Shuffling Based Mechanism for DDoS Prevention on Cloud Environment

Thesis submitted in partial fulfillment

of the requirements for the degree of

Master of Technology

in

Computer Science and Engineering

by

Sidharth Sharma

(Roll No: 213CS2172)

under the guidance of

Prof. Sanjay Kumar Jena

Department of Computer Science and Engineering
National Institute of Technology, Rourkela
Rourkela-769 008, Odisha, India
May, 2015.

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Declaration

I certify that

- I have complied with all the benchmark and criteria set by NIT Rourkela Ethical code of conduct.
- The work done in this project is carried out by me under the supervision of my mentor.
- This project has not been submitted to any other institute other than NIT Rourkela.
- I have given due credit and references for any figure, data, table which was being used to carry out this project.

Sidharth Sharma
National Institute of Technology
Rourkela-769008

Place: NIT,Rourkela-769008
Date: 01 - 06 - 2015

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India.

Certificate

This is to certify that the work in the thesis entitled ***”Shuffling based approach for DDoS prevention on cloud environment”*** submitted by ***Sidharth Sharma*** is a record of an original research work carried out by him under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of Master of Technology in Computer Science and Engineering, National Institute of Technology, Rourkela. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Prof. Sanjay Kumar Jena
Professor
Department of CSE
National Institute of Technology
Rourkela-769008

Place: NIT,Rourkela-769008
Date: 01-06-2015

Acknowledgment

First of all, I would like to express my deep sense of respect and gratitude towards my supervisor Prof. Sanjay Kumar Jena, who has been the guiding force behind this work. I want to thank him for introducing me to the field information security and giving me the opportunity to work under him. His undivided faith in this topic and ability to bring out the best of analytical and practical skills in people has been invaluable in tough periods. Without his invaluable advice and assistance it would not have been possible for me to complete this thesis. I am greatly indebted to him for his constant encouragement and invaluable advice in every aspect of my academic life. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

I wish to thank all faculty members and secretarial staff of the CSE Department for their sympathetic cooperation.

During my studies at N.I.T. Rourkela, I made many friends. I would like to thank them all, for all the great moments I had with them. I would also like to thank Santosh Kumar Sahoo (Phd. scholar) for helping me during the work.

When I look back at my accomplishments in life, I can see a clear trace of my family's concerns and devotion everywhere. My dearest mother, whom I owe everything I have achieved and whatever I have become; my beloved father, for always believing in me and inspiring me to dream big even at the toughest moments of my life; who were always my silent support during all the hardships of this endeavor and beyond.

Sidharth Sharma

Abstract

Cloud Computing has evolved as a new paradigm in which users can use on-demand services, according to their needs. However, security concerns are primary obstacles to a wider adoption of clouds. Newly born concepts that clouds introduced, such as multi-tenancy, resource sharing and outsourcing, create new challenges for the security research. DDoS (Distributed Denial of service) attack is the biggest threat to the cloud since it affects the availability of services. There are a lot of techniques proposed by various researchers to prevent DDoS attacks on a cloud infrastructure. We are using a Shuffling Based approach for preventing DDoS in the cloud environment. This approach is reactive and uses the resource elasticity of the cloud. The aim of this technique is to save the maximum number of benign clients from the attack through shuffling. For assignment of clients to the replica servers, we are using a greedy algorithm. Every time we call this algorithm, we estimate the number of malicious clients using a proposed random function for that round of shuffle. We have shown that we can save a desired percentage of benign clients from the ongoing attacks after some shuffles. To detect the attack on each server, a detector is deployed that uses an entropy-based approach for detecting DDoS. A significant deviation in entropy represents the DDoS attack. We have also performed some tests to select the suitable attributes for entropy-based DDoS detection in different type of DDoS attacks. So in our work we have worked on both detection and prevention of DDoS on cloud infrastructure.

Contents

List of Figures

Chapter 1

Introduction

Cloud Computing

Cloud Security

DDoS Attack

Motivation

Research Contribution

Organization of Thesis

CHAPTER 1

Introduction

This chapter describes the overview of the thesis. It covers cloud computing, security issues in cloud computing and distributed denial of service attack (DDoS) which are the pillars of this thesis.

1.1 Cloud Computing

Cloud computing is one of the emerging area in the information technology field. Cloud providers provide on demand services to customers over the internet and charge them according to their usage. Cloud computing shares distributed resources over the network. Cloud uses technologies like virtualization, multi-tenancy for its operation. Users enjoy the services on the cloud without the headache of knowing the background technology and controlling it. Cost effectiveness is the primary benefit of the adoption of cloud. NIST (National Institute of Standards and Technology) defines cloud computing as [10] — "Cloud Computing enables ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g. Networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"

Cloud computing have five essential characteristics-

1. On-demand self-service-

In cloud computing, a customer can get all the required services without requirement of any human interaction with the cloud service provider.

2. Ubiquitous network Access-

All the services are available over the network and can be accessed through standard mechanisms over heterogeneous platforms, for example, computers, tablets, mobile phones. High- Bandwidth connections must be available to connect cloud services.

3. Resource Pooling-

Resources of cloud service providers are pooled to serve multiple consumers using a multi-tenant model, in which resources are dynamically assigned to the different customers according to their usage. Service providers must have enough number of resources to meet customer's need. The resources can be placed at many geographical locations and assigned as virtual components of computation as required.

4. Rapid Elasticity-

Capabilities can be dynamically allocated and released. According to the need of the customer, services are assigned to them automatically. Whenever demand increases more number of resources assigned to serve the client. The consumer sees it as an unlimited pool of resources.

5. Measured Services-

Services that are used by the customers are measured by providers. Resource usage is monitored, and a report is sent to both of the parties involved. It makes the charging system more transparent and accessible.

Cloud Services can be classified under three service models-

1. Software as a Service(SaaS)-

In this model, a subscriber can use a software that is running on Cloud Infrastructure. Applications are accessible from client resources such as web browsers. Applications are hosted on the service provider's platform. In this model, consumers do not have direct control over the underlying infrastructure like networks, servers, operating systems. Pay-per-use and subscription agreement model can be used for billing SaaS customers. True SaaS provides multi-tenant application infrastructure.

2. Platform as a Service(PaaS)-

In this service model subscribers can deploy their applications on the cloud environment without installing tools on their machines. It provides operating system support and software development tools to users. Using PaaS, developers can create their applications without installing software building tools. The consumer does not have control over networks, servers, operating systems and storage but have control over application deployment. PaaS can provide complete development cycle and development resources to developers, so it is very beneficial for entry level developers. PaaS vendors are less because their target users are less in comparison to SaaS, but some of the SaaS vendors start offering PaaS services as a logical extension of their services for example-Amazon Web Services.

3. Infrastructure as a Service(IaaS)-

This model shows the complete difference between traditional IT model and cloud computing. In this model, customers are provided with essential resources like processing, storage, networks. Customers can deploy their operating systems and applications on virtual machines equipped with these resources. The consumer has full control over operating systems, storage, deployed applications. IT organizations, having high expenses of purchasing dedicated hardware. Employing IaaS provide scalability, resource elasticity, and maintenance better than the typical IT infrastructure.

There are four types of delivery models present in the cloud-

1. Public Cloud-

This Cloud Model exists ‘Externally’ with very few restrictions for public use. It means this infrastructure is available for public use across the world on the basis of ‘pay to use’; this ‘public’ can be individuals, corporations, academic organizations or government agencies. Public Cloud infrastructure exists on the premise of cloud service provider. Shared Infrastructure, dynamic licensing and remote hosting are features of public cloud infrastructure. Amazon web services, Google App Engine, Microsoft Windows Azure are the examples of public clouds. This infrastructure is very suitable for small scale organizations. Security is a primary concern in public cloud adoption as many rival organizations may use it at the same time. Their data kept on the same servers so, sometimes malicious users may harm the data of others by using some tricks. Depending upon the needs of organization configuration requirements, SLAs can be customized.

2. Private Cloud-

This Cloud Model explicitly builds for a single organization. They are also referred as 'internal Clouds' as in most of the cases they exist on the premise of the user. But, in some instances it can be off-premise too. The private cloud typically hosted within the customer's organization. Unlike public cloud infrastructure is not shared among users in a private cloud. Security is considered to be more up to date. In private cloud infrastructure is owned by the organization, so it has full control on how applications are deployed on it. Private Cloud also considers being more flexible and easily extensible for the future upgrades.

3. Community Cloud-

This Cloud model is created for a specific community of organizations having same interests like security,implementation,compliance and policy. It may be owned by one or more organizations or a third party. It may exist On premise or Off premise.

4. Hybrid Cloud-

Simply Hybrid Cloud is combination of Public/Private Cloud models. NIST defines it as [10]-"A composition of two or more clouds(private, community or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (e.g. cloud bursting for load balancing between cloud)". A typical example of a hybrid cloud is an organization that is running non-critical applications on a public cloud and critical applications on private cloud infrastructure. An important feature of Hybrid Cloud is "Cloudburst". Cloudburst generally refers to a dynamic deployment model in which application runs on the private cloud, which exist on premise, but as soon as demand increase it can also be shifted to the public cloud outside premise of an organization.

1.2 Cloud Security

Even though there are many benefits of adoption of clouds, but there are many security issues related to it.

Here we are classifying security issues under following categories- [15]

The Security Standards category covers security policies, security agreement between clients and cloud service providers to ensure the required level of security in the cloud. This category deals with auditing, service level agreements and other agreements between two parties.

The Network category deals with the security of networks through which clients are connected to the cloud. It covers request from the client's browser, connection establishment and cryptographic issues.

The Access Control category ensures that only authorized users get access rights on the cloud resources. A proper identification and authorization mechanism need to be set-up on the server side.

The Cloud Infrastructure category deals with all the threats to the infrastructure of the cloud. It consists issues related to virtualization, IaaS, PaaS and SaaS. This category also takes care of issues related to virtual machines also.

The Data category deals with data integrity and confidentiality issues with the customer's data on cloud environment.

1.2.1 Known Attacks on Cloud Infrastructure

1. Denial of Services Attack-

This attack is on the availability of the cloud architecture. The majority of the attacks in cloud computing are the denial of service attacks (DoS); these are flooding attacks which floods the target server with unnecessary packets. DDoS attack is an extension of the standard DoS attack in which many users across the globe floods the server to make it unavailable for the legitimate clients. The cloud users initiate requests in XML, then send requests over HTTP protocol and usually build their system interface through REST protocols for example Microsoft Azure and Amazon EC2. DoS attacks are easier to implement and very difficult for security personnel to countermeasure. An XML-based and HTTP-based DDoS attacks are very common attacks on the cloud because the cloud operations heavily use these protocols.

2. Theft of Service Attack-

The Theft of Service attack utilizes vulnerabilities of the scheduler of some hypervisors. In this attack scheduling mechanism used by the hypervisor fails to detect usage of processing by some virtual machines. This allows malicious clients to use services for the expenses of other clients. This attack is relevant to the public client in which customers are charged according to the amount of time their virtual machine runs. In the theft of Service attack, the attacker makes sure that its process never scheduled.

3. Malware Injection Attack-

In malware injection attack, attacker uploads a copy of the victim's service instance to the cloud, aiming some of the service requests of client process within that malicious instance. By this attacker can get access over some of the client data. Critical information leakage and unauthorized access are the primary motives of this attack. The challenge is to find out the particular server on which attacker has uploaded the manipulated copy of the victim's malicious instance.

4. Phishing Attack-

Phishing attacks are attempts for accessing critical information through some social engineering technique. The attacker sends a link to web pages or instant messages to the victim who appears to be genuine, leading to legitimate sites like bank login page but take the victim to the fake locations. Through this activity, attacker gets login information of the victim. Hijacking of the accounts in the cloud by using some social networking techniques is also a variant of phishing attack.

