PREVENTION OF DDOS TCP FLOOD ATTACK USING VIRTUAL NETWORK LAYER

A PROJECT REPORT

Submitted by

PRASANNA VENKATESH B [Reg No: RA1411003010210]
BENIEL DENNYSON [Reg No: RA1411003010176

Under the guidance of

Dr. A. JEYSHEKAR

(Assistant Professor (S.G), Department of Computer Sciene & Engineering)

in partial fulfillment for the award of the degree

of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENE & ENGINEERING

of

FACULTY OF ENGINEERING AND TECHNOLOGY

S.R.M. Nagar, Kattankulathur, Kancheepuram District

APRIL 2018

SRM UNIVERSITY

(Under Section 3 of UGC Act, 1956)

BONAFIDE CERTIFICATE

Certified that this project report titled "**PREVENTION OF DDOS TCP FLOOD ATTACK USING VIRTUAL NETWORK LAYER**" is the bonafide work of " **PRASANNA VENKATESH B [Reg No: RA1411003010210], BENIEL DENNYSON [Reg No: RA1411003010176, , ,** ", who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE

Dr. A. JEYSHEKAR
GUIDE
Assistant Professor (S.G)
Dept. of Computer Sciene & Engineering

SIGNATURE

Dr. B. AMUDHA
HEAD OF THE DEPARTMENT
Dept. of Computer Sciene & Engineering

Signature of the Internal Examiner

Signature of the External Examiner

ABSTRACT

Because of exponential growth of the users using Internet applications, providing services to them is prominent requirement of internet. The node providing service to them is to be available anytime and anywhere. But the denial-of- service attacker attempts to exhaust the victimâĂŹs resources such as bandwidth, processing capacity, storage etc by sending huge unwanted traffic to victim node. By doing so, the attacker makes the victim node to prevent the legitimate nodes from accessing the resources or degrade the services provided by victim node. Hence the network/servers become greater risk. A distributed denial-of- service attack is a large scale attack which launches the many DoS attacks directly or indirectly in the distributed manner. The DDoS attack takes large number of compromised nodes in a network to flood the victim nodes simultaneously from multiple places. This kind of attacks is unpredictable and leads to deadly consequences. Therefore in this paper, we explore the scope of the DDoS TCP Flood attack and the possible ways of prevention of the attacks. We propose a prevention mechanism against the DDoS TCP Flood attack that uses a threshold based attack detection and prevention mechanism. The proposed mechanism is set on a virtual machine of a victim node ie server so that the service requests generated by the attackers are not allowed to the server. The experimental results show that the proposed mechanism performs well as compared with other mechanisms..

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my guide, Dr.A. JEYSHEKAR his valuable guidance, consistent encouragement, personal caring, timely help and providing me with an excellent atmosphere for doing research. All through the work, in spite of his busy schedule, he has extended cheerful and cordial support to me for completing this research work.

Prasanna V Balaji , Beniel Dennyson

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

DDoS	Distributed Denial Of Service
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
LSSVM	Least Square Support Vector Machine
IP	Internet Protocol
SDN	Software-Defined Networking
DARPA	Defense Advanced Research Projects Agency
IRC	Internet Relay Chat
ICMP	Internet Control Message Protocol
GAV	General Attack Vector
Ts	Threshold
AQC	Attack Qualifier Condition
FV	Feature Vector
DCAP	DDoS Capture Attack Pattern
WSN	Wireless Sensor Network
IOT	Internet of Things

LIST OF SYMBOLS

α, β	Damping constants
θ	Angle of twist, rad
ω	Angular velocity, rad/s
b	Width of the beam, m
h	Height of the beam, m
$\{f(t)\}$	force vector
$[K^e]$	Element stiffness matrix
$[M^e]$	Element mass matrix
$\{q(t)\}$	Displacement vector
$\{\dot{q}(t)\}$	Velocity vector
$\{\ddot{q}(t)\}$	Acceleration vector

