Journal of Information Security, 2013, 4, 150-164
http://dx.doi.org/10.4236/jis.2013.43018 Published Online July 2013 (http://www.scirp.org/journal/jis)

Hadoop Based Defense Solution to Handle Distributed Denial of Service (DDoS) Attacks

Shweta Tripathi[1], Brij Gupta[1*], Ammar Almomani[2], Anupama Mishra[1], Suresh Veluru[3]
[1]School of Computing Science & Engineering, Galgotias Universiy, Greater Noida, India
[2]Faculty of Computing and Information Technology, North Jeddah Branch, King Abdulaziz University, Jeddah, Saudi Arabia
[3]School of Engineering and Mathematical Sciences, City University London, London, UK
Email: *gupta.brij@gmail.com

Received May 21, 2013; revised June 22, 2013; accepted June 30, 2013

ABSTRACT

Distributed denial of service (DDoS) attacks continues to grow as a threat to organizations worldwide. From the first known attack in 1999 to the highly publicized Operation Ababil, the DDoS attacks have a history of flooding the victim network with an enormous number of packets, hence exhausting the resources and preventing the legitimate users to access them. After having standard DDoS defense mechanism, still attackers are able to launch an attack. These inadequate defense mechanisms need to be improved and integrated with other solutions. The purpose of this paper is to study the characteristics of DDoS attacks, various models involved in attacks and to provide a timeline of defense mechanism with their improvements to combat DDoS attacks. In addition to this, a novel scheme is proposed to detect DDoS attack efficiently by using MapReduce programming model.

Keywords: DDoS; DoS; Defense Mechanism; Characteristics; Hadoop; MapReduce

1. Introduction

DDoS attack is a distributed, large scale coordinated attempt of flooding the network with an enormous amount of packets which is difficult for victim network to handle, and hence the victim becomes unable to provide the services to its legitimate user and also the network performance is greatly deteriorated [1]. This attack exhausts the resources of the victim network such as bandwidth, memory, computing power etc. The system which suffers from attacked or whose services are attacked is called as "primary victim" and on other hand "secondary victims" is the system that is used to originate the attack. These secondary victims provide the attacker, the ability to wage a more powerful DDoS attack as it is difficult to track down the real attacker [2].

Denial of Service (DoS) attacks is used to consume all the resources of the target machine (victim's services) and becomes a known issue in 1980's. But, in 1990's these attacks have been noticed as it becomes a serious problem to the Internet society gradually [2-4]. DDoS attack is a distributed, large scale coordinated attempt of exhausting the network with an enormous amount of request, which overload the victim's machine and the victim's machine becomes unable to provide the services to its legitimate user and hence the network performance will be greatly deteriorated.

In DDoS attack, the attacker selects the compromised machine (*i.e.* those machines which have loopholes) and network of the compromised machines are called botnet. These botnets are further instructed to execute commands in order to consume all the resources available on victim's system. Currently attacks are being launched by using two approaches. The first approach is to send malicious packet injected with virus, worms as a running application, is called as vulnerability attack. The other very common method is to debilitate the victim's system, by exhausting the resources such as input-output bandwidth, database bandwidth, CPU, memory, etc. [5].

A group called "Izz ad-Din al-Qassam Cyber Fighters" [6] has launched DDoS attack against many US Banks such as Bank of America, Citi Group, HSBC and Capital One. As a result, these online banking sites have degraded. **Figure 1** shows various attacks over the years. From the figure, we can see that total number of attacks increases gradually every year. **Table 1** shows some serious DDoS attack incidents in past years. It is noted that attacks incidents are increasing gradually specially in financial market.

*Corresponding author.

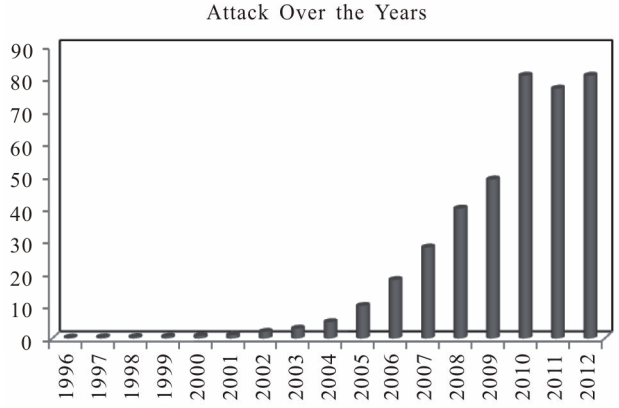

Figure 1. Timeline of attacks over the years.

Rest of the paper is organized as follows: Section 2 describes DDoS attacker's motivation factors, Section 3 contains history of DDoS attacks, Section 4 presents DDoS attack characteristics and models, Section 5 describes DDoS attack toolkit, Section 6 presents how DDoS attacks are preformed using botnet, Section 7 describes various DDoS defense mechanisms, Section 8 contains our proposed model for DDoS attacks detection and finally, Section 9 concludes the paper and discusses some future work.

2. DDoS Attacker's Motivation Factors

Human beings are not born to become an attacker. They are enough motivated due to some reasons to launch the attack. Based on some obvious reasons and facts, the motivation factor can be categorized as [7]:

1) Financial Benefit

The attackers of this category are highly skilled and hard to be detected. They only concern here is to have financial gain.

2) Professional Skills

The attackers target systems for experiment purpose to check their vulnerabilities and strength of security mechanism. The attackers who are very much enthusiastic and ready to face challenges fall into this category.

3) Payback Attitude

In this category, the attackers are usually very much frustrated and low skilled persons, perform attack only to take revenge.

4) Cyber Warfare

In this category, attackers are usually high skilled and intellectual person who generally belong to military or terrorist organizations of a country. They attack to defend their country or their organizations [8].

3. History of DDoS Attacks

3.1. Analytical Study of DDoS Attacks

A long run way which has no end point of attacks can be seen even in advanced technical society. To develop defense mechanism, behavior of attack can be analyzed, which leads to the categorization of DDoS attack.

Practical Unix and Internet Security [14], the "bible" for many system administrators of the early commercial web, offers a chapter on denial of service attacks. Carnegie Mellon's Computer Emergency Response Team*

Table 1. DDoS attack statistics.

2013	The Czech financial sector was targeted in cyber attacks on Wednesday, at the same time on the national bank and stock exchange websites which get disrupted by dedicated denial of service (DDoS) attacks—London, 8 March, 2013.
2012	US and UK Government Sites Knocked Down by Anonymous—April 16, 2012. DDoS Attack Impacts Canadian Political Party Elections—March 24, 2012.
2011	A DDoS attack on Sony was used—April 16-20 2011.
2010	PayPal Transaction is suspended over WikiLeaks website after attacked by DDoS—December 3-5, 2010.
2009	The Mydoom virus code was re-used to launch DDoS flooding attacks against major government news media and financial websites in South Korea and the United States in July 2009 [9].
2008	BBC hit by DDoS Attack, two DDoS attacks on Amazon.com and eBuy.
2007	Estonia Cyber Attack [10].
2006	US Banks have been targeted for financial gain.
2004	SCO Group website inaccessible to legitimate users.
2003	Mydoom defiled thousands of victims to attack SCO and Microsoft [11].
2002	13 root servers that provide the Domain Name System (DNS) service to Internet users around the world shut down for an hour because of a DDoS flooding attack [12].
2001	First major attack involving DNS servers as reflectors. The target was Register.com. The Irish Government's Department of Finance server was hit by a denial of service attack carried out as part of a student campaign from NUI Maynooth.
2000	Yahoo! Experienced one of the first major DDoS flooding attacks that kept the company's services off the Internet for about 2 hours incurring a significant loss in advertising revenue [13].

(CERT) [15,16] published its first bulletin on SYN flooding* (a popular technique for overwhelming target system) in September 1996, and a more thorough bulletin on denial of service in October 1997, suggesting that denial of service was beginning to emerge as a priority for network administrators. While CERT and others offered helpful advice for mitigating DDoS attacks, the particular attack documented in 1996—SYN flooding—is still common today, pointing to the wide gap between understanding these attacks and successfully defending against them. Similarly, the US National Information Infrastructure Protection Act of 1996 took steps to criminalize DDoS, redefining computer fraud "damage" as preventing the right to use a computer system. Previous definitions had focused on unauthorized access and damage to systems. But as per Arbor's annual survey reports, many system administrators do not bother to reporting DDoS attacks to the authorities.