5. Botnet Attacks-

In stepping-stone attack, Attacker maliciously acquires access to many of the nodes and installs attack tools on them. Attacker achieves their goals like spying, DoS through these stepping stones. An attacker can also hide their locations and identities so that they never be traced. Stepping stones attack the cloud server concurrently this set of nodes called as botnets. In recent years, botnet attacks have been reported in Amazon EC2, Google App Engine.

6. VM Rollback Attack-

The virtualization layer is most vulnerable to attack in a cloud environment. Hypervisors can suspend any VM's execution and take a snapshot of it. After some time, it can reload that snapshot and resume the operation this feature is very useful in fault tolerance and VM maintenance. On the other hand, this process also opens doors for VM rollback attack. In this attack, an attacker runs the snapshot of the victim without the knowledge of the victim and clears the history after running it. This attack is useful in running a Brute force attack to get login password in case some restrictions on the number of attempts for example block the user after 3 incorrect login attempts, the attacker will rollback VM to its initial state after each attempt. It can also clear the counter and run the attack unless it gets the login password.

7. Cross VM Side-Channel Attacks-

Side channel attack requires attacker's VM to be on the same physical server on which client's VM resides. Cache based side channel attacks are examples of this type of attack. All virtual machines on the same physical server share resources like cache. Using timing-based attack malicious VM can analyze which of the possible keys is utilized by the victim VM to encrypt its connections. Many researchers proved that AES and DES keys could easily be guessed by using cache-based side channel attacks. Timing channels are especially hard to detect since they leave no traces.

8. Audio Steganography Attacks-

The audio steganography attack is an attack on the cloud storage systems. In this attack, users hide their secret data in regular audio files. In this attack, users can protect their secret files in the media files and send them which looked like regular files.

1.3 Distributed Denial of Service Attacks

Distributed Denial of service attack possesses the biggest threat to Internet services. The attacker creates a botnet of the internet connected computers(each of them has been maliciously taken over through malware like Trojan horses) and attack through them, so it is a hard to locate attackers and block them. This makes research to detect and mitigate DDoS more attractive. Attackers are finding new techniques to perform attacks day by day so as researchers are using new ideas to counter them.The architecture of DDoS is shown in Fig.1.1.

1.3.1 Types of DDoS

We can classify DDoS under following categories-

1. TCP-SYN flood attack-

This class of DDoS attack represents 90% DDoS attacks. This attack exploits the vulnerability of TCP connection establishment. A TCP connection always starts with a three way handshake between clients and server. In the first phase, an empty TCP packet is sent by the client with a SYN flag raised. This packet asks the server to open a new connection and allocate resources for the client. In the second phase, the server sends an acknowledgment to the client along with SYN+ACK flag raised, and in third phase client sends an ACK to the server with the ACK flag raised. In

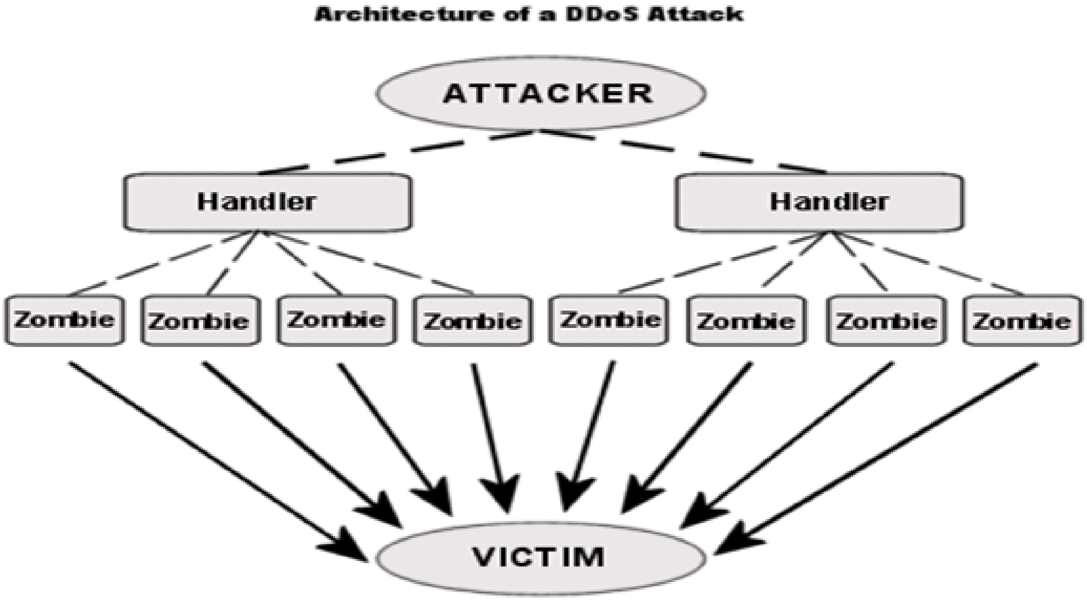

Figure 1.1: DDoS Attack

SYN flood attack, the attacker exploits this mechanism by quickly opening new connections to a remote server until the maximum number of open connections to that server is reached. In this situation, the legitimate clients are no longer able to open any new connections. This scenario makes the server unavailable for legitimate clients. In other words, SYN flooding occurs when the attacker sends an enormous amount of packets to the server, but does not complete the process of the three-way handshake. Then the server waits to complete the process. If no ACK packet is received by the server within a specified time duration, connection will be timed out and server will release the allocated resources since SYN packets are flooded so every time server waste lot of resources for these fake connections. Now a day, attackers send these SYN packets from spoofed IP addresses. Thus, the server is unable to reply these requests, which keeps resources busy.

2. UDP flood attack-

In UDP Connections hosts need not establish a valid connection to the server (no need of handshaking, like TCP) so a massive volume of traffic can be sent over UDP channels to any server. This means that UDP floods are highly effective and could be executed with fewer resources. In this type of attack attacker sends large number of UDP Packets to random ports on server machine usually from spoofed

IP addresses. The server checks whether any application is listening on these ports or not. If not, then it replies to ICMP Destination Unreachable packets. In case of flooding server needs to send a lot of ICMP Destination Unreachable packets which consumes server's resources and make it unavailable for legitimate users. A variant of the UDP flood attack is Fraggle attack that is similar to Smurf attack, the difference is that the attacker sends UDP traffic to port 7 (echo) and 19(chargen) to an IP broadcast address with victim's spoofed IP address.

3. Smurf attack-

In Smurf attack (ICMP flooding attack), the attacker spoofs the IP address of the victim and sends a large number of ICMP echo request packets to the broadcast address of his network. When hosts on this network receive that ICMP echo request packets they reply victim with ICMP echo reply packets, which floods the victim (server) and make it unavailable for legitimate users.

4. HTTP flooding attack-

HTTP flooding attack is a high rate flooding attack on the application layer of the TCP/IP stack. In this type of attack, the attacker sends a large number of legitimate type HTTP requests(basically GET and POST) to the target server requesting web-pages, files. The server starts to serve these requests and waste lot of resources for them (processing and memory) and unable to serve the request of actual legitimate clients.

1.4 Motivation

From the above section, it is clear that DDoS is biggest threat for any internet technology that also includes cloud environments. A research estimated that the average cost of one minute of downtime due to a DDoS attack is $22,000. Since, most of the organizations are moving to the cloud, security of cloud environment is a significant concern. During studies related to attacks on the cloud, we found DDoS handling is very crucial for proper operation of the cloud. So we have decided to work in this field. Primary motive of DDoS attacker is to affect availability of services, if somehow we can maintain a required level of services for innocent customers our motive of mitigating the effect of DDoS is fulfilled.

1.5 Contributions

Our work is about DDoS detection and prevention in cloud environment.

- Modification in greedy algorithm for client assignment in shuffling based approach.
- Proposed a random function for estimation of malicious clients which is used by greedy algorithm.
- Incorporated detection and shuffling based prevention techniques for cloud environment.
- Proper selection of attributes for entropy based detection of DDoS in case of different types of DDoS attack is explained.

1.6 Organization of Thesis

The rest of thesis is organized as follows:

Chapter 2:This chapter describes about DDoS detection strategy.

Chapter 3:This chapter explains about Shuffling based mechanism for DDoS prevention on cloud.

Chapter 4:This chapter elaborates simulation result and implementation.

Chapter 5:This chapter describes conclusions and future work.

Chapter 2

Detection of DDoS Attack

Literature Review

Entropy Based Detection

Summary

CHAPTER 2

Detection of DDoS Attack

Detection of distributed denial of service attack is extremely important part of DDoS handling. This chapter explains about various strategies proposed earlier for DDoS detection. We have used entropy-based detection that is explained deeply in this chapter.

2.1 Literature Review

Since the evolution of DDoS attacks different type of detection mechanism being proposed. Cabrera et al. [5] used network management systems that use MIB (Management Information Base) variables to detect precursors of attack, change in these MIB variables during attack used as a technique to detect attacks. This method was statistical in nature. Jeong et al. [11] stated that during denial of service attack very few IP addresses appear which are very less in comparison to the appearance during flash events they also compared these two events on various parameters. Lee et al. [16] proposed a unique path fingerprint scheme that represents the route of the IP packet has traversed. For each client, they create an entry for corresponding fingerprint if a spoofed packet is coming from some other route it has been discarded. Liao et al. [18] used a K- nearest neighbor classifier to classify patterns into normal or intrusive classes. They have tested with the 1998 DARPA BSM dataset. Gavrillis et al. [9] presented an approach using radial basis function neural network detector for DDoS attacks. A small number of statistical descriptors were used to distribute behavior of DDoS and classification is achieved using RBF-NN. We have analyzed entropy-based approach to detect DDoS. Many authors have worked previously on entropy based approach to detect DDoS. [4], [13], [17], [23]

2.2 Entropy Based Detection

Entropy is a well-known and valuable concept in information theory. It is the measure of uncertainty associated with a random variable and describes the degree of dispersal or concentration of a distribution. It was introduced by Claude E. Shannon in ”A Mathematical Theory of communication”, 1948. Entropy can be well utilized in the detection of DDoS because after analyzing DDoS attacks, researchers examined that Higher Volume of traffic, incomplete Connections and flooding of packets are characteristics of the attack. Therefore, entropy could be used to calculate the distribution randomness of the packet attributes. These attributes could be Source IP, Destination IP, Source Port, Destination Port, Length, Protocol, Flags. We can check the entropy of all useful attributes of the packet. If entropy is high, it means distribution is random. Entropy is calculated by examining a series of packets, refer as the window. If a window consists N packets, then entropy could vary from 0 to log N. Entropy is zero when all the values of the distribution are same, it is highest when all the values are different. If there are N elements in a window, then Entropy of random variable X is defined as H(X),where a random variable X is taking values from $x_1, x_2, x_3....x_N$ and probability of occurrence of these values are $p_1, p_2, p_3......p_N$ respectively.

$$H(X) = \sum_{i=1}^{N} p_i log_2 p_i \tag{2.1}$$

where p_i = (number of times X having value x_i)/(size of the window)

Under normal network conditions, the entropy value of the particular attribute may deviate up to some extent, but in case of DDoS attack, it will have remarkable fluctuations. Normalized entropy value can be used in order to set the threshold and raising an alert. Normalized entropy is used to find out the overall probability distribution in packet window W.

Normalized entropy = H(X) / $log_2 N_0$

Where N_0 is the number of distinct X values in the given window, H(X) is the entropy of X in window W. If Normalized Entropy < Threshold value, mark window as suspected. An alert can be raised if we get some continuous suspected windows.

2.3 Selection of Attributes for the Entropy Based Detection of DDoS

The effect on the entropy of all the useful packet attributes during DDoS attack is analyzed and their usefulness is also tested against famous types of distributed denial of service attack. During analysis, the proper choice of attributes one should make to get a better threshold is explained.