CHAPTER 1

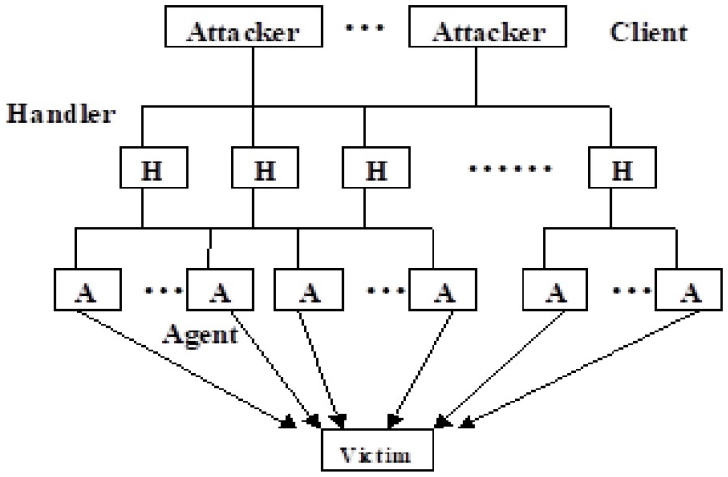
INTRODUCTION

1.1 DDoS DDoS Growth and Effects

drastic growth and success of internet changes the life style of human by the way of getting various services such as banking, transportation, medicine, education etc through internet. Therefore the service providers are able to give service to all, anytime and anywhere. It is accomplished by providing adequate resources such bandwidth, processing capacity, storage etc to the service provider. Therefore the providing high degree of availability of such resource to users is very critical in the internet. Besides it, the attacker also attempts to exhaust the service providerâĂŹs resource which leads to degrade the availability of resources and services. The attack which exhausts the resources of victims by sending flood of requests/messages, resulting denial of service by the service provider to its legitimate users is called denial-of-service attack (DoS). Syn-flood, teardrop, smurf, ping-of-death, finger bomb, black hole, octopus, snork and ARP cache poisoning are some examples of DoS attack. On the other hand, Distributed denial-of-service attack (DDoS) uses multiple malicious nodes distributed globally that generate more HTTP traffic to overwhelm victim resources as shown in Fig. 1. It is very difficult to distinguish the attack HTTP traffic from normal traffic [17]. To defend against DDoS attack, traffic control mechanism such as ingress filtering, route-based packet filtering and rate limiter are used. Ingress filtering and packet filtering mechanism detect the packets with spoofed source IP addresses and drop them. But the effectiveness of these mechanisms depends on global deployment of filtering in the internet which is difficult. Rate limiter controls the traffic in the link of victim when it is overwhelmed by the attacker by sending the unwanted traffic. It is suitable for the attacks having high data rate on a link but not suitable for the attacks with low data rate [5]. The rate limiting techniques are very simple and easy to understand also but it is hard to set up proper threshold values for detecting the attacks [15]. In this paper, we propose a threshold

based DDoS detection and prevention mechanism which uses a virtual machine set on the victim node. The virtual machine allows only the service requests from the legitimate nodes to the server and other requests are not allowed to the server. Therefore the severity due to the DDoS attack is reduced. When the number of service requests generated by a node exceeds the threshold, then the IP address of the node is assumed as suspicious attacker and marked as blacklist node for a limited period and the service requests from blacklist node are not allowed to the victim.