Shortly after denial of service emerged as a concern for system administrators, activists began using it as a political technique. Ricardo Dominguez, co-founder of Electronic Disturbance Theatre, was one of the leaders in using denial of service as a tool for activists in 1998. He built FloodNet, a tool designed to allow activists to crash the websites of the Frankfurt Stock Exchange, the Pentagon, and Mexican President Ernesto Zedillo [17]. But as these protests failed to shut down the sites, they were not much discussed outside the art community. Denial of service took on new visibility and importance in February 2000, when it took down several websites like Yahoo, Buy.com, ZDNet.com, eBay, CNN, Amazon.com.

DDoS attacks became more common in 2000 and in 2001, it is used to compromise large numbers of Windows systems. Worms and Trojan horse programs sent via email demonstrated the ability to exploit known vulnerabilities to compromise large numbers of systems [18]. At the same time, attackers began to organize compromised computers into networks centrally controlled by IRC "bots". These "botnets", allows a single controller to manipulate thousands of compromised computers and order them to send spam email and do other mischievous things like stealing credit card information, or mounting DDoS attacks.

Most existing techniques for defending against denial of service attacks were based on identifying the attacking computers by IP address. Botnets invalidated many of these techniques because a single botnet could include thousands of computers with randomly distributed IP addresses, which is very difficult to distinguish by IP address alone.

Despite the rise of botnets, various other forms of DDoS have continued to ask for media coverage and attention. Recently, an organization named "Help Israel Win" invited individuals to install a software package ("Patriot DDoS") on their PCs which would give a remote administrator the capability to harness the machine in an attack on a (Palestinian) target [19].

In 2010, during the Iranian Green Movement protests, protesters used a page refreshing service to manually execute a DDoS attack that was an attempt to bring down President Mahmoud Ahmadinejad's website [20,21]. On the similar lines, "Operation Payback" requires participants to download a software named "Low Orbit Ion Cannon" that allows a computer to become part of a botnet controlled by administrators of the Anonymous group via IRC.

Some (In) famous DDoS Attacks.

The Iranian Cyber Army: It happened on December 17, 2009, when attackers replaced the front page of a famous social networking site, Twitter.com with an image of the Iranian flag along with text including: "This site has been hacked by the Iranian Cyber Army", although they could not succeed in their act, but managed to change the twitter.com domain name to point to a other IP address. The attack causes Twitter to take down its home page and twitter.com remained down for a couple of hours [22,23].

The attacks on the major Web sites began in early February 2000, with the first major attack being on Yahoo! The surprise attack took the Yahoo! Site down for more than three hours. It was based on the Smurf attack, and most likely, the Tribe Flood Network technique. At the peak of the attack, Yahoo! was receiving more than one gigabit per second of data requests.

In February, 2010, a group of people loosely connected through Internet forums calling themselves "Anonymous" executed a DDoS attack against the Australian Parliament's website. The attack not only took down the site for two days but also defaced the Prime Minister's website, by replacing the front page with pornographic images for a brief period of time. The attack was termed "Operation Titstorm" by its organizers [24] referring to a mandatory Internet filtering policy proposed by Australia's ruling party designed in part to counter pornography [25].

3.2. DDoS Observations

1) The ideology of an attacker and the method chosen for attacks is not correlated.

2) It is found that there is specific geographic pattern of DDoS attacks.

3) Easily accessible tools that helps to make successful attacks on small websites, suggests that distressed individuals may use DDoS as a weapon for building score or making a political point.

3.3. Recent Attacks

1) Mt. Gox under largest DDoS attack

The largest bit coin exchange said that on April 4 2013, it is fighting an intense distributed denial-of-service attack and it believes that it is intended at manipulating the price of virtual currency, which has seen unstable price fluctuation in the past few days. According to Facebook, Mt. Gox, which is based in Tokyo, the attacks have caused its worst trading lags ever and caused error pages to be displayed to traders. As per their own estimation, 80 percent of the bit coin trades in US dollars are executed on Mt. Gox's trading platform and a significant amount of trade in other currencies [26-28].

2) American Express under DDoS Attack

American Express confirms it was hit by a distributed-denial-of-service attack that disrupted online-account access for about two hours during the late afternoon on March 28. The attack began at about 3:00 PM ET on March 28, caused intermittent disruptions. It was said that there is no evidence to suggest that customer data or account information was exposed or compromised during the attack. AmEx issued a statement regarding the attack on how their operations were getting affected by DDoS attack [29-31].

3) Attack on Spamhaus

UK and Switzerland-based nonprofit organization, which operates a filtering service, has been strike by distributed denial of service (DDoS) attack, which has proven to be the largest DDoS till today. Security firm Kaspersky Lab confirmed the attack and claimed it to be the largest DDoS cyber-attack. As per Kaspersky Lab the attack was evaluated to at 300 Gigabits per second and supposed to be one of the largest DDoS operations to date [32].

4) Latest attacks on banks of US

In December 14, 2012, the major US banks websites were attacked. The attackers, who call themselves the Izz ad-Din al-Qassam Cyber Fighters, launched attacks on Tuesday against the websites of US Bancorp, JP Morgan Chase & Co., Bank of America, PNC Financial Services-Group and SunTrust Banks. Dan Holden, who is director of security research at Arbor Networks, said "While the DDoS attack could not hamper the online operation of bank but they taught lesson to those who faced the threat" [29].

In Nov. 8 2012, Webster Bank and Zions Bancorp joined the list of banks which experienced the online outages linked to distributed-denial-of-service attacks. Webster, a $20 billion institution based in Connecticut, a DDoS attack hit its website at about 4:30 p.m. Nov. 6 and continued until about 2 a.m. Nov. 7. And Zions, a $53 billion bank based in Utah, an attack caused four hours of intermittent outages for online-banking and website access during the late afternoon and evening of Nov. 8.

5) Go Daddy stopped by DDoS attack

An attacker has claimed responsibility for DoS attack that has knocked out millions of website hosted by world's largest domain registrar GoDaddy [33].

6) iMassage DDoS attack

A group of iOS developers are targeted with a series of rapid-fire texts sent over Apple's iMessage system. The messages which seem to be transmitted via the OS X Messages application used a simple AppleScript which rapidly fill up the Messages app on iOS or the Mac with text and force users to constantly clear both notifications and messages. In some of the cases, the messages were so large that they completely lock up the Messages app on iOS, constituting a "denial of service" (DoS) attack [32].

3.4. Well-Known DoS Attacks Mechanism

This paper would be incomplete without reference to some of the most well-known DDoS attacks. Some of the most famous standard DDoS attacks are summarized as follows:

- Apache 2: This attack is build up against an Apache Web server where the client asks for a service by sending a request with many HTTP headers. Upon receiving the large amount of HTTP request Apache Web server cannot outface the load and it crashes.
- ARP Poison: Address Resolution Protocol (ARP) Poison attacks claims the attacker to have key in to the victim's LAN. The attacker spoof the hosts of a specific LAN by providing them with wrong MAC addresses for hosts with already-known IP addresses. This can be done by the attacker through the following procedure: The network is monitored for "who-has" requests type which is an ARP request. The moment such a request is received; the malevolent attacker tries to respond as fast as feasible to the questioning host so that it can mislead it for the requested address.
- Back: In Back type of attack the requests are send an apache Web server, where the server is flooded with requests containing a large number of front-slash (/) characters in the URL description. When the server tries to process all these requests, it becomes unable to process other legitimate requests and hence it denies service to its legitimate user.
- CrashIIS: The CrashIIS attack is commonly a projected towards Microsoft Windows NT IIS Web server. The attacker sends the victim a malicious GET request, which causes the Web server to crash.
- Land: In this type of attack the attacker sends TCP SYN packet to the victim that contains the same IP address as the source and destination addresses. Such a packet completely blocks the victim's system.
- DoS Nuke: This kind of attack is launched against the Microsoft Windows NT victim is inundated with

"out-of-band" data (MSG_OOB). The packets that are sent by the attacking machines are flagged "urg" because of the MSG_OOB flag. This causes the target to get down, and this leads to displays a "blue screen of death" on the victim machine.