DDoS attack may affect entropy values of different network attributes. These values vary from different types of DDoS attack. Experiments were performed on the different types of DDoS attack and usefulness of various packet attributes for entropy based detection is analyzed. For experiments, NUST datasets for TCP-SYN flood, UDP flood and Smurf attacks [3] were used. Attack and non-attacked packets are marked in this dataset. We have preprocessed the dataset and took 100000 non attack packets from each dataset(TCP-SYN flood, UDP flood and Smurf attacks dataset). In that, 10,000 attack packets from packet count 10000 to 20000 were mixed. We have examined entropy values for all significant attributes of the packet header. Major categories of DDoS attacks and effect on entropy of different attributes is explained below-

2.3.1 TCP-SYN Flood Attack

TCP-SYN flood attack exploits the weakness of TCP connection establishment. On the entropy values of every important attribute of packet header, we analyzed the effect of DDoS.

1. Source IP and Destination IP address Entropy

In case of TCP-SYN flood attack attacker may use a random population of spoofed IP addresses. So, there may be cases where the entropy of Source IP can't be too effective for detecting TCP-SYN flooding. From Fig.2.1, it is clear that we can't set a proper threshold when the source IP population is random. Destination IP of these TCP-SYN attack packets are normally the victim's IP; all the spoofed Source

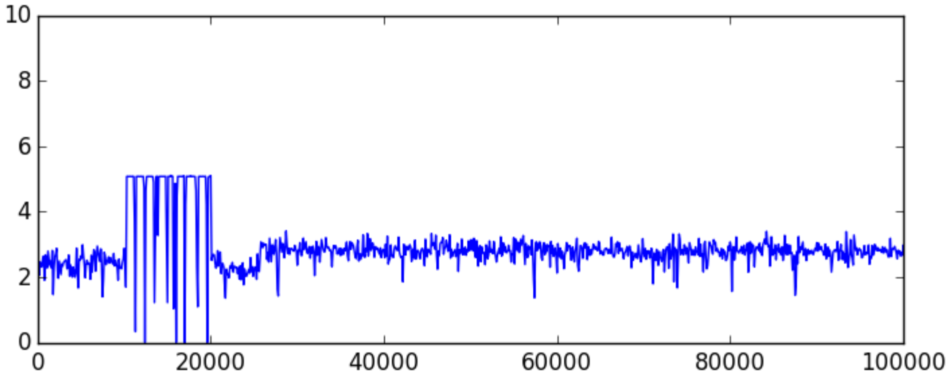

Figure 2.1: Source IP entropy during TCP-SYN attack

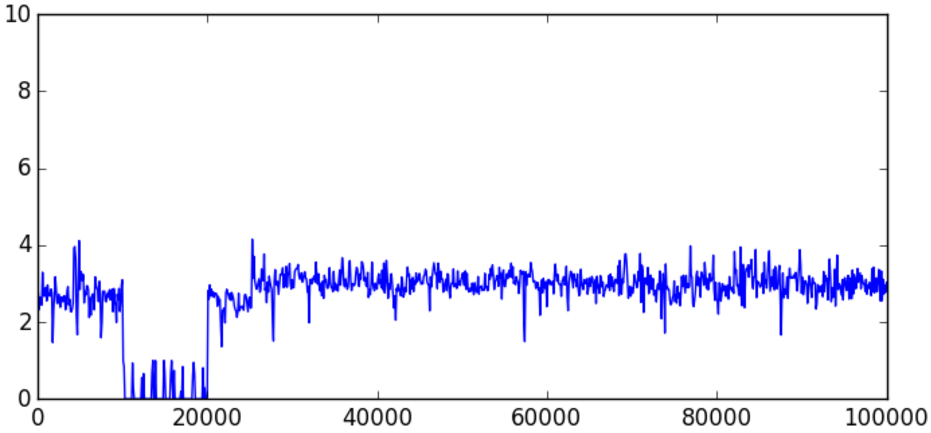

Figure 2.2: Destination IP entropy during TCP-SYN attack

IP packets have the victim's IP as their destination IP. Hence, we get repetition in this field values. Because of this entropy typically decrease during the attacks as shown in Fig.2.2.

2. Source Port and Destination Port Entropy

In the current attack scenario, attackers use random source Port numbers in attack packets. That is the main reason why the entropy of Source Port is not heavily affected during the attack as in Fig.2.3. The scenario remains true for destination Port entropy because attackers are intelligent enough to attack random ports on the victim machine as in Fig.2.4, hence, no variations in entropy here. But, sometimes it may be the case that they are targeting only few ports of victim machine that will decrease the randomness of Destination Port values and decrease the entropy during

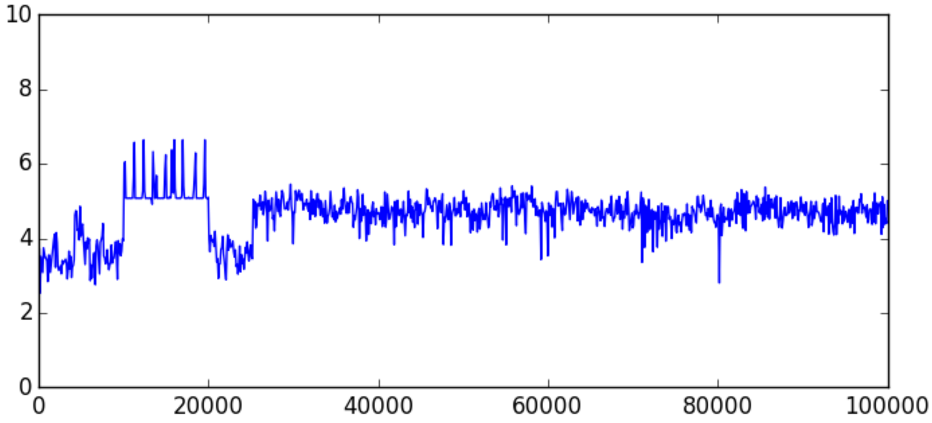

Figure 2.3: Source port entropy during TCP-SYN attack

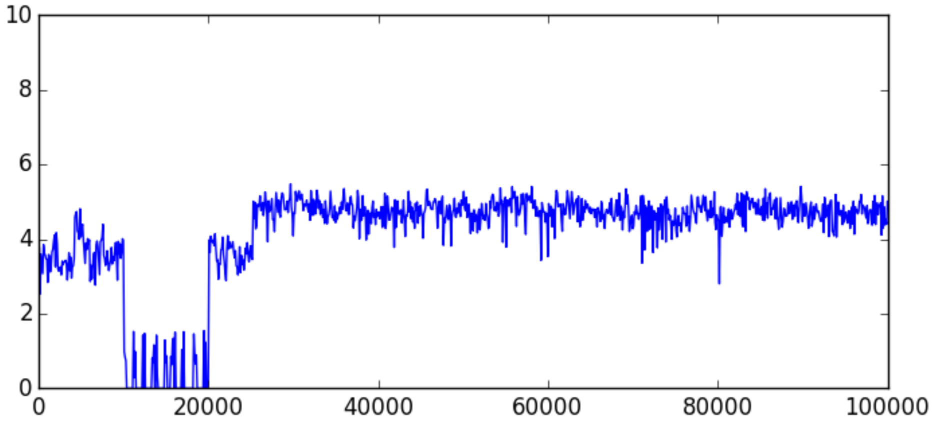

Figure 2.4: Destination port entropy during TCP-SYN attack

the attack as well.

3. Flag, Protocol and Length Entropy

In case of TCP-SYN attack packets only SYN flag raised. So, whenever attack traffic floods, packets with only SYN flag raised shall flow in the network. This will decrease the entropy of the flag field during attack scenario as in Fig.2.5. It is obvious that TCP-SYN attack packets use the TCP protocol in the protocol field of the header. During an attack, we get a large proportion of packets with protocol field TCP, this will decrease the entropy of the protocol field during the attack shown in Fig.2.6, But in most of the connections most of the packets use TCP protocol so this measure may not be that useful in those cases. Length is a very useful metric to judge entropy variations during the TCP - SYN attack. The main reason behind

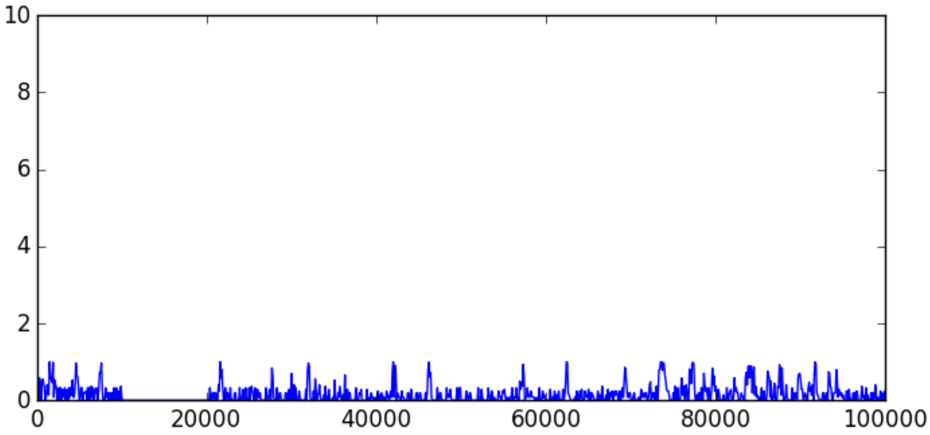

Figure 2.5: Flag field entropy during TCP-SYN attack

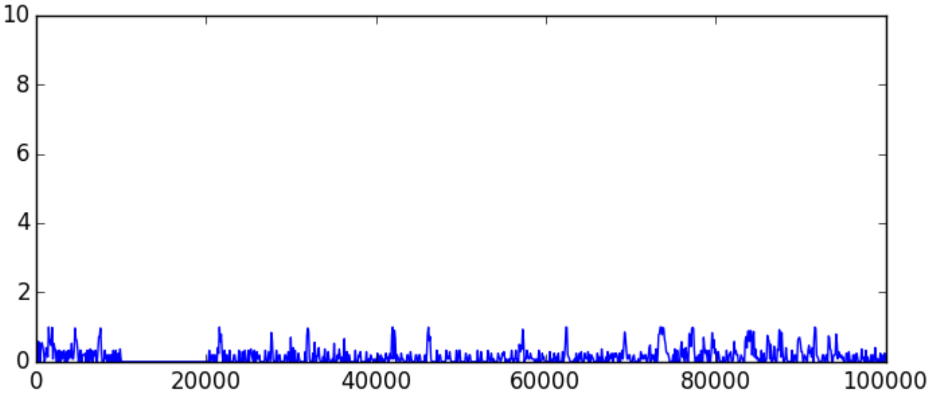

Figure 2.6: Protocol field entropy during TCP-SYN attack

that is normally attack packets have similar lengths [7]. When the length of packets remains same, then entropy decreases as shown in Fig.2.7.

2.3.2 Smurf Attack

Smurf attack uses ICMP icho reply packets to attack a server. Characteristics of attributes during smurf attack are explained.

