille, 6–8 November 2002, pp.71–82. New York: ACM.

100. Huang Y and Pullen JM. Countering denial-of-service attacks using congestion triggered packet sampling and filtering. In: *Proceedings of the 10th international conference on computer communications and networks*, Scottsdale, AZ, 15–17 October 2001, pp.490–494. New York: IEEE.

101. Lee W and Stolfo SJ. Data mining approaches for intrusion detection. In: *Proceedings of the 7th conference on USENIX security symposium*, San Antonio, TX, 26–29 January 1998. Berkeley, CA: USENIX Association.

102. Talpade R, Kim G and Khurana S. NOMAD: traffic-based network monitoring framework for anomaly detection. In: *Proceedings of the IEEE international symposium on IEEE computers and communications*, Red Sea, Egypt, 6–8 July 1999, pp.442–451. New York: IEEE.

103. Xie Y and Yu SZ. A large-scale hidden semi-Markov model for anomaly detection on user browsing behaviors. *IEEE ACM T Network* 2009; 17(1): 54–65.

104. Xiang Y, Li K and Zhou W. Low-rate DDoS attacks detection and traceback by using new information metrics. *IEEE T Inf Foren Sec* 2011; 6(2): 426–437.

105. Jiang D, Xu Z, Zhang P, et al. A transform domain-based anomaly detection approach to network-wide traffic. *J Netw Comput Appl* 2014; 40: 292–306.

106. Ma X and Chen Y. DDoS detection method based on chaos analysis of network traffic entropy. *IEEE Commun Lett* 2014; 18(1): 114–117.

107. Bhuyan MH, Bhattacharyya D and Kalita JK. E-LDAT: a lightweight system for DDoS flooding attack detection and IP traceback using extended entropy metric. *Secur Commun Netw* 2016; 9(16): 3251–3270.

108. Jyothi V, Wang X, Addepalli SK, et al. BRAIN: behavior based adaptive intrusion detection in networks: using hardware performance counters to detect DDoS attacks. In: *Proceedings of the 2016 29th international conference on VLSI design and 2016 15th international conference on embedded systems (VLSID)*, Kolkata, India, 4–8 January 2016, pp.587–588. New York: IEEE.

109. Savage S, Wetherall D, Karlin A, et al. Practical network support for IP traceback. *ACM SIGCOMM Comp Com* 2000; 30(4): 295–306.

110. Chen R, Park JM and Marchany R. *Track: a novel approach for defending against distributed denial-of-service attacks*. Technical report TR ECE-06-02, February 2005. Blacksburg, VA: Department of Electrical and Computer Engineering, Virginia Tech.

111. Dean D, Franklin M and Stubblefield A. An algebraic approach to IP traceback. *ACM T Inform Syst Se* 2002; 5(2): 119–137.

112. Song DX and Perrig A. Advanced and authenticated marking schemes for IP traceback. In: *Proceedings of the 20th annual joint conference of the IEEE computer and communications societies (INFOCOM 2001)*, Anchorage, AK, 22–26 April 2001, vol. 2, pp.878–886. New York: IEEE.

113. Law TK, Lui JC and Yau DK. You can run, but you can't hide: an effective statistical methodology to trace back DDoS attackers. *IEEE T Parall Distr* 2005; 16(9): 799–813.

114. Goodrich MT. Probabilistic packet marking for large-scale IP traceback. *IEEE ACM T Network* 2008; 16(1): 15–24.

115. Tian H, Bi J, Jiang X, et al. A probabilistic marking scheme for fast traceback. In: *Proceedings of the 2010 2nd international conference on evolving internet (INTERNET)*, Valcencia, 20–25 September 2010, pp.137–141. New York: IEEE.

116. Peng T, Leckie C and Ramamohanarao K. Adjusted probabilistic packet marking for IP traceback. In: *Proceedings of the 2nd international networking conference on networking technologies, services, and protocols; performance of computer and communication networks; mobile and wireless communications (NETWORKING 2002)*, Pisa, 19–24 May 2002, pp.697–708. London: Springer-Verlag.