- Mail bomb: In this type of attack, the victim's mail queue is flooded by a huge amount of messages, causing system failure.
- SYN Flood: A SYN flood attack take place during the three-way handshake that marks the onset of a TCP connection. In the three-way handshake, a client sends a TCP SYN packet to a server requesting for a new connection. Thereby, the server sends a SYN/ ACK packet back to the client and places the connection request in a queue. As a final point, the client acknowledges the SYN/ACK packet. When an attack takes place, however, the attacker sends an abundance of TCP SYN packets to the victim, forcing it for both: 1) to open a lot of TCP connections and 2) to respond to them. Then the attacker does not execute the final step of the three-way handshake that follows, exposing the victim that is not capable to accept any new incoming connections, since its queue is full of half-open TCP connections.
- Ping of Death: In Ping of Death attacks, the attacker creates a packet that contains more than 65,536 bytes, which is out of the limit of the IP protocol. This packet can produce different kinds of damage to the machine that receives it, that results in crashing and rebooting.
- Process Table: This attack use the feature of some network services to generate a new process each time a new TCP/IP connection is set up. The attacker considers making as many uncompleted connections to the victim as possible in order to force the victim's system to generate as many as processes. For this reason, as the number of processes that are running on the system cannot be very much large, the attack renders the victim unable to serve any other request.
- Smurf Attack: In a "smurf" attack, the victim is thronged with Internet Control Message Protocol (ICMP) "echo-reply" packets. The attacker sends voluminous ICMP "echo-request" packets to the broadcast address of numerous subnets. These packets have the source IP address field updated with victims address. Every machine that is associated with any of these subnets responds by sending ICMP "echo-reply" packets to the victim. Smurf attacks are very alarming, because they are intensely distributed attacks.
- SSH Process Table: This attack makes large amount of connections to the victim with the Secure Shell (SSH) Protocol without carrying out the login process. In this way, the zombie contacted by the SSH on the victim's system is indulged to start so many SSH processes that it is fatigued.
- Syslogd: In this type of attack the Solaris 2.5 server is banged by sending large amount of messages with illegal source IP address.
- TCP Reset: In TCP Reset attacks, the network is scrutinized for "tcp connection" requests which are send to the victim. The moment such a request is found; the malicious attacker sends a spoofed TCP RESET packet to the victim and obliges it to lay off the TCP connection.
- Teardrop: A Teardrop attack causes a stream of IP fragments with their offset field overloaded. As a packet travels from the source machine to the destination machine, it is broken up into smaller sections or fragments, through the process of fragmentation. The destination host that tries to reassemble these abnormal fragments in the long run clangs or reboots.
- UDP Storm: In a User Datagram Protocol (UDP) connection, when it receive a UDP packet, a character generation ("chargen") service generates a series of characters, while an echo service echoes any character it receives. Manipulating the above two services, the attacker sends a packet to another machine with the source misleading to be that of the victim. Then, the echo service of the anterior machine echoes the data of that packet back to the victim's machine and the victim's machine, consecutively, responds in the similar fashion. Hence, a constant stream of unserviceable load is created that problems the network [32].

4. DDoS Attack: Characteristics and Models

4.1. Characteristics of DDoS Attack

Following are the different ways to characterize the distributed denial of service attack:

1) Disruptive/Degrade Impact

After being a part of attack, the victim either to stop providing services to the client or the services are degraded that means some of the services are still being provided to the client even the victim's system is under the attack.

2) Exploiting Vulnerability

Network of machines which follows the instructions of master attacker to send request for a service on a victim's machine to consume its all the resources.

3) Dynamic Attack Rate

Sometime attacker make down the websites very quickly by sending large no of request more than its capacity, is known as constant attack rate. While sometimes attacker takes time to make it down by sending packets in variable length of request that is not constant, known as variable attack rate.

4) Automated Tools

Attackers can be classified by automated tools also and their skills. Attack can be performed manually; semi automated or fully automated tools

4.2. DDoS Attacks Components

Figure 2 describes the component of DDoS attack, who initiates the attack by selecting vulnerable system as agents and further the agents use botnet to exhaust the victim's system.

1) Master Mind/Planner: The Original Attacker, who creates reasons and answers for, why, when, how and by whom the attack will be performed.

2) Controller/Handler: Co-ordinator of original attacker, who may be one or more than one machine, is used to exploit other machines to process DDoS attack

3) Agents/Zombies/Botnets: Agents, also known as slaves or attack daemons, sub ordinates are programs that actually conduct the attack on the victim. These programs are usually deployed on host computers. These daemons influence both the machines: target and the host computers. It facilitates the attacker to gain access and infiltrate the host computers.

4) Victim/Target: A victim is a target host that has been selected to receive the impact of the attack.

4.3. DDoS Architecture Models

Two types of DDoS attack networks have emerged: the Agent-Handler model and the Internet Relay Chat (IRC)-based model [1,5,35].

1) The Agent-Handler model of a DDoS attack consists of agents, handlers and client. **Figure 3** shows the Agent-Handler Model, in which the Agent and handler knows each-others identity. The client is the interface where the attacker/mastermind communicates with the rest of the DDoS Components. The handlers are software

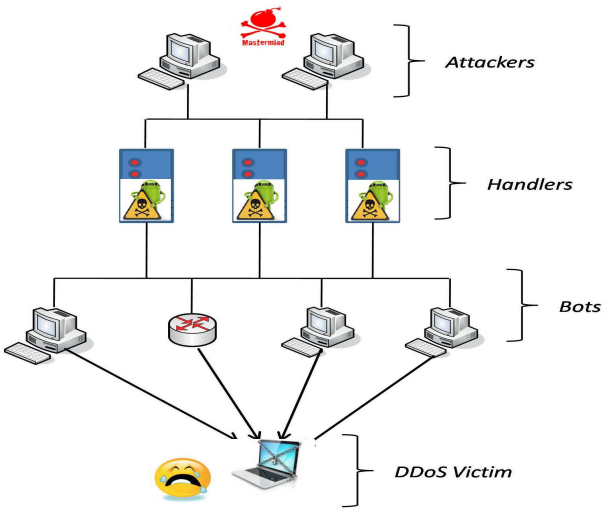

Figure 2. Components of DDoS attack [34].

packages distributed all over the Internet so that it helps to client to convey its command to the agents. The agent software's are vulnerable systems, compromised by the handlers and actually launch the attack on victim's machine. The agent's status and schedule for launching attack can be upgraded by the handler when it is required. Communication relation between agent and handler is either one to one or one to many. Most Common way to attack is by installing handler instructions either on compromised route on network layer or on network server. This makes it difficult to identify messages exchanged by the client-handler and between the handler-agents.

2) The IRC-based DDoS attack: IRC *i.e.* Internet Relay Chat, **Figure 4** shows the architecture of this model where attacker and agent does not know their identity. It is a communication channel to connect the clients to the agents, which provides some additional benefits to the attacker such as use of IRC ports to send the commands to the agents. Because of this, tracking the DDoS command packets becomes difficult. In addition to that, because of heavy traffic going through IRC servers attacker can easily hide its presence. As the attacker has direct access of IRC server, the attacker has access to a list of all available agents [36]. The attacker does not need to have a list of the agents. The agent software that installed in the IRC network which communicates to the IRC channel, notifies the attacker on when the agent is up and running.

5. DDoS Attack Toolkit

With time the attackers are using sophisticated tools to materialize the attacks, this sections lists the tool kits used in some of the attacks discussed in this paper.

1) Trinoo: It uses TCP to communicate between attacker and control master program. The communication between the trinoo master and daemon is held using UDP packets. It implements UDP flood attack against victim. The master and daemons are password protected and prevent system administrators to take control of the trinoo network [5].

2) WIN TRINOO: This is a variant trinoo that works on Windows platform. It sends large amount of UDP packets to the victim as an action of attack.