1. Source IP and Destination IP address Entropy

   In case of smurf attack, Source IP addresses or IP addresses of hosts whom the attacker sends the ICMP echo request messages. The number of hosts may vary depend upon the size of the attacker's network. In our case, we are getting deviation in entropy of Source IP addresses it means Source IP addresses are repeated during

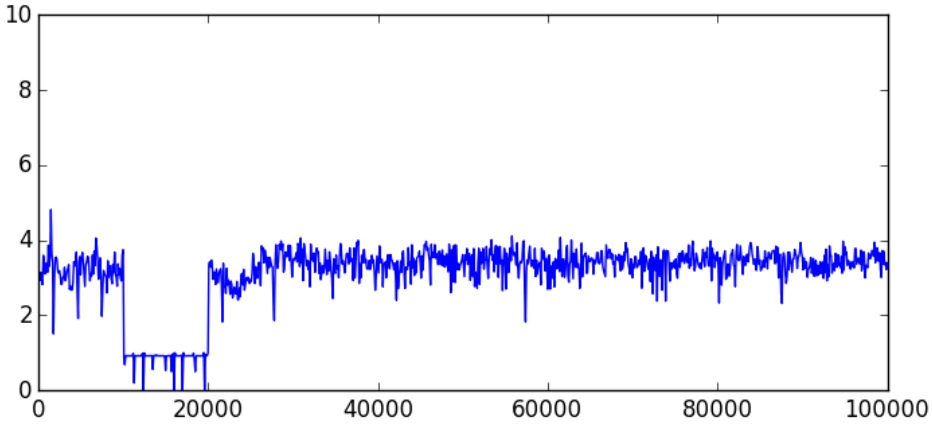

Figure 2.7: Length field entropy during TCP-SYN attack

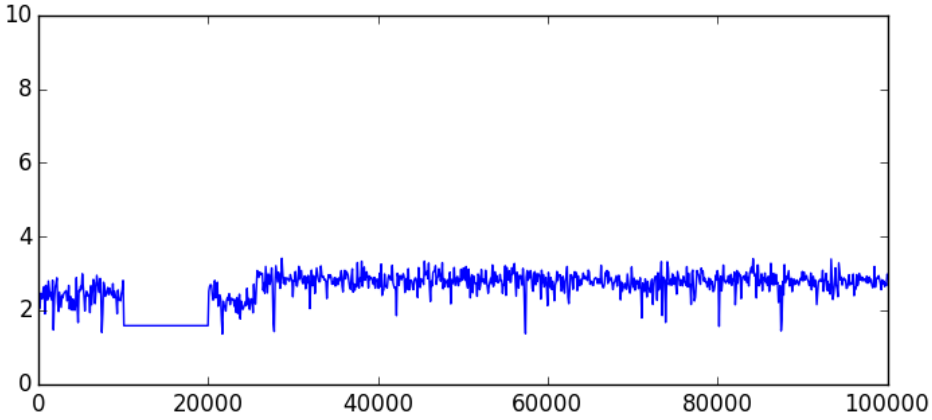

Figure 2.8: Source IP entropy during Smurf Attack

the attack as shown in Fig.2.8. We are getting this useful deviation because in case of smurf attack the attacker has not generated Source IPs randomly. Every host in the attacker's network will send the ICMP echo reply to Victim (server), so destination IP will remain same in all the attack packets that will decrease the entropy of the Destination IP field shown in Fig.2.9.

2. Length and Protocol Field Entropy

   As we said earlier in this section, attack packets usually have the similar size so the entropy of length field will show deviation during the attack shown in Fig.2.10. Since ICMP flood packets use the ICMP protocol, so when we plotted entropy of Protocol field in Fig.2.11 during the attack it was zero because all attack packets are having the same protocol(ICMP)

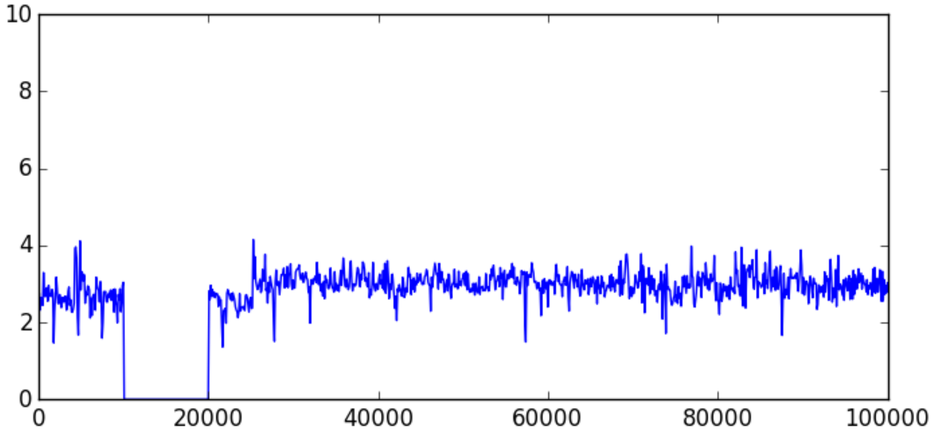

Figure 2.9: Destination IP entropy during Smurf Attack

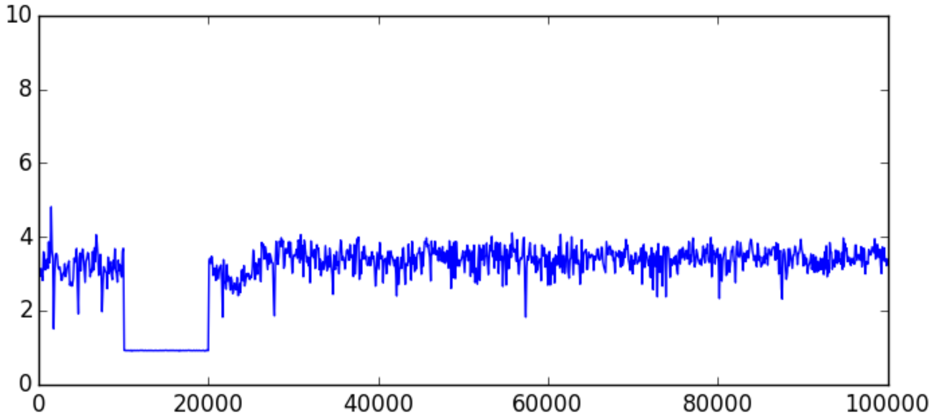

Figure 2.10: Length field entropy during Smurf Attack

smurf attacks use ICMP packets that do not have port numbers, Flag fields.

2.3.3 UDP Flood Attack

UDP flood attack is easiest DDoS attack to perform. It targets ports of server randomly so that server wastes its resources in sending ICMP destination unreachable messages.

1. Source IP and Destination IP address Entropy-

Source IP again spoofed, so an attacker chooses some random Source IPs, but sometimes it chooses from a limited pool that will decrease the entropy of Source IPs during the attack as in Fig.2.12. Every bot targets a single or less number of servers that reduce the entropy of destination address during the attack as shown in Fig.2.13.

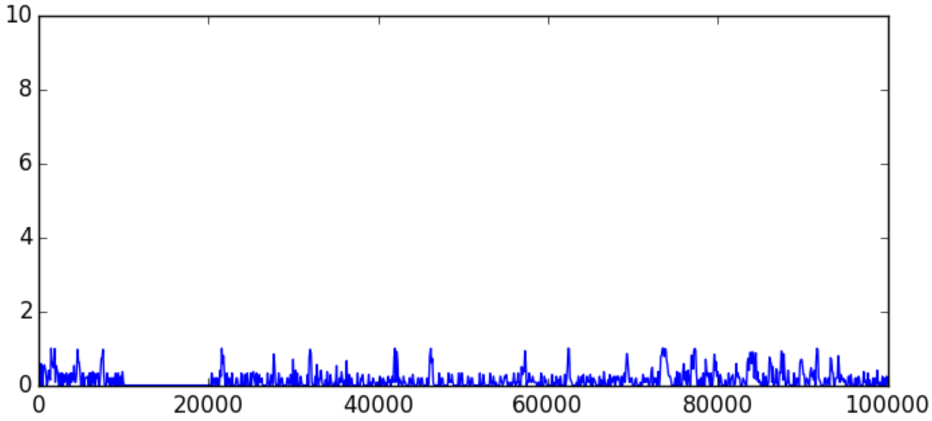

Figure 2.11: Protocol field entropy during Smurf Attack

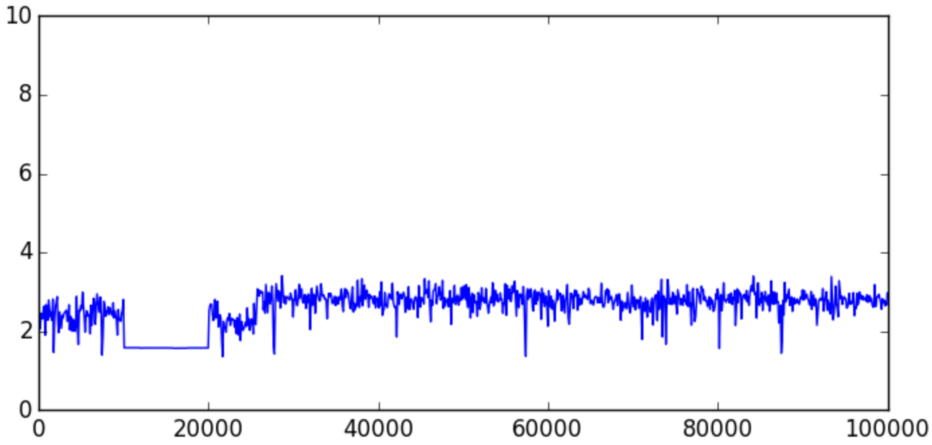

Figure 2.12: Source IP entropy during UDP Flood Attack

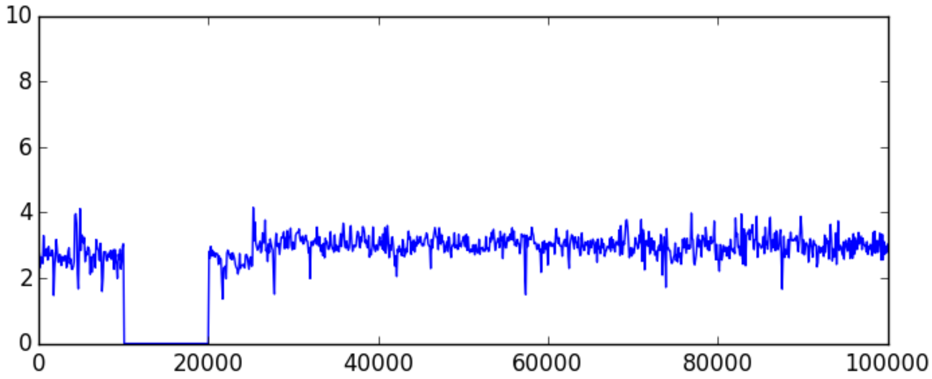

Figure 2.13: Destination IP entropy during UDP Flood Attack

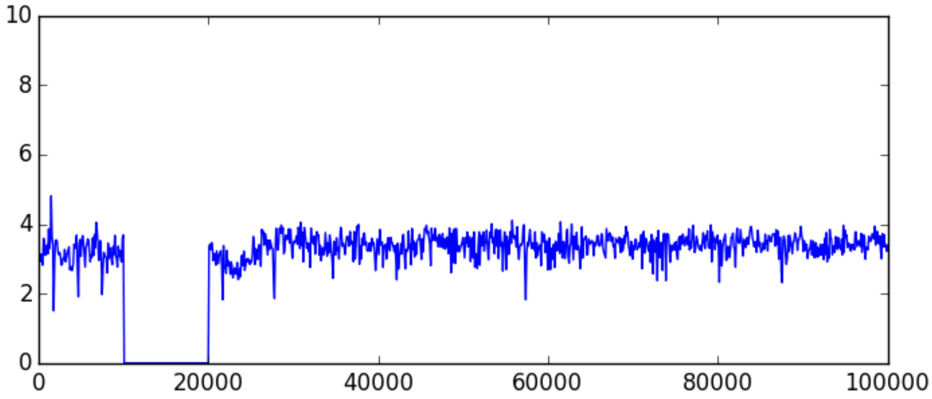

Figure 2.14: Length field entropy during UDP Flood Attack

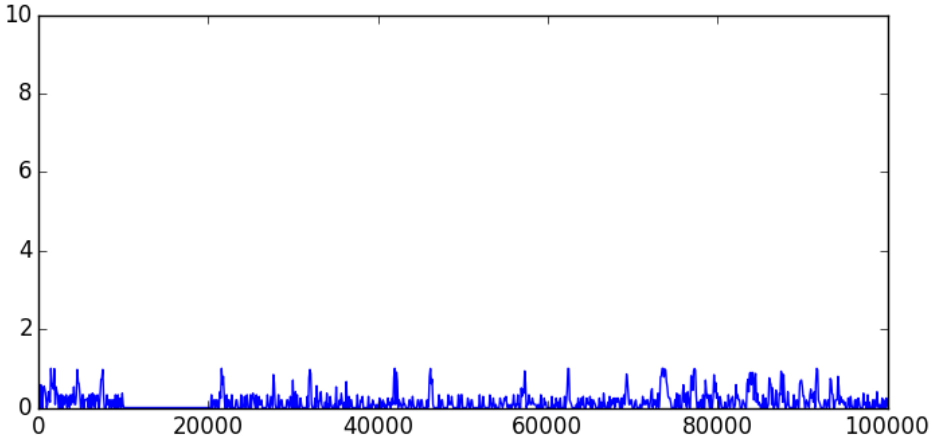

Figure 2.15: Protocol field entropy during UDP Flood Attack

2. Flag, Length and Protocol Entropy

Again the scenario is same as TCP-SYN flood, here attacker uses UDP protocol so flags field will not be used. As we explained the earlier size of UDP attack packets also remains similar during the attack, which considerably decrease entropy values of length shown in Fig.2.14. As the name suggests, UDP flood packets use UDP protocol, so this repetition of protocol decreases the entropy of the protocol field clearly shown in Fig.2.15.