117. Wikipedia. Luby transform code—wikipedia, the free encyclopedia, 2017, https://en.wikipedia.org/w/index.php?title = Luby_transform_code&oldid = 780920020 (accessed 19 May 2017).

118. Fadlallah A. Adaptive probabilistic packet marking scheme for IP traceback. In: *Proceedings of the 2014 world congress on computer applications and information systems (WCCAIS)*, Hammamet, Tunisia, 17–19 January 2014, pp.1–5. New York: IEEE.

119. Yu S. *Distributed denial of service attack and defense*. New York: Springer, 2014.

120. Belenky A and Ansari N. IP traceback with deterministic packet marking. *IEEE Commun Lett* 2003; 7(4): 162–164.

121. Belenky A and Ansari N. Tracing multiple attackers with deterministic packet marking (DPM). In: *Proceedings of the 2003 IEEE Pacific rim conference on communications, computers and signal processing, 2003 (PACRIM)*, Victoria, BC, Canada, 28–30 August 2003, vol. 1, pp.49–52. New York: IEEE.

122. Jin G and Yang J. Deterministic packet marking based on redundant decomposition for IP traceback. *IEEE Commun Lett* 2006; 10(3): 204–206.

123. Xiang Y, Zhou W and Guo M. Flexible deterministic packet marking: an IP traceback system to find the real source of attacks. *IEEE T Parall Distr* 2009; 20(4): 567–580.

124. Yu S, Zhou W, Guo S, et al. A dynamical deterministic packet marking scheme for DDoS traceback. In: *Proceedings of the 2013 IEEE global communications conference (GLOBECOM)*, Atlanta, GA, 9–13 December 2013, pp.729–734. New York: IEEE.

125. Yu S, Zhou W, Guo S, et al. A feasible IP traceback framework through dynamic deterministic packet marking. *IEEE T Comput* 2016; 65(5): 1418–1427.

126. Bellovin SM, Leech M and Taylor T. ICMP traceback messages, 2003, https://academiccommons.columbia.edu/catalog/ac:127253

127. Lee HC, Thing VL, Xu Y, et al. ICMP traceback with cumulative path, an efficient solution for IP traceback. In: *Proceedings of the international conference on information and communications security*, Huhehaote, China, 10–13 October 2003, pp.124–135. Berlin; Heidelberg: Springer.