Figure 1.1: Overview of a DDoS Attack



CHAPTER 2

LITERATURE SURVEY

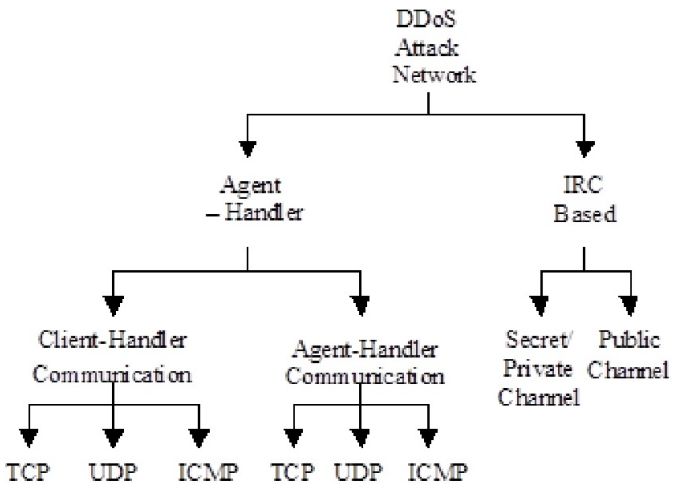
2.1 DDoS and Countermeasures

In a DDoS flood attack, the victim is flooded the with Internet Protocol (IP) IP traffic by the attacker [16]. A huge volume of packets sent by the attacker to the victim system slows it down, crashes the system or saturates the network bandwidth which prevents any legitimate users from accessing the victimâĂŹs network or system. There are many types of UDP flood attack and TCP flood attack in the Internet as shown in Fig 2.1. In UDP flood attack, large numbers of UDP packets are sent to victim system which eventually depletes the bandwidth available for legitimate user [10]. In a DDoS UDP Flood attack, the UDP packets are sent to either random or specified ports on the victim system. UDP flood attacks are designed to attack random victim ports. This causes the victim system to process the incoming data to try to determine which applications have requested data. If the victim system is not running any applications on the targeted port, then the victim system will send out an ICMP packet to the sending system indicating a âĂIJdestination port unreachableâĂİ message. There are several other DDos Attacks which shuts down an entire network, such as Smurf Attacks, Fraggle Attacks, Protocol Exploit Attacks, TCP SYN AttacksPUSH + ACK Attacks, Malformed Packet Attacks

There are a lot of techniques proposed in the literature for mitigating the effects of DDoS attack as shown in Fig 2.2. Honeypots are systems that are set up with limited security to be an enticement for an attacker so that the attacker attacks the honeypot and not the actual system. Honeypots typically have values both in deflecting attacks from hitting the systems they are protecting and serving as a means for gaining information about attackers by storing a record of their activity and learning the types of attacks and software tools used by the attacker. A number of current researches discuss the use of honeypots that mimic all aspects of a legitimate network, in order to attract potential DDoS attackers. This prevents a lot of legitimate systems from getting compromised

and allows the honeypot owner to track the handler or agent behavior and better understand how to defend against future DDoS installation attacks [8].

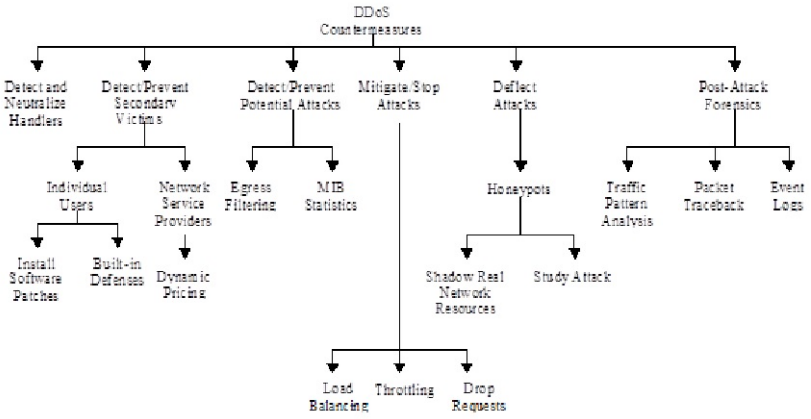
Figure 2.1: Different types of DDoS Attack



Applying a genetic algorithm to find the feature set of each attack, there has been a solution proposed by the SOM for the classification problem. An optimal solution could only be assured with a brute force approach and a significant time commitment (applying soft computing technique to intrusion detection).The distributed nature of DDos attacks make them extremely difficult to combat or trace back, Attackers normally use three methods to spam service requests which are as follows; Spoof (fake IP), Bots to generate service requests, Morph a legitimate users IP address to spam service requests. Various classifications of DDos attacks have been proposed in literature over the past decade. In this paper, we mainly focus on DDos flood attacks as one of the most common forms and provide a solution to the existing drawbacks.