3) MStream: The mstream program which is based on the "stream.c" attack, includes a "master controller" and a "zombie". As the name indicates master controller controls all of the zombie agents. There is no encryption in the communications between the client, master, and zombie. An attacker connects to the master controller using Telnet to control the zombies. The zombie can slow a computer down by using up CPU cycles via a modified version of stream's attack .The attack consumes network bandwidth when the target host tries to send

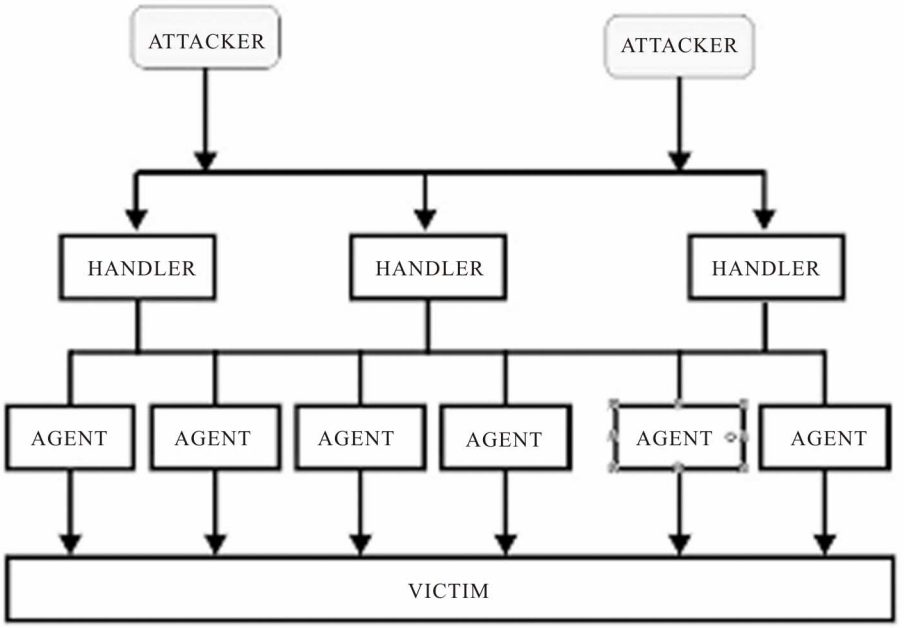

Figure 3. Agent-handler model.

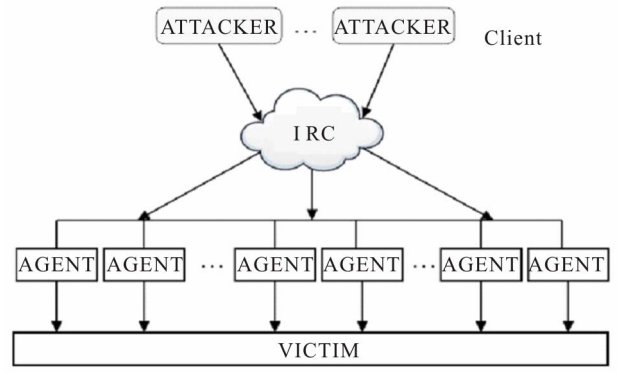

Figure 4. IRC model.

TCP RST packets to non-existent IP addresses in addition to the incoming ACK packets which cause Routers to return ICMP host/network unreachable packets to the victim, consequential the starvation of bandwidth. This consumes large amount of network bandwidth and at the same time distributed method of attack multiplies the effect on the CPU.

4) Tribe Flood Network (TFN): In this technique, a command line interface is used to communicate between attacker and control master program. The communication between the two is done through ICMP Echo reply packets. Following attacks are implemented through TFN's attack daemons: Smurf attack, SYN flooding, UDP flood and ICMP flood attack [37,38].

5) Stacheldraht: Stacheldraht is another master/slave DDoS attack toolkit based on TFN attack. But unlike TFN, it uses an encrypted TCP connection to communicate between attacker and master control program. Communication between master and daemon is held using TCP and ICMP and it involves an automatic update technique for attack daemons. Following attacks are implemented through stacheldraht attack daemons: smurf, UDP flood, ICMP flood attacks, SYN flood [39].

5) Shaft: It is modeled after trinoo. But unlike trinoo, the communication between control master program and attack daemons is achieved using UDP packets and they communicate via a simple TCP Telnet connection. An important feature of shaft is its ability to switch control master servers and ports in real time and hence making detection by intrusion detection tools difficult. Hence, attacks implemented through Shaft are difficult to detect [40].

6) TFN2K: Uses TCP, UDP, ICMP or all three to communicate between control master and program and the attack daemons. Communication between the real attacker and control master is encrypted using key based CAST-256 algorithm.

6. DDoS Attack Using Botnet

Botnets implement under a command and control (C & C) management infrastructure and compromise a network of machines with programs referred as bot, zombie, or drones [41]. The Botnets affects a series of systems using various tools and by installing a bot that can remotely control the victim using IRC. Present botnets are most frequently used to spread DDoS attacks on the Web [34]. Moreover, the attackers can change their communication approach during the creation of the bots. Majority of bots varied its potentials to participate in such attacks. The most classic and generally implemented Botnet attack on

application layer is the HTTP/S flooding attack, which launches bots created by the HTTP server. Such bots are thus called, Web-based bots [42].

The goal of a Botnet based DDoS attack is to entail damage at the victim side. In general, the mysterious intention behind this attack is personal which means block the available resources or degrade the performance of the service which is required by the target machine. Therefore, DDoS attack is committed for the revenge purpose. Another aim to perform these attacks can be to gain popularity in the hacker community. In addition to this, these attacks can also perform for the material gain, which means to break the confidentiality and use data for their use.

7. Defense Mechanisms against Attacks

With the passage of time, DDoS attack techniques have become technically more advanced and hence difficult to detect .There are a number of safety measures that can be performed to make network and neighbor network more secure and reliable to use. The classifications are:

7.1. Prevention Techniques

There are some prevention techniques to prevent the attack. **Table 2** shows not only the various prevention techniques but also focuses on their limitations.

1) Filtering routers: It involves filtering all the packets that either enter or leave the network. This defense mechanism protects the network from malicious attacks and prevents itself from unaware attacker. Even this method can be implemented to defense the DDOS in cloud environment also [43]. This measure requires installation of ingress and egress packet filter on all routers.

2) Disabling unused services: If UDP echo or other unused services exist then services should be disabled to prevent tampering and attacks [44].

3) Applying security patches: To prevent denial of service attacks, host computers must be reorganized with the most recent security patches and techniques. For example, in the case of the SYN Flood attack [29], following measures are taken: increase the size of the connection queue, decrease the time-out waiting for the three-way handshake, and employ vendor software patches to detect and circumvent the problem.

4) IP hopping: DDoS attacks can be prevented by changing the victim computer's IP address with a pre-specified set of IP address ranges, thereby invalidating the old address [44].

5) Disabling IP broadcast: The malicious part of this attack is that the attacker can use a low-bandwidth connection to destroy high-bandwidth connections. The amount of packets that are sent by the attacker is multiplied by a factor equal to the number of hosts behind the router that reply to the ICMP echo packets. So, disabling IP broadcast can be used to defend against the DDoS attack.

So prevention schemes are not reliable because they prevent only IP spoofing which is an outdated way of attacking the host. According to the Internet Architecture Working Group (2005), the percentage of spoofed attacks is declining. Only 4 out of 1127 customer impacting DDoS attacks on a large network used spoofed sources in 2004 [3,44].

7.2. Detection Techniques

DDoS detection mechanism can be classified based on two primary criterions.

1) Detection Timing—Passive detection is a form of detection which is done by analyzing the logs, after the attacker has finished this mission, the detection can be on time if the attack can be detected during the time of attack proactive detection is the detection of attack before it approaches the target machine or before the ruin of the service.

2) Detection activity—Here we are presenting some of the existing detection approaches [45-51]. **Table 3** briefly describes those approaches and their limitations. Based on detection activity the categorization is as follows.

a) Signature based—It involves priori knowledge of attack signatures [52]. SNORT are the two widely used signature-based detection approaches.

b) Anomaly based—It treats any incoming traffic that is violating the normal profile as an anomaly. For detecting DDoS attacks it is first require to know the normal behavior of the host and then finding deviations from that behavior.