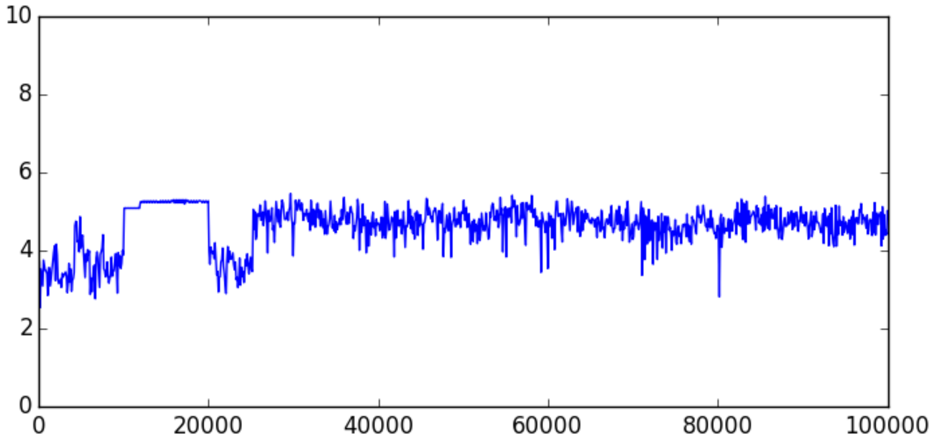

Figure 2.16: Source port entropy during UDP Flood Attack

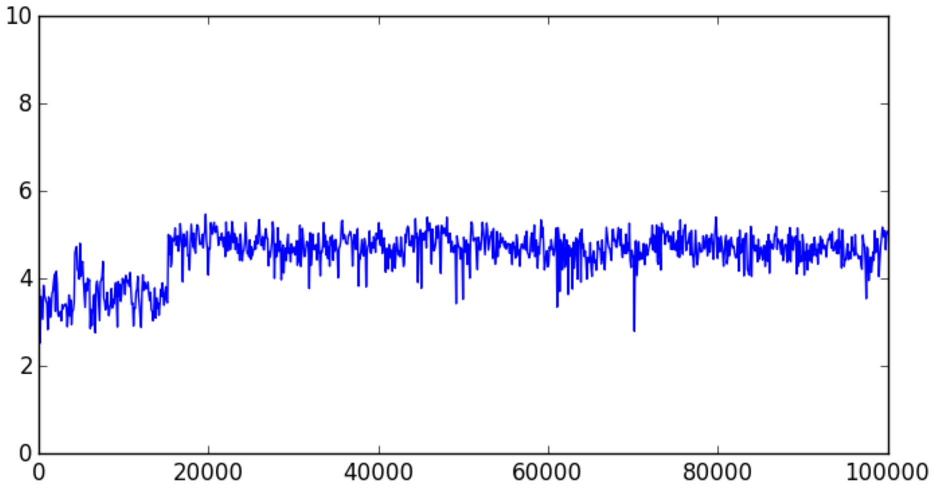

Figure 2.17: Destination port entropy during UDP Flood Attack

3. Source Port and Destination Port Entropy

As discussed earlier in this section, Source port of attack packets are usually random these days hence, not a big deviation during the attack as shown in Fig.2.16. In case of destination port attacker first check which ports are open on the server side. Then the attacker targets only those ports that are open to disturb the service given by the server. Since attacker attacks all the open ports randomly, therefore, this hardly affects entropy of destination ports as shown in Fig.2.17.

2.4 Summary

In this chapter, we dealt with DDoS detection problem. An entropy-based approach is explained which we have incorporated with the shuffling-based prevention mechanism explained in next chapter. We have examined different packet attributes against all famous types of DDoS attack for entropy based detection. Attributes like destination IP, protocol, length provide better threshold for all type of DDoS attacks.

Chapter 3

Prevention of DDoS Attack

CHAPTER 3

Prevention of DDoS Attack

This chapter elaborates about prevention of DDoS on cloud environment. After explaining different types of techniques used by researchers so far, we have chosen shuffling-based approach for DDoS prevention. Entropy-based detection has been incorporated with the prevention model. A greedy algorithm for client assignment is explained in the chapter along with necessary assumptions.

3.1 Literature Review

A number of mechanisms proposed in the last few years to mitigate DDoS can be grouped under various approaches.

1. Filtering-based Defense-

For stopping DDoS attack traffic, filtering-based defenses deploy packet filters on Internet core and edge routers. Aim of these filters is to block clients with spoofed IP addresses, hosts emitting excessive traffic, or flows deemed as malicious by their receivers. Ingress filtering [8] was proposed to validate the source address of each packet at its network entry point. These filters only allows traffic from a limited range of IP addresses. Although ingress filtering helps individual ISPs to regulate Internet users and prevent network misuse, this mechanism has not received wide enough adoption to eradicate IP spoofing. Attackers now a days use to create large botnets and attack without spoofing IP addresses. For mitigating flooding attacks with or without IP spoofing Mahajan et. al [20] proposed Pushback to rate-limit inordinate senders via aggregate-based congestion control. An aggregate is a collection of packets sharing some common properties, such as address prefix, application, or packet type. According to Pushback attack traffic can be classified into different aggregates from legitimate traffic. Internet routers are expected to identify and rate-limit all attacking aggregates in a collaborative manner. However, differentiating well-masked attack traffic from legitimate traffic is a difficult job that requires deep packet inspection, costing precious CPU cycles from the routers. Liu et. al [19] deployed Stopit servers on the destination side that collect complaints about malicious source nodes and tells Stopit servers on the source side to block malicious flows locally. Each Stopit server manages one autonomous system (AS) that is considered a fate-sharing unit.

2. Capability-based Defense

Filtering-based defense encounters attacker identification problems. In capacity-based mechanism destination node decides whether a particular source node is allowed or not. Destination node also sets the limit on the volume of traffic that can be sent by a flow within a specified time window. Source nodes that violate the assigned capability will be considered malicious. In early work related to capability-based mechanism, an Internet host that wants to send packets to another host must first request the receiver's approval. The source of a request to send is attested by the forwarding routers. The approval from the destination node is issued in the form of a flow-bound capability token that also defines the maximum throughput allowed. Although only a small portion of the overall bandwidth is expected to carry capability requests under normal conditions, some early solutions are vulnerable to denial of capability (DoC) attacks. DoC attacks simply flood the capability request channel to deny access to legitimate communications. Parno et. al [21] proposed a proof-of-work (PoW) solution that guarantees per-computation fairness. Before capability requests from a particular source can be delivered to the destination, the source node must solve crypto-puzzles disseminated by well-provisioned services (e.g. DNS) and attach the answers to the requests. Participating forwarding routers are responsible for verifying the answers and dropping all invalid requests. Enforcing PoW will reduce attackers traffic rate and increase the chances that legitimated capability requests are going to reach the desired destinations.

3. Overlay-based Defense

Secure overlay networks provide an network of proxy between clients and server. Secure Overlay networks were proposed to provide authentication, filtering, indirection, attack tracking and tolerance on top of the physical Internet infrastructure. SOS [14] introduced a three-tier network design with doubly indirect routing to assure DDoS resiliency. A small portion of all overlay nodes are employed to form each tier. Incoming flows are forwarded to the beacon nodes after successful authentication, and then routed to the secret servlets that eventually lead to the protected server. Overlay nodes can join and exit the network at run time without affecting the overall stability. It was robust against pure flooding attacks in that attackers have to bombard a large number of overlay nodes to cause a service disruption. A spread-spectrum method [22] was proposed to encourage clients sending duplicated packets to multiple overlay nodes to ensure a high packet delivery rate during an attack. Same as SOS, Phalanx [6] recommended using a swarm of proxy nodes to serve as mailboxes that bounce end-to-end traffic in the middle of the Internet. The destination hosts are expected to actively contact the mailboxes to pick up legitimate packets, leaving attack traffic to be filtered out. Overlay networks are relatively static. Thus, the overlay itself may become a target of a flooding DDoS attack.

4. Shuffling based Mechanism-

This approach was proposed by Jia et al. [12]. Cloud infrastructures are static and homogeneous for ease of administration, so it leaves ample opportunity for attackers to compromise them. To counter this in shuffling based approach, they shuffled clients intelligently among servers to save maximum number of benign clients from on-going attack by isolating them from malicious clients. By doing this targets (servers) become dynamic, so it's hard to follow them by attackers. This solution may increase the cost for potential attackers by complicating the attack process to make the network more secure against naive and persistent attacks. We have also adopted this technique for DDoS prevention with few modifications.

We have used the resource elasticity capability of the cloud for our defense mechanism. When the attack occurred, we instantiate new replica servers and clients on the attacked servers are reassigned to new replica servers using an algorithm that is discussed in the following sections. So, we are creating an environment that will evade both types of malicious clients that is naïve and intelligent. Naïve malicious clients will not able to follow the movement. Intelligent malicious clients may follow the move, but we are using a greedy algorithm for assignment that will maximize the number of innocent clients saved in each round of shuffle. It will try to isolate benign clients from the malicious ones by shuffling. Our approach does not use any authentication mechanism as used in [12], it provides internet services to anonymous users.

3.2 Shuffling Based Techniques

This strategy is a reactive approach. We have created a mathematical model of the approach. Initially, we started with N number of clients. We have S replica servers on which we will assign these customers. The number of replica servers will remain constant for each shuffle. Initially, we assign N clients on S replica servers randomly. When an attack occurs on some replica servers, we mark them and all the clients associated with these servers are also marked. Every marked client is equally suspect to be malicious. We do not disturb clients on non-attacked servers, so we instantiate new replica servers in place of these non-attacked servers to make the number of replica servers constant. Clients on attacked servers considered for the next shuffle. All the attacked replica servers are taken offline and reused after recycling for the next shuffle. Now, we have a new value of N that is the number of clients on attached servers. As we mentioned the number of replica servers would remain constant, so we will use S replica servers for each shuffle. Now, we will estimate the number of malicious clients that are M according to random estimation function proposed in subsequent sections. By using a greedy shuffling algorithm, we will assign these N clients on S replica servers. We will repeat the shuffling process until we save the desired number of innocent clients. The aim of this approach is to save the maximum number of benign clients from ongoing DDoS attack because we do not want harmless clients to be affected by the DDoS attack. We are also using a limit of clients on each replica server which will maintain quality of services for each client. Since shuffling is a stochastic process, we can only calculate the probability that a given replica server is attacked or not. T is number of benign clients saved, by using that probability we can derive E (T), the expected number of benign clients to be saved in one round. Therefore, solving the following optimization problem will maximize the number of benign clients to be saved.

$$E\left(T\right) = \sum_{i=1}^{S} p_i x_i = \frac{\sum_{i=1}^{S} \begin{pmatrix} N - x_i \\ M \end{pmatrix} x_i}{\begin{pmatrix} N \\ M \end{pmatrix}} \tag{3.1}$$

subject to $\sum_{i=1}^{S} x_i = N$

where $x_1, x_2...x_i$ denotes no. of clients assign to each replica server S, p_i denotes the probability that i_{th} replica is not under attack.