128. Izaddoost A, Othman M and Rasid MFA. Accurate ICMP traceback model under DoS/DDoS attack. In: *Proceedings of the international conference on advanced computing and communications, 2007 (ADCOM 2007)*, Guwahati, India, 18–21 December 2007, pp.441–446. New York: IEEE.
129. Mankin A, Massey D, Wu CL, et al. On design and evaluation of "intention-driven" ICMP traceback. In: *Proceedings of the 10th international conference on computer communications and networks*, Scottsdale, AZ, 15–17 October 2001, pp.159–165. New York: IEEE.
130. Saurabh S and Sairam AS. ICMP based IP traceback with negligible overhead for highly distributed reflector attack using bloom filters. *Comput Commun* 2014; 42: 60–69.
131. Yao G, Bi J and Vasilakos AV. Passive IP traceback: disclosing the locations of IP spoofers from path backscatter. *IEEE T Inf Foren Sec* 2015; 10(3): 471–484.
132. Liu X, Yang X and Lu Y. To filter or to authorize: network-layer dos defense against multimillion-node botnets. *ACM SIGCOMM Comp Com* 2008; 38(4): 195–206.
133. Mahajan R, Bellovin SM, Floyd S, et al. Controlling high bandwidth aggregates in the network. *ACM SIGCOMM Comp Com* 2002; 32(3): 62–73.
134. Yaar A, Perrig A and Song D. Siff: a stateless internet flow filter to mitigate DDoS flooding attacks. In: *Proceedings of the 2004 IEEE symposium on security and privacy*, Berkeley, CA, 12 May 2004, pp.130–143. New York: IEEE.
135. Yang X, Wetherall D and Anderson T. A DoS-limiting network architecture. *ACM SIGCOMM Comp Com* 2005; 35(4): 241–252.
136. Xiong S, Tian L, Li X, et al. Fault-tolerant topology evolution and analysis of sensing systems in IoT based on complex networks. *Int J Sens Netw* 2015; 18(1–2): 22–31.
137. Briscoe B, Jacquet A, Cairano-Gilfedder CD, et al. Policing congestion response in an internetwork using re-feedback. *ACM SIGCOMM Comp Com* 2005; 35(4): 277–288.
138. Liu X, Yang X and Xia Y. NetFence: preventing internet denial of service from inside out. *ACM SIGCOMM Comp Com* 2010; 40(4): 255–266.
139. Menth M, Martin R and Charzinski J. Capacity overprovisioning for networks with resilience requirements. *ACM SIGCOMM Comp Com* 2006; 36(4): 87–98.
140. Naili M, Achroufene A and Naili M. Election-based method for fault tolerance in a hierarchical sensor network EFTOHSN: a case study of an indoor localisation system. *Int J Sens Netw* 2016; 22: 158–165.
141. Sivasubramanian S, Szymaniak M, Pierre G, et al. Replication for web hosting systems. *ACM Comput Surv* 2004; 36(3): 291–334.
142. Yan J, Early S and Anderson R. The Xenoservice—a distributed defeat for distributed denial of service. In: *Proceedings of the ISW, Wollongong*, NSW, Australia, 20–21 December 2000, vol. 2000. Berlin; Heidelberg: Springer.
143. Azure M. What is cloud computing? A beginner's guide, https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/
144. Somani G, Gaur MS, Sanghi D, et al. DDoS attacks in cloud computing: issues, taxonomy, and future directions. *Comput Commun* 2017; 107: 30–48.
145. Xiao Z and Xiao Y. Security and privacy in cloud computing. *IEEE Commun Surv Tut* 2013; 15(2): 843–859.
146. Fu Z, Shu J, Wang J, et al. Privacy-preserving smart similarity search based on simhash over encrypted data in cloud computing. *J Internet Technol* 2015; 16(3): 453–460.
147. Xia Z, Wang X, Zhang L, et al. A privacy-preserving and copy-deterrence content-based image retrieval scheme in cloud computing. *IEEE T Inf Foren Sec* 2016; 11(11): 2594–2608.
148. Fu Z, Wu X, Guan C, et al. Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement. *IEEE T Inf Foren Sec* 2016; 11(12): 2706–2716.
149. Fu Z, Huang F, Ren K, et al. Privacy-preserving smart semantic search based on conceptual graphs over encrypted outsourced data. *IEEE T Inf Foren Sec* 2017; 12(8): 1874–1884.
150. Fu Z, Huang F, Sun X, et al. Enabling semantic search based on conceptual graphs over encrypted outsourced data. *IEEE T Serv Comput*. Epub ahead of print 27 October 2016. DOI: 10.1109/TSC.2016.2622697.
151. Shen J, Liu D, Shen J, et al. A secure cloud-assisted urban data sharing framework for ubiquitous-cities. *Pervasive Mob Comput* 2017; 41: 219–230.
152. Tara E. Q1 2015 DDoS attacks spike, targeting cloud, https://www.infosecurity-magazine.com/news/q1-2015-ddos-attacks-spike/
153. Darwish M, Ouda A and Capretz LF. Cloud-based DDoS attacks and defenses. In: *Proceedings of the 2013 international conference on information society (i-Society)*, Toronto, ON, Canada, 24–26 June 2013, pp.67–71. New York: IEEE.
154. Al-Haidari F, Sqalli MH and Salah K. Enhanced EDoS-shield for mitigating EDoS attacks originating from spoofed IP addresses. In: *Proceedings of the 2012 IEEE 11th international conference on trust, security and privacy in computing and communications (TrustCom)*, Liverpool, 25–27 June 2012, pp.1167–1174. New York: IEEE.
155. Karnwal T, Sivakumar T and Aghila G. A comber approach to protect cloud computing against XML DDoS and HTTP DDoS attack. In: *Proceedings of the 2012 IEEE students' conference on electrical, electronics and computer science (SCEECS)*, Bhopal, India, 1–2 March 2012, pp.1–5. New York: IEEE.
156. Sqalli MH, Al-Haidari F and Salah K. EDoS-shield—a two-steps mitigation technique against EDoS attacks in cloud computing. In: *Proceedings of the 2011 4th IEEE international conference on utility and cloud computing (UCC)*, Victoria, NSW, Australia, 5–8 December 2011, pp.49–56. New York: IEEE.
157. Jia Q, Wang H, Fleck D, et al. Catch me if you can: a cloud-enabled DDoS defense. In: *Proceedings of the 2014 44th annual IEEE/IFIP international conference on*

dependable systems and networks (DSN), Atlanta, GA, 23–26 June 2014, pp.264–275. New York: IEEE.