Limitation: The common challenge for all anomaly-based intrusion detection systems is that it is difficult to take into account the data that provide all types of normal traffic behaviour. As a result, legitimate traffic can be classified as attack traffic which will result in a false positive. In order to reduce the false positive rate, a many parameters are used to provide more accurate normal profiles, which may increase the computational overhead to detect attack.

c) Hybrid attack detection: Hybrid attack detection has the optimistic features of both: 1) pattern-and 2) anomaly-based attack detection models to achieve high detection accuracy, low false positives and negatives, and increased level of cyber conviction. Even though hybrid attack detection approach decreases false positive rate, it also increases complexity and cost of implementation [52].

d) Third party detection: Mechanisms that deploy third-party detection do not handle the detection process themselves but rely on an external third-party that signals the occurrence of the attack [53]. Examples of mechanisms

Table 2. Prevention techniques-limitations.

Prevention Technique	Limitation
Filtering routers	New signatures cannot be detected.
Disabling unused services	By default the installations of operating systems often include many applications not needed by a user.
Applying security patches	New security patches are launched every day
IP hopping	The attacker can launch the attack at the new IP address.
Disabling IP broadcast	Defense against attacks that use intermediate broadcasting nodes e.g. Smurf attacks, ICMP flood attacks etc. will be successful only if host computers and all the neighboring networks disable IP broadcast.

Table 3. Detection and response techniques-limitations.

Detection Technique	Limitation
Signature based	It cannot warn firsthand attack signature or signature that to some extent varies from old attacks.
Anomaly based	The common defy for all anomaly-based intrusion detection systems is that it is difficult to take into account the data that provide all types of normal traffic behavior. As a result, genuine traffic can be classified as attack traffic which will cause a false positive. In order to bring down the false positive rate, a larger set of parameters is used to provide more accurate normal profiles, which may cause an increase in the computational overhead to detect attack.
Hybrid attack detection	Complexity and cost of implementation is very high to deploy in practice.
Third party detection	Economic factor, security related issues may occur.
Attack source/Path identification	It is not stress-free to trajectory IP traffic to its source as IP protocol is stateless in nature. The attacker can easily satire the source IP address field in the packets and send the packets to the victim without notice.
Filtering	These techniques cause a large number of false positives as it is always challenging to distinguish malicious packets from legitimate packets.

that use third-party detection are easily found among traceback mechanisms [54-57].

7.3. Response Techniques

The aim of appalling response techniques is to reduce the impact of the attack and let the attack causes the minimal damage to the victim. We have classified the response techniques as follows:

1) Attack Source/Path Identification: After detecting an attack ideally the attack traffic should be blocked at its source. Unfortunately, it is not easy to track IP traffic to its source as IP protocol is stateless in nature. The attacker can easily spoof the source IP address field in the packets and send the packets to the victim without notice. To address this limitation, several ideas have been proposed to support IP traceability [58]. Attack source identification mechanisms provide the victim with information about the identity and path taken by the machines that are responsible for performing the attack [59].

2) Filtering: Filtering techniques are used to filter out incoming traffic that has been characterized as malicious by the detection mechanism only. Though, it is difficult to distinguish rouge packets from the legitimate packets; therefore, thus techniques cause a high number of false positive.

3) Rate Throttling: Rate-throttling is a moderate response technique that imposes a rate throttle on the incoming traffic that has been characterized as malicious by the detection mechanism. It is usually deployed when the detection mechanism has a high level of false positives or cannot precisely characterize the malicious traffic [60-62].

4) Reconfiguration: Reconfiguration mechanisms [63] modify the topology of the victim or the intermediate network by either adding more resources to the victim or to isolate the attack machines.

7.4. DDoS Attack Tolerance and Mitigation Techniques

Attack tolerance and mitigation technique assumes that it is impossible to prevent or abort DDoS attack completely. Therefore, this technique try to minimizing the attack impact and focuses on providing optimal level of service as per quality of its service requirement to legitimate users while the service provider is still under attack.

This is not a comprehensive solution in any way; parallel and achieve their goals by providing sufficient assurance and gentle heal in terms of time to providers that the legitimate clients are being served. **Table 4** shows a comparative study of mitigation approaches. Attack tolerance and mitigation classifications are as following:

1) Over Provisioning of Resources

An abundance of resources, for example, high bandwidth link between victim machine, a pool of servers with load balancer and upstream routers are used to tolerate these attacks [64,65]

2) Router's Queue Management

Router's queue management techniques aim to reduce

Table 4. Mitigation approaches [35].

Mitigation Approach	Benefits	Limitations
IntServ	It provides service classes, which closely match the different application types described earlier and their requirements.	How to authorize and prioritize reservation requests, and what happens when signaling is not deployed end-to-end.
DiffServ	Scalability and flexibility is much better then IntServ.	DiffServ does not keep per flow state information. This makes it more difficult to support end-to-end QoS.
Class Based Queuing (CBQ)	Avoid bandwidth starvation problem.	Does not perform fair allocation of bandwidth, if the packet size is not same (variable size).
Proactive Server Roaming	Provide good response time in case of attack.	It has insignificant overhead in case of attack free situation.
Resource Accounting	Each flow gets a fair amount of resources.	Needed client puzzle software.
Resource Pricing	By employing different price and purchase function, architecture can achieve QoS.	System can be populated with fake request by the malicious user at low cost.
Pushback Approach	Upstream routers are not needed. Incremental deployment approach.	Great storage requirement.
Throttling	Helps to define an accurate and efficient packet filter.	At the time of implementation it is still hard to differentiate between legitimate traffic and malicious traffic.

the impact of attack or congestion simply without providing fairness between the traffic flows. Therefore, NPSR for these schemes is very low [66,67].

3) Router's Traffic Scheduling

Router's traffic scheduling algorithm reduces the congestion or attack impact and manages the flow of traffic along with it but they are too expensive in terms of delays and state monitoring [68-70].

4) Target Roaming

Active servers change their location within distributed homogeneous servers proactively to eliminate or chop DDoS attacks impact [71].

7.5. Detection of DDoS Using Hadoop

Hadoop [72], which was created by Doug Cutting, is the Apache Software Foundation open source and Java-based implementation of the MapReduce framework.

Hadoop provides the tools for processing vast amounts of data using the MapReduce framework and, implements the Hadoop Distributed File System (HDFS) [73,74]. It can be used to process vast amounts of data in parallel on large clusters in a reliable and fault-tolerant fashion. Yeonhee Lee and Youngseok Lee [75] presented two algorithm using MapReduce that detect the DDoS attack. There are two distinct algorithms that have been proposed:

1) Counter based method: This method relies on three key parameters: time interval which is the duration during which packets are to be analyzed, threshold which indicates frequency of requests and unbalance ratio which denotes the anomaly ratio of response per page requested between specific client and server.

The number of requests from a specific client to the specific URL within the same time duration is counted using the masked timestamp. The reduce function aggregates the number of URL requests, number of page requests, and total server responses between a client and a server. Finally values per server are aggregated by the algorithm. When the threshold is crossed and the unbalance ratio is higher than normal ratio from h, the clients are marked as attackers.

The key advantage of utilizing this algorithm is its low complexity. However the authors have indicated that the threshold value determination could be a key deciding factor in the implementation but do not offer any further information on how to determinate the value.

2) Access pattern based method: This method is based on a pattern which differentiates the normal traffic from DDoS traffic. This method requires more than two MapReduce jobs:

- First job gets the access sequence to the web page between a client and a web server and computes the spending time and the bytes count for each request of the URL;
- Second job finds infected hosts by comparing the access sequence and the spending time among clients trying to access the same server.

Limitation: This method used First In Fist Out scheduling in which ad-hoc queries are delayed.

3) Triangle Exception defense mechanism: This method is based on the fact that attacker machines uses Command and Control server to send the attacking command to Zombie Systems, which they use to attack the target web server. In triangle expectation defense mechanism network connection information from many routers is collected to analyze the triangle expectations. Once the Triangle expectations are computed, the zombie systems are identified and blocked. The sampling method used in this approach is called DOULION and is implemented with Map reduce.