As explained in each shuffle, we instantiate new replica servers to make number of replica servers constant.Hence we need a big pool of replica servers, initially we start with enough number of replica servers so that they could serve all the clients effectively. Shuffling process is stateless in order to reduce overhead.

3.2.1 Assumptions

The following assumptions were made during our work-

1. If a malicious client is associated to a replica server, it will always cause Distributed denial of service attack on that server.

2. Every replica server has enough number of resources so that it can easily serve allocated number of clients.

3. In each shuffle we are assuming a different set of malicious clients. Because it may be the case that all malicious clients not attacking concurrently.

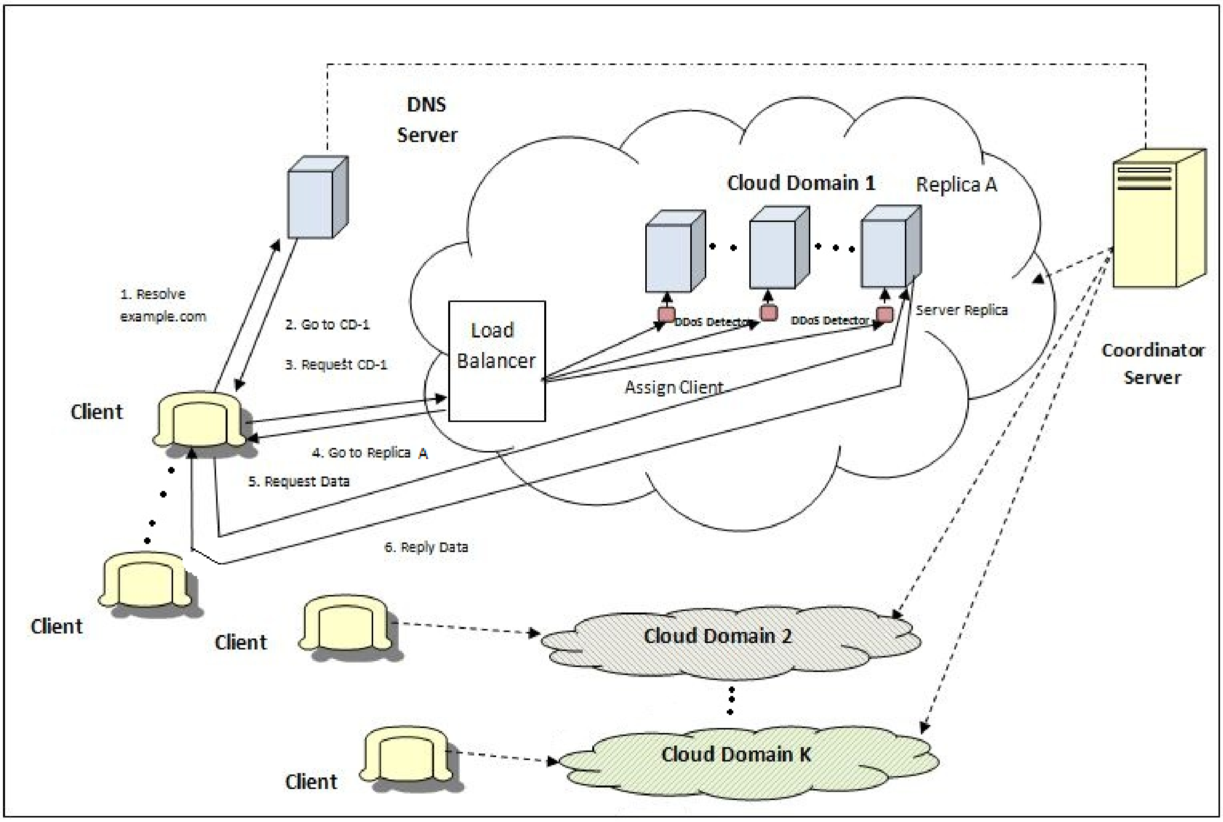

Figure 3.1: Architecture of Cloud Environment

4. For web services redirection can be implemented by HTTP redirection code(3xx),or run a snippet of redirection code inside client browsers.

5. Replica servers that are being attacked can be recycled after some time.

3.2.2 Architecture

The overall architecture of system is shown in Fig.3.1 is described below-

1. DNS Servers-

The DNS server will redirect the client to a cloud domain where the serving replica is present. The DNS server contact to Coordination server which has a list of active replicas, each replica has an IP address that will uniquely identify it. The client will initially assign to any one of these replica servers decided by the load balancer.

2. Load Balancer-

Load balancer assigns new clients to an active replica server. Each client IP is only assigned to one replica server. In case of attack when shuffling starts, the load balancer will inform the replica server and client about the new assignment. This is done by HTTP redirection code(3xx) the client redirection task is prioritized over any other task, Client redirection traffic is also prioritized in the cloud environment. At least one load balancer is installed in each cloud domain to keep track of assignment of clients to servers. There can be more than one load balancer to ensure fault tolerance.

3. Replica Servers-

Replica Servers are the main target for attackers. Each replica server has a unique IP address. When there is no attack, a small number of replica servers are maintained to serve the clients. When an attack happens, we instantiate new replica servers in place of attack servers and reassignment process take place. Replica servers perform whitelist-based filtering, only allow clients whose IPs are confirmed by the load balancer.

4. Coordination Server-

Coordination Server is the central pillar of the whole system, it tracks the number of clients associated with each server and marks the replicas that are under attack on the basis of the information gathered from detectors. Based on number of attacked replicas it estimates malicious clients and assign the affected clients on newly instantiated replicas using an algorithm described in following sections.

5. DDoS Detectors-

On each replica server, a DDoS detector is also deployed. This detector monitors all incoming traffic on that replica server and uses an entropy-based approach explained in the previous chapter. As soon as detector analyzed deviation from threshold it reports directly to Coordination server about the attack.

3.2.3 Greedy Algorithm for Assignment

A greedy algorithm is used for client assignment. This algorithm uses two more algorithms MaxAssign, which is assigning clients to servers based on the probability function described earlier. The second algorithm is ShuffleNeeded, which is checking that whether we have saved required number of clients if not then perform more shuffles. Before executing actual algorithm, we are taking desired percentage of benign clients to be saved as an input.

Algorithm 1: Greedy Algorithm

```
Data: Number of clients N, Number of malicious clients M, Number of servers S,
      Limitation of clients on a server L
Global variables $tac = 0, tas = 0$;
if $N \leq S$ then
    Assign a replica server for each client
    Mark tac, tas in case of attack
    if All clients are assigned to servers then
        ShuffleNeeded()
else if $S$==1 then
    Assign all clients to that server
    if an attack happens then
        We can't save any clients
    else
        All clients are saved from attack.
    Exit
else if $M == 0$ then
    Evenly distribute clients to replica servers
    Mark tac, tas in case of attack(because M=0 is just an estimation)
    if All clients are assigned to servers then
        ShuffleNeeded()
else
    $\mu = MaxAssign(N, 0, N-1, M, L)$
    Mark tac, tas in case of attack
    if All clients are assigned to servers then
        ShuffleNeeded()
    RemN=N-$\mu$
    RemS=S-1
    RemM=round $(M * RemN)/N$
    Greedy(RemN, RemM, RemS,L)
```

Algorithm 2: MaxAssign

```
Data: N,LB,UB,M,L
max=0,maxassign=0;
fi=Min(UB,L)
for i= LB to fi do
    Save = $\begin{pmatrix} N - x_i \\ M \end{pmatrix}^{x_i} / \begin{pmatrix} N \\ M \end{pmatrix}$
    if save>max then
        max= Save
        maxassign=i
return maxassign
```

Algorithm 3: ShuffleNeeded

```
Saved clients from attack till now = N - tac
if saved % of clients > desired percentage then
    Process complete with sh shuffles
    Exit
else
    sh=sh+1
    new_N=tac
    y = number of malicious clients in previous shuffle
    new_mal = random [tas, min(y,tac)]
    tas=0
    tac=0
    Greedy(new_N,new_mal,S,L)
```

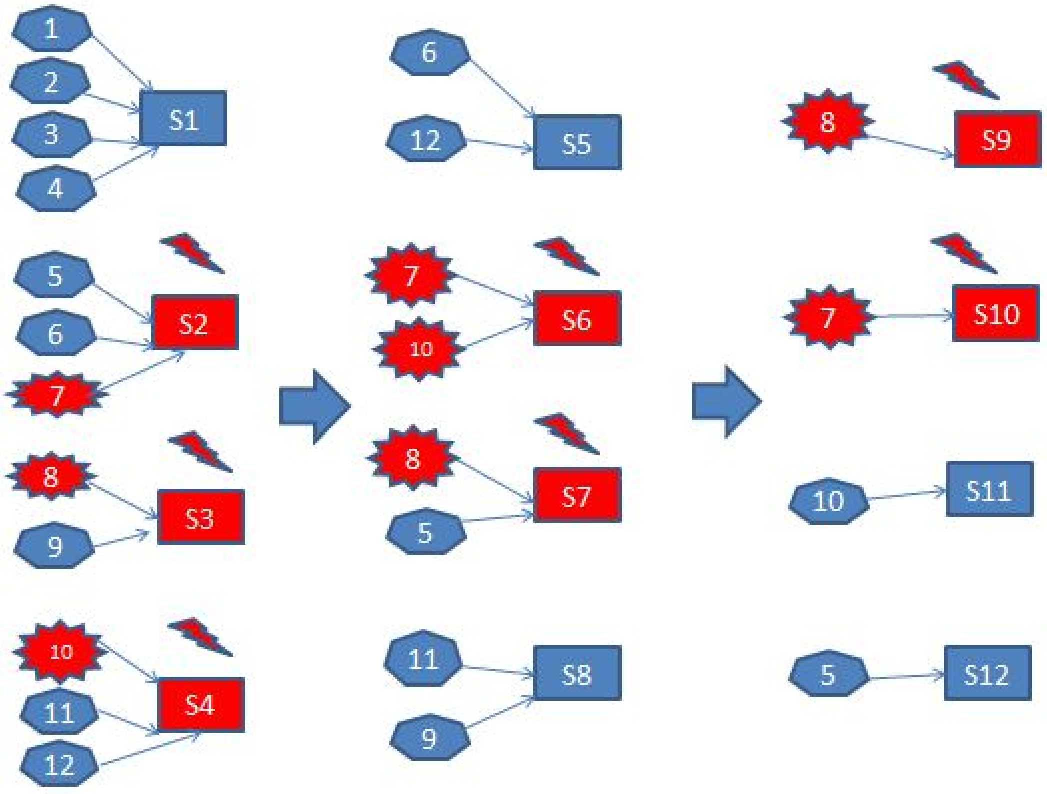

Figure 3.2: Saving of all benign clients after two rounds of shuffle

The shuffling process is explained through a Fig.3.2. in which, we have taken 12 clients and 4 servers. Initially clients were assigned randomly. When attack happens on server S2,S3 and S4 we mark all clients on these servers and reassign them on S5,S6,S7 and S8. Now, suppose S6,S7 were attacked. We will reassign all clients on these attacked servers on server S9,S10,S11 and S12. If attack happens on S9 and S10 we do not initiate any action because client 8 and 7 were only clients on these servers. So, they have caused the attack and our motive is to isolate attackers from benign clients. Therefore, we have saved all the clients after two shuffles.