158. Wang H, Jia Q, Fleck D, et al. A moving target DDoS defense mechanism. *Comput Commun* 2014; 46: 10–21.
159. Baig ZA and Binbeshr F. Controlled virtual resource access to mitigate economic denial of sustainability (EDoS) attacks against cloud infrastructures. In: *Proceedings of the 2013 international conference on cloud computing and big data (CloudCom-Asia)*, Fuzhou, China, 16–19 December 2013, pp.346–353. New York: IEEE.
160. Saini B and Somani G. Index page based EDoS attacks in infrastructure cloud. In: *Proceedings of the SNDS: international conference on security in computer networks and distributed systems*, Trivandrum, India, 13–14 March 2014, pp.382–395. Berlin; Heidelberg: Springer.
161. Idziorek J, Tannian M and Jacobson D. Detecting fraudulent use of cloud resources. In: *Proceedings of the 3rd ACM workshop on cloud computing security workshop*, Chicago, IL, 21 October 2011, pp.61–72. New York: ACM.
162. Ismail MN, Aborujilah A, Musa S, et al. Detecting flooding based DoS attack in cloud computing environment using covariance matrix approach. In: *Proceedings of the 7th international conference on ubiquitous information management and communication*, Kota Kinabalu, Malaysia, 17–19 January 2013, p. 36. New York: ACM.
163. Latanicki J, Massonet P, Naqvi S, et al. Scalable cloud defenses for detection, analysis and mitigation of DDoS attacks. In: *Proceedings of the future Internet assembly*, Valencia, 15–16 April 2010, pp.127–137, http://www.future-internet.eu/
164. Li B, Niu W, Xu K, et al. You can't hide: a novel methodology to defend DDoS attack based on botcloud. In: *Proceedings of the 6th international conference on applications and techniques in information security*, Beijing, China, 4–6 November 2015, pp.203–214. Berlin; Heidelberg: Springer.
165. Alqahtani S and Gamble RF. DDoS attacks in service clouds. In: *Proceedings of the 2015 48th Hawaii international conference on system sciences (HICSS)*, Kauai, HI, 5–8 January 2015, pp.5331–5340. New York: IEEE.
166. Yu S, Tian Y, Guo S, et al. Can we beat DDoS attacks in clouds? *IEEE T Parall Distr* 2014; 25(9): 2245–2254.
167. Wang B, Zheng Y, Lou W, et al. DDoS attack protection in the era of cloud computing and software-defined networking. *Comput Netw* 2015; 81: 308–319.
168. Wang X, Chen M and Xing C. SDSNM: a software-defined security networking mechanism to defend against DDoS attacks. In: *Proceedings of the 2015 9th international conference on frontier of computer science and technology (FCST)*, Dalian, China, 26–28 August 2015, pp.115–121. New York: IEEE.
169. Aloul F, Al-Ali A, Al-Dalky R, et al. Smart grid security: threats, vulnerabilities and solutions. *Int J Smart Grid Clean Energ* 2012; 1(1): 1–6.
170. Lu Z, Lu X, Wang W, et al. Review and evaluation of security threats on the communication networks in the smart grid. In: *Proceedings of the military communications conference 2010 (MILCOM 2010)*, San Jose, CA, 31 October–3 November 2010, pp.1830–1835. New York: IEEE.
171. Wang W and Lu Z. Cyber security in the Smart Grid: survey and challenges. *Comput Netw* 2013; 57(5): 1344–1371.
172. Locke G and Gallagher PD. *Nist framework and roadmap for smart grid interoperability standards* (release 1.0). Gaithersburg, MD: National Institute of Standards and Technology, 2010, p.33.
173. Liu J, Xiao Y, Li S, et al. Cyber security and privacy issues in smart grids. *IEEE Commun Surv Tut* 2012; 14(4): 981–997.
174. Sheng Y, Tan K, Chen G, et al. Detecting 802.11 MAC layer spoofing using received signal strength. In: *Proceedings of the 27th conference on computer communications (INFOCOM 2008)*, Phoenix, AZ, 13–18 April 2008, pp.1768–1776. New York: IEEE.
175. Xu W, Trappe W, Zhang Y, et al. The feasibility of launching and detecting jamming attacks in wireless networks. In: *Proceedings of the 6th ACM international symposium on mobile ad hoc networking and computing*, Urbana-Champaign, IL, 25–27 May 2005, pp.46–57. New York: ACM.
176. Li M, Koutsopoulos I and Poovendran R. Optimal jamming attacks and network defense policies in wireless sensor networks. In: *Proceedings of the 26th IEEE international conference on computer communications (INFOCOM 2007)*, Barcelona, 6–12 May 2007, pp.1307–1315. New York: IEEE.
177. Mantas G, Lymberopoulos D and Komninos N. Security in smart home environment, 2011, http://www.irma-international.org/viewtitle/47126/
178. Auriemma L. Samsung devices with support for remote controllers, 2012, http://aluigi.altervista.org/adv/samsux_1-adv.txt (Zugriff am 16, 2014).
179. Nunes G. Sony Bravia—remote denial of service, 2012, https://www.exploit-db.com/exploits/18705/
180. Yoon S, Park H and Yoo HS. Security issues on smarthome in IoT environment. In: Park J, Stojmenovic I, Jeong H, et al. (eds) *Computer science and its applications*. Berlin; Heidelberg: Springer, 2015, pp.691–696.
181. BBC. "Smart" home devices used as weapons in website attack, 2016, http://www.bbc.com/news/technology-37738823
182. DDoS attack that disrupted internet was largest of its kind in history, experts say. *The Guardian*, 2016, https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet
183. Heer T, Garcia-Morchon O, Hummen R, et al. Security challenges in the IP-based internet of things. *Wireless Pers Commun* 2011; 61(3): 527–542.
184. Strategy I and Unit P. *ITU Internet reports 2005: the Internet of Things*. Geneva: International Telecommunication Union (ITU), 2005.
185. DuBravac S and Ratti C. The internet of things: evolution or revolution? 2015, https://www.aig.com/content/dam/aig/america-canada/us/documents/insights/aig-white-paper-iot-english-digital-brochure.pdf
186. Trend Micro, Inc. The internet of things ecosystem is broken. How do we fix it? 2016, http://blog.trendmicro.com/trendlabs-security-intelligence/internet-things-ecosystem-broken-fix/