8. Proposed Model

8.1. Scheduling in Hadoop

8.1.1. FIFO Scheduling

By default Hadoop uses First-in First-out (FIFO) scheduling. It can be implemented on a single node as well as cluster nodes. Its job can be assigned by sharing cluster resources. It uses the concept of Master-Slave. Job scheduling in Hadoop is performed by Job Tracker (master node). A Job tracker splits the job into number of chunks with some target and assigns these tasks to the Slave nodes (Map nodes). Map nodes compute the assigned task and resultant to be reported to Master node. Then master sorts the results and gives the output to the client. Hadoop monitors the progress of the task by using a progress score. Progress score of a task lies between 0 - 1.

Progress score is calculated by using the following formula:

$$PS = \begin{cases} M/N & \text{For Map Task} \\ 1/3 \times \left(K + M/N\right) & \text{For Reduce Task} \end{cases} \quad (1)$$

$$PS_{\text{avg}} = \sum_{i=1}^{R} PS[i] \Big/ T \quad (2)$$

$$\text{For task } T_i : PS[i] < PS_{\text{avg}} - 20\% \quad (3)$$

where *M*—Number of key/value pairs computed

N—Number of key/value pairs to be computed

K—Phase number of the reducer (possible *K* values are 0, 1, and 2)

T—Number of Tasks

PS—Progress Score

PS_{avg}—Progress Score Average

If the above inequality-(3) holds, then the task is considered as slow task and Job tracker copies the task and find the empty slave node and starts executing the task in that slave node. This process is called speculative execution. If the new task executes first then it kills the old slow task else the new task is killed.

The reducer computation has three phases: copy phase, sort phase and reduce phase. Copy phase is to copy results from the map nodes. Sort phase is to sort the results. Reduce phase is to reduce the results based on user specified key. In Hadoop each phase in reducer was given progress score of 0.33 (Hadoop assumes that all the phases in reducer take same amount of time) [75,76]. Now observed Loop Holes in default scheduling:

1) In Hadoop, the values of R1, R2, R3, M1, M2 are 0.33, 0.33. 0.34, 1 and 0 respectively. However in case of heterogeneous environment, R1, R2, R3, M1 and M2 should be dynamic as tasks running on different nodes.

2) Hadoop may launch backup tasks for wrong tasks as it always executes backup tasks for those tasks those PSs are less than $PS_{\text{avg}} - 20\%$.

3) Sometimes Hadoop may launch backup tasks for fast tasks [75].

8.1.2. SAMR Scheduling

To overcome the shortcoming of Hadoop scheduling SAMR scheduling was proposed. **Figure 5** shows the working of SAMR for counter based algorithm. After a job is committed, SAMR splits the job into map and reduce tasks, and assigns them to a series of nodes. In the interim, it reads historical information which stored on every node and updated it after every execution. In that case, SAMR adjusts time weight of each stage of map and reduce tasks according to the historical information respectively.

As a result, it gets the progress of each task accurately and finds which tasks need backup tasks. It identifies slow nodes and classfies them into the sets of slow nodes dynamically. SAMR launches the backup tasks on the basis of information of these slow nodes and ensures that the backup tasks are not slow tasks. It gets the final results of the tasks when either slow tasks or backup tasks finish first. Tentative results show that SAMR significantly decreases the time of execution up to 25% compared with Hadoop's scheduler [72,77].

8.2. Proposed Model

Even though there are many DDoS solutions proposed by different researchers, literature shows that there has been no effective way proposed to defend against DDoS attacks. To Detect DDoS, Counter based and Pattern Based Algorithm are quietly famous approach in Hadoop but still the major challenges are that they still have a lot of orientation towards batch processing and because of this ad hoc query jobs are delayed [77].

Hadoop is open source software based on scalability, distributed and reliability concept. It is best suited for large scale *i.e.* big data, provides optimum analyzed data by distributing big data into multiple chunks. It uses scheduling algorithms for MapReduce.

We already discuss the loopholes available in the existing scheme. Our aim is to proposed a model that uses SAMR Counter based algorithm that improve the efficiency as it reads historical information which stored on every node and updated it after every execution. This give more accurate Progress score and finds which task needs backup task.

This model inputs three parameters: time interval, threshold and unbalance ratio, which are stored in HDFS through packet loader. The packet collector receives IP packets from trace files on the disk, and writes them to HDFS. IP packets are stored in the binary format of libpcap. The threshold and unbalanced ratios for server are passed as parameters along with the timestamp. Job starts at the client and Job Tracker running SAMR scheduler splits the job into map and reduces tasks and

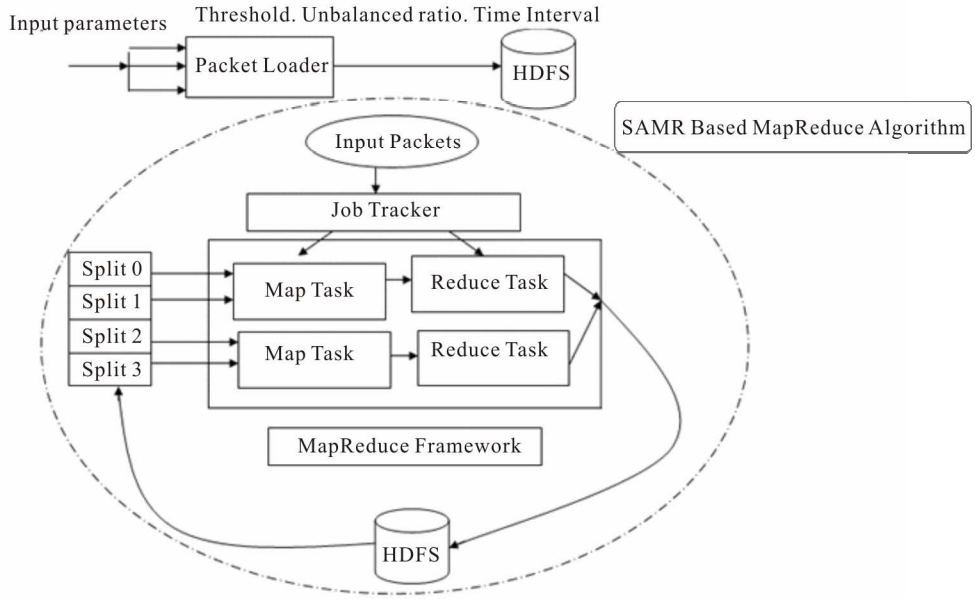

Figure 5. SAMR scheduling based map-reduce algorithm for counter-based DDOS detection.

assigns them to a series of nodes while doing thus it also reads historical information which is stored on every node and is updated after every execution. SAMR then adjusts time weight of each stage of map and reduce tasks as per the historical information respectively. Thus, it gets the progress scores of each task accurately and finds which of the tasks need backup tasks to run and also identifies the slow nodes and classifies them into the sets of slow nodes dynamically.

It gets the final results of the fine-grained tasks when either slow tasks or backup tasks finish first. The map task generates keys to classify the requests and response HTTP messages. Then, the reduce task summarizes the HTTP request messages and marks the abnormal traffic load by comparing it with the threshold. The map task generates keys to classify the requests and response HTTP messages. Then, the reduce task summarizes the HTTP request messages and marks the abnormal traffic load by comparing it with the threshold. The results are saved back to HDFS.

9. Conclusions and Future Work

This paper discusses the history the of DDoS attacks along with some major incidents to provide a better understanding and gravity of the problem. The paper includes latest techniques such as Hadoop along with other available techniques for prevention and detection of distributed denial of service attacks so that a comprehensive solution can be developed with several detection layers to trap the intrusion keeping in mind the limitations of these prevention and detection techniques.

The paper also discusses some of the recent development happened in the sphere of DDoS using Hadoop. Though this technique sounds promising, it can be further optimized. At last a proposed model is given which replace default scheduling via fair scheduler in Hadoop based algorithm to detect DDoS attack.