3.2.4 Estimation of Malicious Clients

For estimating malicious clients among all clients, a random estimation approach is proposed. Since,only number of servers being attacked and clients associated to these servers is known to us. So, only a rough estimation of number of malicious clients can be made.

For that following function is used:

Estimated number of malicious clients = Random [x, min (y, z)]

Where

x=Total affected servers

y= Estimated malicious clients in previous shuffle

z=Total clients on attacked servers

In the first round during the estimation of the total number of malicious clients above function can not be used as value of the term ‘Estimated malicious clients in previous shuffle’, is unknown to us. Therefore, following function is used for first time estimation:

Estimated number of malicious clients for first shuffle = Random [x,z]

But for ease of operation we are taking M as input to our greedy assignment algorithm for the first time.

Note- If a single client is assigned to a replica server and that server is under attack it means that the client is malicious so that client will be stopped from participating in further shuffling because our aim is isolate benign clients from malicious ones.

3.3 Summary

Shuffling-based mechanism for DDoS prevention is discussed in this chapter. An architecture of cloud environment that includes both detection and prevention strategies also explained. A greedy algorithm is also described for client assignment along with a random estimation function. Implementation of this algorithm is described in next chapter.

Chapter 4

Simulation & Analysis

Simulation Environment

Simulation Analysis

Summary

CHAPTER 4

Simulation & Results

4.1 Simulation Environment

In this part, we evaluated the performance of the shuffling mechanism using a greedy algorithm and proposed estimation function. MATLAB is used for simulation of the greedy algorithm.

Entropy-based detection strategy was also implemented. For implementation of entropy based technique 'CAIDA 2007 DDoS attack dataset' is used as a attack dataset [2] and 'CAIDA 2008 anonymized dataset' is used as a normal traffic dataset [1]. By using Wireshark '.pcap'(packet capture) files were converted into '.csv'(comma seperated values) files and then by using a Python script entropy of destination IP in the packet windows of 100 and 1000 was calculated.

4.2 Simulation Analysis

Detection and prevention schemes for DDoS were simulated in different environments. Analysis of the implementation work is explained below-

4.2.1 Detection Analysis

Entropy values of normal traffic in the windows of 100 and 1000 packets were plotted as shown in Fig.4.1 and Fig.4.2 respectively. Attack packets were mixed with normal

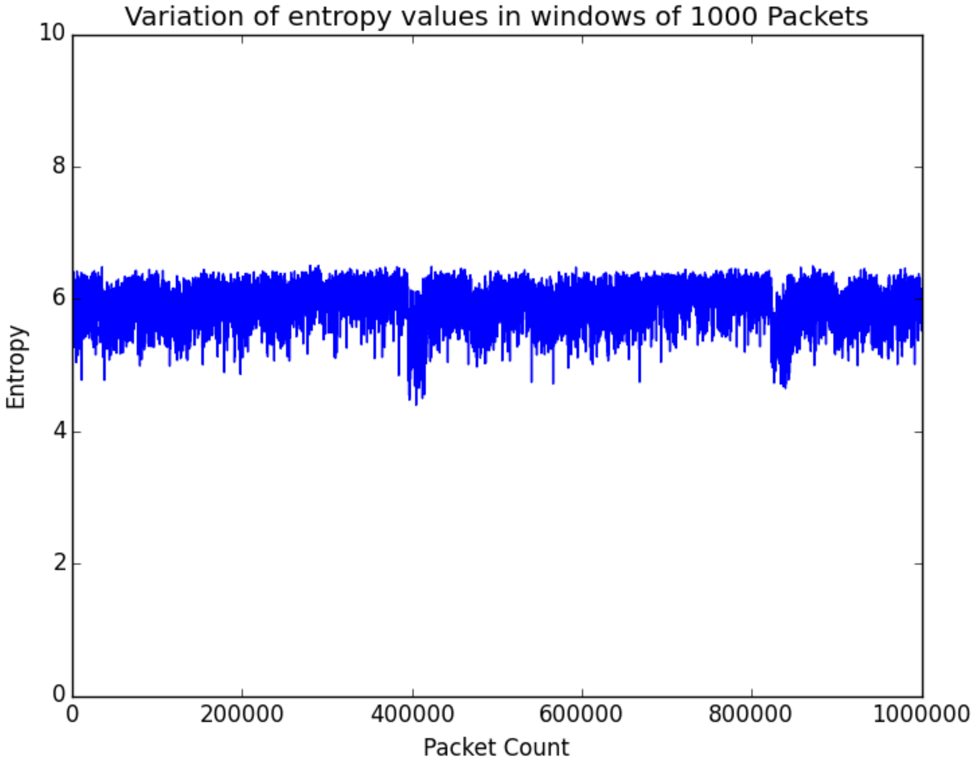

Figure 4.1: Entropy calculation in windows of 100

packets (from packet count 100000 to 200000) remaining packets were kept same. We have seen a fluctuation of entropy values in the range of 100000-200000 packet count where we mixed the attack packets as shown in Fig.4.3 and Fig.4.4. Entropy decreased in this region, It represents that in this region distribution of Destination IPs are less random than normal scenario.

4.2.2 Shuffling Process Analysis

This algorithm was checked for two scenarios-

1. Keeping number of clients fixed as 1000 and number of servers as 50, when we tried to save 80% of benign clients a graph is plotted, each data point simulation was run 10 times to generate average data points.

By increasing number of insiders it is clear from the Fig.4.5 that number of shuffles will also increase.

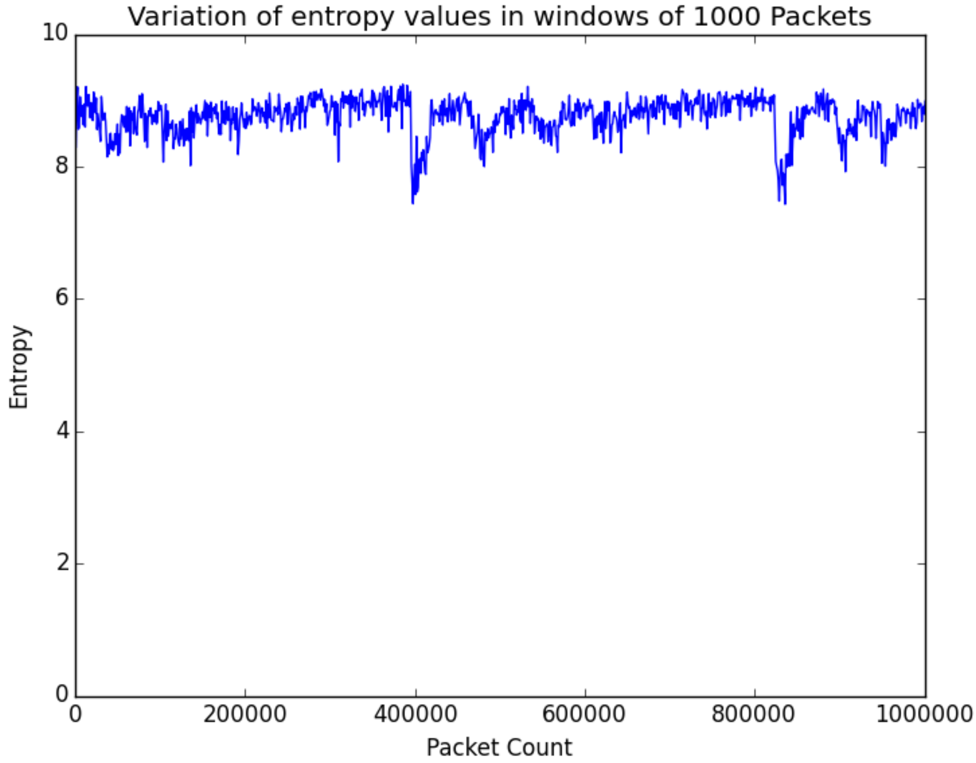

Figure 4.2: Entropy calculation in windows of 1000

2. By keeping clients fixed as 1000 and number of insiders as 50 a second plot is drawn to show relationship between number of servers and number of shuffles, when we try to save 80% of benign clients.

It is clear from the Fig.4.6 that by increase in number of servers, number of shuffles will decrease.

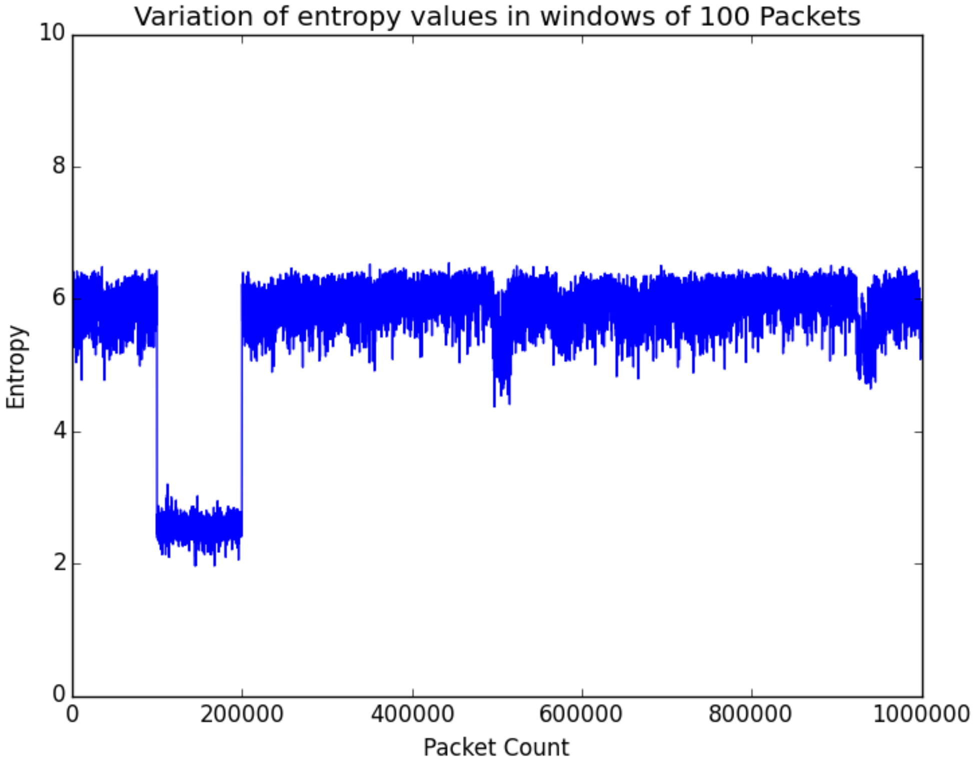

Figure 4.3: Entropy calculation in windows of 100 when attack packet mixed

4.3 Summary

Simulation of entropy-based detection and shuffling-based prevention has explained in this chapter. Entropy-based detection technique has been implemented using Wireshark and python. The shuffling-based mechanism has been simulated using MATLAB. We have shown that entropy of destination IP shows a remarkable deviation during the DDoS attack that is helpful in detecting DDoS. For indicating the usefulness of the shuffling approach, we implemented the relationship of malicious clients with number of shuffles, replica servers with the number of shuffles that displays the performance of the algorithm with changing criteria.