187. European Union Agency for Network and Information Security. Major DDoS attacks involving IoT devices, 2016, https://www.enisa.europa.eu/about-enisa/cookies

188. Kreb B. KrebsOnSecurity hit with record DDoS, 2016, https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/

189. OVH. The DDoS that didn't break the camel's VAC, 2016, https://www.ovh.com/us/news/articles/a2367.the-ddos-that-didnt-break-the-camels-vac

190. Hilton S. Dyn analysis summary of Friday October 21 attack, 2016, https://dyn.com/blog/dyn-analysis-summary-of-friday-october-21-attack/

191. Baheti R and Gill H. Cyber-physical systems. In: Samad T and Annaswamy AM (eds) *The impact of control technology*, vol. 12. New York: IEEE, 2011, pp.161–166.

192. Rajkumar RR, Lee I, Sha L, et al. Cyber-physical systems: the next computing revolution. In: *Proceedings of the 47th design automation conference*, Anaheim, CA, 13–18 June 2010, pp.731–736. New York: ACM.

193. Xiao Y. Flow-net methodology for accountability in wireless networks. *IEEE Network* 2009; 23: 30–37.

194. Shen J, Shen J, Chen X, et al. An efficient public auditing protocol with novel dynamic structure for cloud data. *IEEE T Inf Foren Sec* 2017; 12: 2402–2415.