REFERENCES

[1] B. M. Leiner, V. G. Cerf, D. D. Clark, R. E. Kahn, L. Kleinrock, D. C. Lynch, J. Postel, L. G. Roberts and S. Wolff, "A Brief History of the Internet," 2000. http://www.isoc.org/internet/history/brief.shtml

[2] B. B. Gupta, R. C. Joshi and M. Misra, "Defending against Distributed Denial of Service Attacks: Issues and Challenges," *Information Security Journal*: *A Global Perspective*, Vol. 18, No. 5, 2009, pp. 224-247.

[3] C. Douligeris and A. Mitrokotsa "DDoS Attacks and Defense Mechanisms: Classification and State of the Art," *Elsevier Science Direct Computer Networks*, Vol. 44, No. 5, 2004, pp. 643-666. doi:10.1016/j.comnet.2003.10.003

[4] S. M. Specht and R. B. Lee, "Distributed Denial of Service: Taxonomies of Attacks, Tools, and Countermeasures," *Proceedings of the International Workshop on Security in Parallel and Distributed Systems*, San Francisco, 15-17 September 2004, pp. 543-550.

[5] A. Mishra, B. B. Gupta and R. C. Joshi, "A Comparative Study of Distributed Denial of Service Attacks, Intrusion Tolerance and Mitigation Techniques," *European Intelligence and Security Informatics Conference*, *EISIC* 2011, 12-14 September 2011, pp. 286, 289.

[6] T. Kitten, "DDoS: Lessons from Phase 2 Attacks," 2013.

http://www.bankinfosecurity.com/ddos-attacks-lessons-from-phase-2-a-5420/op-1

[7] A. ALmomani, T.-C. Wan, B. B. Gupta, A. Altaher, E. A. Lmomani and S. Ramadass, "A Survey of Phishing Email Filtering Techniques," *IEEE Communications Surveys & Tutorials*, Vol. PP, No. 99, 2013, pp. 1-21.

[8] S. Zargar, J. Joshi and D. Tipper, "A Survey of Defense Mechanisms against Distributed Denial of Service (DDoS) Flooding Attacks," *Communications Surveys & Tutorials, IEEE*, Vol. PP, No. 99, 2013, pp. 1-24. doi:10.1109/SURV.2013.031413.00127

[9] K. Zetter, "Lazy Hacker and Little Worm Set off Cyberwar Frenzy," 2009. http://www.wired.com/threatlevel/2009/07/mydoom/

[10] L. Greenemeier, "Estonian Attacks Raise Concern over Cyber "Nuclear Winter"," Information Week, 2007. http://www.informationweek.com/estonian-attacks-raise-concern-over-cybe/199701774

[11] J. Vijayan, "Mydoom Lesson: Take Proactive Steps to Prevent DDoS Attacks," 2004. http://www.computerworld.com/s/article/89932/Mydoom_lesson_Take_proactive_steps_to_prevent_DDoS_attacks?%20taxonomyId=017

[12] "Powerful Attack Cripples Internet," 2002. http://www.greenspun.com/bboard/q-and-a-fetch-msg.tcl msgid=00A7G7

[13] Yahoo on Trail of Site Hackers," Wired.com, 2000. http://www.wired.com/techbiz/media/news/2000/02/34221

[14] S. Garfinkel and G. Spafford, "Practical Internet and UNIX Security," O'Reilly Media, 1996

[15] "CERT Advisory: SYN Flooding and IP Spoofing Attacks," CERT® Coordination Center Software Engineering Institute, Carnegie Mellon, 2010. http://www.cert.org/advisories/CA-1996-21.html

[16] CERT, "Tech Tips: Denial of Service Attacks," CERT® Coordination Center Software Engineering Institute, Carnegie Mellon, 2010. http://www.cert.org/tech_tips/denial_of_service.html

[17] "Notable Hacks," PBS Frontline, 2010. http://www.pbs.org/wgbh/pages/frontline/shows/hackers/whoare/notable.html

[18] K. J. Houle, G. M. Weaver, N. Long and R. Thomas, "Trends in Denial of Service Attack Technology," CERT® Coordination Center, 2001.

[19] N. Schactman, "Wage Cyberwar against Hamas, Surrender Your PC," Wired: Danger Room Blog, 2009.

[20] P. Wilkinson, "Briton's Software a Surprise Weapon in Iran Cyberwar," Cable News Network, Atlanta, 2009.

[21] B. Martin, "Have Script, Will Destroy (Lessons in DoS)," 2000. http://attrition.org/~jericho/works/security/dos.html

[22] X. Wang and M. Reiter, "WRAPS: Denial-of-Service Defense through Web Referrals," *Proceedings of the 25th IEEE Symposium on Reliable Distributed Systems*, (*SRDS*'06), Leeds, 2-4 October 2006, pp. 51-60.

[23] R. Mackey, "'Iranian Cyber Army' Strikes Chinese Website," New York Times Lede Blog, 2011.

[24] D. Kravetz, "Anonymous Unfurls 'Operation Titstorm'," Wired Threat Level Blog, 2010.

[25] J. Nazario, "Politically Motivated Denial of Service Attacks," Arbor Networks, 2009.

[26] DDoS-for-Hire Service Is Legal and Even Lets FBI Peek in, Says a Guy with an Attorney," 2012. http://www.ddosdefense.net

[27] "Internet Creaks Following Cyber Attack on Spamhaus," 2013. http://www.cbronline.com/news/security/internet-slows-down-following-ddos-attack-on-spamhaus-280313

[28] T. Kitten, "2 More Banks Are DDoS Victims," 2012. http://www.bankinfosecurity.com/2-more-banks-are-ddos-victims-a-5298

[29] T. Kitten, "DDoS Strikes American Express," 2013. http://www.bankinfosecurity.com/american-express-a-5645

[30] "iMessage DDoS Attacks Foreshadow a Bigger Threat," 2013. http://soshitech.com/2013/04/01/imessage-ddos-attacks-foreshadow-a-bigger-threat/

[31] J. Kirk, "Mt. Gox under Largest DDoS Attack as Bitcoin Price Surges," 2013. http://www.computerworld.com/s/article/9238118/Mt._Gox_under_largest_DDoS_attack_as_bitcoin_price_surges

[32] "Mstream Distributed Denial of Service Tool (Zombie Detected) (DdosMstreamZombie)," 2013. http://www.iss.net/security_center/reference/vuln/ddos-mstream-zombie.htm

[33] N. McAllister, "GoDaddy Stopped by Massive DDoS Attack," 2012. http://www.theregister.co.uk/2012/09/10/godaddy_ddos_attack/

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

[35] B. B. Gupta, M. Misra and R. C. Joshi, "FVBA: A Combined Statistical Approach for Low Rate Degrading and High Bandwidth Disruptive DDoS Attacks Detection in ISP Domain," 16*th IEEE International Conference on Networks*, 12-14 December 2008, New Delhi, pp. 1-4.

[36] J. Lo, *et al*., "An IRC Tutorial," 1997. http://www.irchelp.org/irchelp/irctutorial.html#part1

[37] D. Dittrich, "The Tribe Flood Network Distributed Denial of Service Attack Tool," University of Washington, Seattle, 1999. http://staff.washington.edu/dittrich/misc/tfn.analysis.txt

[38] J. Barlow and W. Thrower, "TFN2K—An Analysis," Axent Security Team, 2000. http://security.royans.net/info/posts/bugtraq_ddos2.shtml

[39] D. Dittrich, "The Stacheldraht Distributed Denial of Service Attack Tool," University of Washington, Seattle, 1999. http://staff.washington.edu/dittrich/misc/stacheldraht.analysis.txt

[40] D. Dittrich, S. Dietrich and N. Long, "An Analysis of the 'Shaft' Distributed Denial of Device Tool," *USENIX Systems Administration Conference*, March 2000. http://www.soscholar.net/detail?paper_id=2bb7f2f9-2ed7-3422-78d2-e938aaaf44af

[41] F. Freiling, *et al*., "Botnet Tracking: Exploring a Root-Cause Methodology to Prevent Distributed Denial-of-Service Attacks," *Computer Security-ESORICS* 2005, Milan, 12-14 September 2005, pp. 319-335.