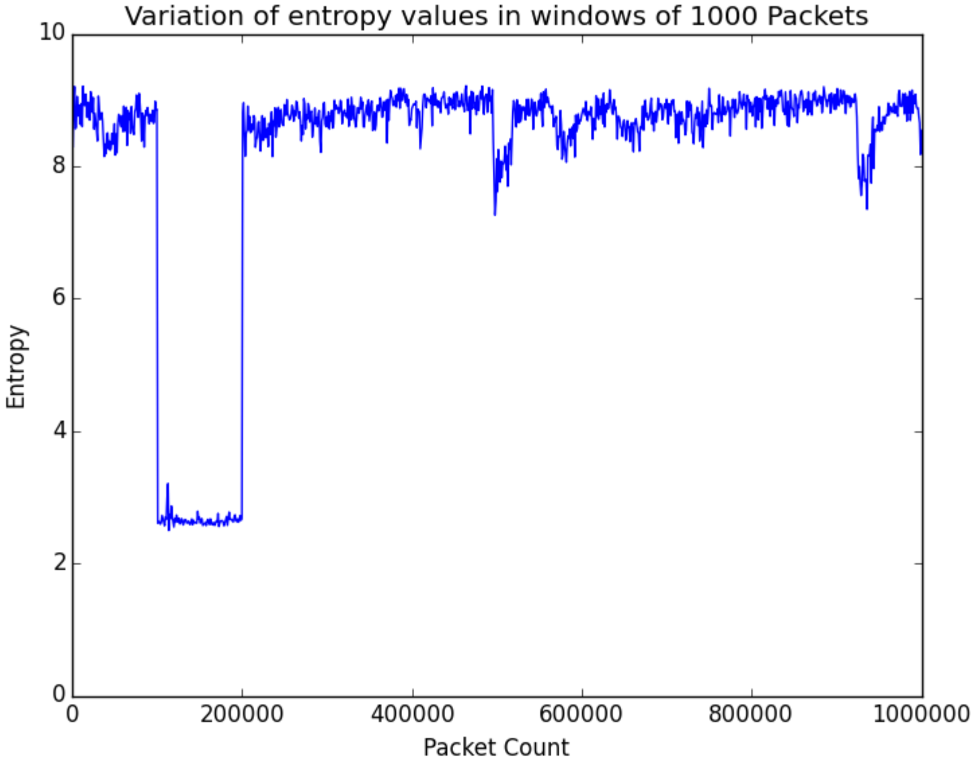

Figure 4.4: Entropy calculation in windows of 1000 when attack packet mixed

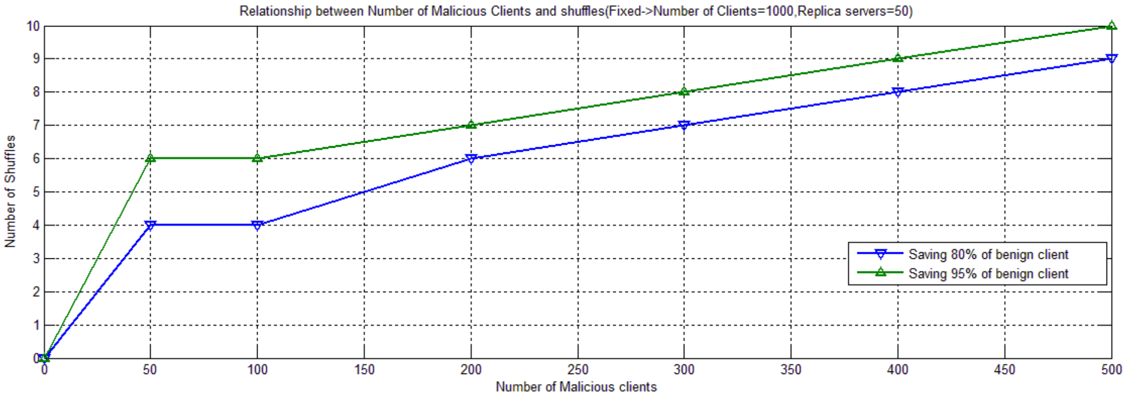

Figure 4.5: Relationship between number malicious clients and shuffles

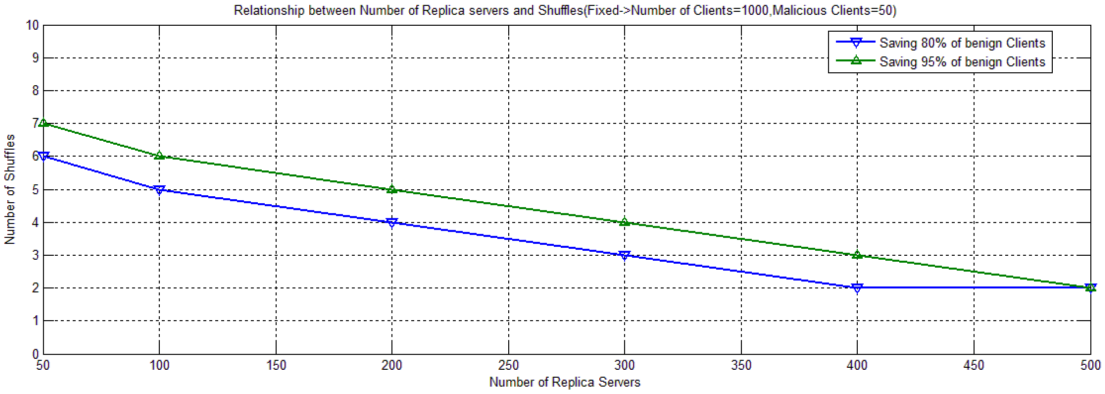

Figure 4.6: Relationship between number of replica servers and shuffles

Chapter 5

Conclusion and Future Work

CHAPTER 5

Conclusion and Future Work

In this thesis, we have explained about DDoS detection and prevention model in the cloud environment. A shuffling-based mechanism for DDoS prevention is used in which our aim was to save maximum number of benign clients from ongoing attack through intelligent shuffles of clients. For detection of attacks, an entropy-based technique is utilized. We have analyzed that some attributes like packet size, destination IP and protocols can also be utilized as an essential attribute in case of entropy-based detection that has not been used by researchers earlier. Their usefulness for all the famous types of DDoS attacks is explained through simulation. A model that incorporates both detection and prevention mechanisms of DDoS is presented. A greedy algorithm is proposed for client assignment that assigns clients intelligently on the replica servers that uses a proposed random function for malicious client estimation. Through experiments, it is proved that desired percentage of benign clients can be saved from the attack using this algorithm in few shuffles. Since, shuffling based prevention method is new a lot of improvements can be made in the future. During shuffles, migration of client sessions can also be addressed.

Bibliography

[1] "The caida ucsd anonymized internet traces 2008." `http://www.caida.org/data/passive/passive_2008_dataset.xml`. Accessed: 2015-05-24.

[2] "The caida ucsd "ddos attack 2007" dataset." `http://www.caida.org/data/passive/ddos-20070804_dataset.xml`. Accessed: 2015-05-24.

[3] "Nust dataset." `http://wisnet.seecs.nust.edu.pk/projects/nes/datasets.html`. Accessed: 2015-05-24.

[4] BELLAICHE, M. and GREGOIRE, J.-C., "Syn flooding attack detection based on entropy computing," in *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*, pp. 1–6, IEEE, 2009.

[5] CABRERA, J. B., LEWIS, L., QIN, X., LEE, W., PRASANTH, R. K., RAVICHANDRAN, B., and MEHRA, R. K., "Proactive detection of distributed denial of service attacks using mib traffic variables-a feasibility study," in *Integrated Network Management Proceedings, 2001 IEEE/IFIP International Symposium on*, pp. 609–622, IEEE, 2001.

[6] DIXON, C., ANDERSON, T. E., and KRISHNAMURTHY, A., "Phalanx: Withstanding multimillion-node botnets.," in *NSDI*, vol. 8, pp. 45–58, 2008.

[7] DU, P. and ABE, S., "Detecting dos attacks using packet size distribution," in *Bio-Inspired Models of Network, Information and Computing Systems, 2007. Bionetics 2007. 2nd*, pp. 93–96, IEEE, 2007.

[8] Ferguson, P., "Network ingress filtering: Defeating denial of service attacks which employ ip source address spoofing," 2000.

[9] Gavrilis, D. and Dermatas, E., "Real-time detection of distributed denial-of-service attacks using rbf networks and statistical features," *Computer Networks*, vol. 48, no. 2, pp. 235–245, 2005.

[10] Hogan, M., Liu, F., Sokol, A., and Tong, J., "Nist cloud computing standards roadmap," *NIST Special Publication*, vol. 35, 2011.

[11] Jeong, S., Hyunwoo, K., and Sehun, K., "An effective ddos attack detection and packet-filtering scheme," *IEICE transactions on communications*, vol. 89, no. 7, pp. 2033–2042, 2006.

[12] Jia, Q., Sun, K., and Stavrou, A., "Motag: Moving target defense against internet denial of service attacks," in *Computer Communications and Networks (ICCCN), 2013 22nd International Conference on*, pp. 1–9, IEEE, 2013.

[13] Jun, J.-H., Lee, D., Ahn, C.-W., and Kim, S.-H., "Ddos attack detection using flow entropy and packet sampling on huge networks," in *ICN 2014, The Thirteenth International Conference on Networks*, pp. 185–190, 2014.

[14] Keromytis, A. D., Misra, V., and Rubenstein, D., "Sos: Secure overlay services," in *ACM SIGCOMM Computer Communication Review*, vol. 32, pp. 61–72, ACM, 2002.

[15] Khalil, I. M., Khreishah, A., and Azeem, M., "Cloud computing security: a survey," *Computers*, vol. 3, no. 1, pp. 1–35, 2014.

[16] Lee, F.-Y. and Shieh, S., "Defending against spoofed ddos attacks with path fingerprint," *Computers & Security*, vol. 24, no. 7, pp. 571–586, 2005.

[17] Li, L., Zhou, J., and Xiao, N., "Ddos attack detection algorithms based on entropy computing," in *Information and Communications Security*, pp. 452–466, Springer, 2007.

[18] LIAO, Y. and VEMURI, V. R., “Use of k-nearest neighbor classifier for intrusion detection,” *Computers & Security*, vol. 21, no. 5, pp. 439–448, 2002.

[19] LIU, X., YANG, X., and LU, Y., “To filter or to authorize: Network-layer dos defense against multimillion-node botnets,” *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 195–206, 2008.

[20] MAHAJAN, R., BELLOVIN, S. M., FLOYD, S., IOANNIDIS, J., PAXSON, V., and SHENKER, S., “Controlling high bandwidth aggregates in the network,” *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 3, pp. 62–73, 2002.

[21] PARNO, B., WENDLANDT, D., SHI, E., PERRIG, A., MAGGS, B., and HU, Y.-C., “Portcullis: protecting connection setup from denial-of-capability attacks,” in *ACM SIGCOMM Computer Communication Review*, vol. 37, pp. 289–300, ACM, 2007.

[22] STAVROU, A. and KEROMYTIS, A. D., “Countering dos attacks with stateless multipath overlays,” in *Proceedings of the 12th ACM conference on Computer and communications security*, pp. 249–259, ACM, 2005.

[23] ZSEBY, T., BROWNLEE, N., KING, A., and OTHERS, “Nightlights: Entropy-based metrics for classifying darkspace traffic patterns,” in *Passive and Active Measurement*, pp. 275–277, Springer, 2014.

Dissemination

1. Sidharth Sharma, Sanjay Kumar Jena, “An Entropy Based Model to Detect Distributed Denial of Service Attack on Cloud Environment”, paper presented at *”International Conference on Communication, Information and Computing Technology”* to be published in *”International Journal of Advance Foundation and Research in Computer*(IJAFRC)” (Accepted)

2. Sidharth Sharma, Santosh Kumar Sahu, Sanjay Kumar Jena. “On the Selection of Attributes for Entropy Based Detection of DDoS” , *4th IEEE International conference on advances in computing, communications informatics* (Communicated)