[42] Z. S. Zhu, G. H. Lu, Y. Chen, Z. Fu, P. Roberts and K. Han, "Botnet Research Survey," 32*nd Annual IEEE International Conference on Computer Software and Applications*, *COMPSAC*'08, Turku, 28 July-1 August 2008, pp. 967, 972.

[43] P. Negi, A. Mishra and B. B. Gupta, "Enhanced CBF Packet Filtering Method to Detect DDoS Attack in Cloud Computing Environment," *International Journal of Computer Science Issues*, Vol. 10, No. 1, 2013, pp 142-146.

[44] X. Geng and A. B. Whinston, "Defeating distributed denial of Service Attacks," *IEEE IT Professional*, Vol. 2, No. 4, 2000, pp. 36-42. doi:10.1109/6294.869381

[45] T. M. Gil and M. Poletto, "Multops: A Data-Structure for Bandwidth Attack Detection," *Proceedings of the* 10*th USENIX Security Symposium*, Washington DC, 2001, pp. 23-38.

[46] J. Li, J. Mirkovic, M. Wang, P. Reiher and L. Zhang, "SAVE: Source Address Validity Enforcement Protocol," 21*st Annual Joint Conference of the IEEE Computer and Communications Societies*, New York, 23-27 June 2002, pp. 1557-1566.

[47] B. Bencsath and I. Vajda, "Protection against DDoS Attacks Based on Traffic Level Measurements," *Proceedings of the Western Simulation Multi Conference*, San Diego, 2004, pp. 22-28.

[48] B. B. Gupta, M. Misra and R. C. Joshi, "An ISP Level Solution to Combat DDoS Attacks Using Combined Statistical Based Approach," *International Journal of Information Assurance and Security*, Vol. 3, No. 2, 2008, pp. 102-110.

[49] Y. Chen, K. Hwang and W. Ku, "Collaborative Detection of DDoS Attacks over Multiple Network Domains," *IEEE Transaction on Parallel and Distributed Systems*, Vol. 18, No. 12, 2007, pp. 1649-1662.

[50] L. Feinstein, D. Schnackenberg, R. Balupari and D. Kindred, "Statistical Approaches to DDoS Attack Detection and Response," *Proceedings of DARPA Information Survivability Conference and Exposition*, Washington DC, 22-24 April 2003, pp. 303-314.

[51] A. Lakhina, M. Crovella and C. Diot, "Mining Anomalies Using Traffic Feature Distributions," ACM SIGCOMM Computer Communication Review, Vol. 35, No. 4, 2005, pp. 217-228. doi:10.1145/1090191.1080118

[52] K. Hwang, M. Cai, Y. Chen and M. Qin, "Hybrid Intrusion Detection with Weighted Signature Generation over Anomalous Internet Episodes," *IEEE Transaction on Dependable and Secure Computing*, Vol. 4, No. 1, 2007, 41-55. doi:10.1109/TDSC.2007.9

[53] J. Mirkovic and P. Reiher, "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms," *ACM SIGCOMM Computer Communications Review*, Vol. 34, No. 2, 2004, pp. 39-53. doi:10.1145/997150.997156

[54] S. Savage, D. Wetherall, A. Karlin and T. Anderson, "Practical Network Support for IP Traceback," *Proceedings of ACM SIGCOMM*, Stockholm, 2000, pp. 295-306.

[55] A. C. Snoeren, C. Partridge, L. A. Sanchez, C. E. Jones, F. Tchakountio, S. T. Kent and W. T. Strayer, "Hash-Based IP Traceback," *Proceedings of ACM SIGCOMM*, San Diego, 2001, pp. 3-14.

[56] S. Bellovin, M. Leech and T. Taylor, "ICMP Traceback Messages," 2001. Internet draft: draft-ietf-itrace-01.txt

[57] D. Dean, M. Franklin and A. Stubblefield, "An Algebraic Approach to IP Traceback," *ACM Transactions on Information and System Security*, Vol. 5, No. 2, 2002, pp. 119-137. doi:10.1145/505586.505588

[58] Y. Manzano, "Tracing the Development of Denial of Service Attacks: A Corporate Analogy," 2003. http://www.acm.org/crossroads/xrds10-1/tracingDOS.html

[59] A. Belenky and N. Ansari, "IP Traceback with Deterministic Packet Marking," *IEEE Communication Letter*, Vol. 7, No. 4, 2003, pp. 162-164. doi:10.1109/LCOMM.2003.811200

[60] C. Papadopoulos, R. Lindell, J. Mehringer, A. Hussain, and R. Govindan, "COSSACK: Coordinated Suppression of Simultaneous Attacks," *Proceedings of the DARPA Information Survivability Conference and Exposition*, Vol. 2, Washington DC, 22-24 April 2003, pp. 2-13. doi:10.1109/DISCEX.2003.1194868

[61] J. Mirkovic, G. Prier and P. Reiher, "Attacking DDoS at the Source," 10*th IEEE International Conference on Network Protocols*, Paris, 12-15 November 2002, pp. 312-321.

[62] S. Floyd, S. Bellovin, J. Loannidis, K. Kompella, R. Mahajan and V. Paxson, "Pushback Messages for Controlling Aggregates in the Network," 2001. draft-floyd-pushback-messages-00.txt

[63] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek and R. Morris, "Resilient Overlay Networks," In Proceedings of 18th ACM SOSP, Banff, Canada, 2001, pp. 131-145.

[64] R. B. Lee, "Taxonomies of Distributed Denial of Service Networks, Attacks, Tools and Countermeasures," Princeton University, Princeton, 2003. http://www.princeton.edu/ee/

[65] R. Bush, D. Karrenberg, M. Kosters and R. Plzak, "Root Name Server Operational Requirements," RFC Editor, United States, BCP 40, RFC 2870, June 2000.

[66] S. Floyd and V. Jacobon, "Random Early Detection Gateways for Congestion Avoidance," *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, 1993, pp. 397-413. doi:10.1109/90.251892

[67] S. Floyd and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet," *IEEE/ACM Transactions on Networking*, Vol. 7, No. 4, 1999, pp. 458-472. doi:10.1109/90.793002

[68] A. Demers, S. Keshav and S. Shenker, "Analysis and Simulation of a Fair Queuing Algorithm," *Journal of Internetworking Research and Experience*, Vol. 1, No. 1,

1990, pp. 3-26.

[69] P. Mckenny, "Stochastic Fairness Queuing," 9*th Annual Joint Conference of the IEEE Computer and Communication Societies*, *the Multiple Facets of Integration*, Piscataway, 3-7 June 1990, pp. 733-740.

[70] A. Mankin and K. Ramakrishnan, "Gateway Congestion Control Survey," 1991. http://www.rfc-editor.org/rfc.html

[71] S. M. Khattab, C. Sangpachatanaruk, R. Melhem, D. Mosse and T. Znati, "Proactive Server Roaming for Mitigating Denial of Service Attacks," 1*st International Conference on International Technology*: *Research and Education*, Newark, 2003, pp. 500-504.

[72] Apache Hadoop. http://hadoop.apache.org/

[73] S. Ghemawat, H. Gobio and S.-T. Leung, The Google File System," *ACM SIGOPS Operating Systems Review*, Vol. 37, No. 5, 2003, pp. 29-43.

[74] K. V. Shvachko, "HDFS Scalability: The Limits to Growth," *USENIX*, Vol. 35, No. 2, 2010, pp. 6-16.

[75] Y. Lee, W. Kang and Y. Lee, "A Hadoop-Based Packet Trace Processing Tool," 3*rd International Conference on Traffic Monitoring and Analysis*, Vienna, 27 April 2011, pp. 51-63. doi:10.1007/978-3-642-20305-3_5

[76] T. White, "Hadoop: The Definitive Guide," O'Reilly Media, Yahoo! Press, New York, 2009.

[77] Q. Chen, D. Q. Zhang, M. Y. Guo, Q. N. Deng and S. Guo, "Samr: A Self-Adaptive Mapreduce Scheduling Algorithm in Heterogeneous Environment," *International Conference on Computer and Information Technology*, Bradford, 29 June-1 July 2010, pp. 2736-2743.