There are a lot of solutions proposed and partial solutions available for mitigating the effects of a DDoS attack. Many of those solutions assist in preventing only limited aspects of a DDoS attack. However, there is no comprehensive solution to protect against all known forms of DDoS attacks. Also, many derivative DDoS attacks are constantly being developed by attackers to bypass every new countermeasure employed until today. More research is needed to develop more effective and encompassing countermeasures and solutions. Therefore in this paper, we propose a prevention mechanism for mitigating the DDoS Transmission Control Protocol (TCP) TCP Flood attack using virtual network layer.

Figure 2.2: DDoS Counter Measures



2.2 LSSVM

The Least Square Support Vector Machine Least Square Support Vector Machine (LSSVM) is a capable classifier in the field of example acknowledgment for the location of variations from the norm from signs, pictures and time arrangement signals[1]. The LS-SVM is an effective technique for arranging two unique arrangements of perceptions into their pertinent classes. It is equipped for taking care of high dimensional and non-direct information. In this work, the LS-SVM is utilized to distinguish illicit exercises in a system. The parameters of the LS-SVM are set amid the instructional course to acquire a high extent of distinguished outcomes

2.3 K-Fold Cross-Validation

K-overlay cross-approval is an approval show for estimating how the results of a numerical examination will disentangle to an autonomous dataset[3]. For the most part, it is used to approve the estimation of execution exactness by and by for a prescient model.K-crease cross-approval was utilized to complete an execution examination of the four prescient displaying calculations utilized as a part of $\mathrm{CS}_D DoS : LS - SVM$.

The dataset was separated into six equivalent estimated lumps, k=6. As an approval for show testing, one of the six lumps was held, and the rest of (pieces) were utilized as preparing information[4]. At that point, the procedure of the six-cross model was rehashed six times, so every one of the six pieces were utilized as approval information for each model. The estimations of all folds are nearly the same, which implies that each overlap has roughly a similar rate for every one of the four grouping calculations. Hence, we can assert that the order comes about are steady and precise, since every calculation gives nearly similar outcomes for each overlay.

2.4 Impact of SDN on DDoS attack defense

The most critical two ideas of SDN Software-Defined Networking (SDN) are control plane deliberation and system network virtualization.They present after properties[5]. Centralized system control: The unified organize working framework (NOS) interfaces with all the switches in the system specifically. In this way, NOS can give a worldwide system topology alongside the continuous system status. Simplified parcel forward: The information plane in SDN just advances parcels in light of the sending strategies produced by control programs. Software based system work usage: System works initially actualized inside a switch or a center box are executed as control programs in SDN.These control programs live over the NOS what's more, speak with switches remotely. Virtualized arrange: Similar to a hypervisor in equipment virtualization, the system virtualization conceals the system topology from control programs with the goal that system work engineers can center around the usefulness execution. Executing SDN influences the DDoS assault safeguard extraordinarily in the two headings. On the brilliant side, SDN makes propelled location rationale and rich ensuing forms less demanding to actualize. On the drawback, the gadgets or center boxes initially disseminated inside the system now should be situated previously NOS. Contrasted and equipment based parcel handling, programming forms parcels is much slower. The arrange deferral and activity overhead caused by the interchanges between the control program, i.e., the DDoS assault safeguard plans, and the switches, may turn into the new assault surface.

2.5 DDoS attack defense in cloud computing and SDN

In light of our examination, distributed computing presents new DDoS challenges, i.e., expanded resistance edge furthermore, dynamic system topology because of its new activity demonstrate. To successfully address these difficulties, the cloud supplier must have the capacity to 1) effectively assign the control of its system to cloud clients; 2) quick reconfigure the control as indicated by the system topology changes caused by unique allotments and relocations. On one side, we could profit by the incorporated organize controller and the system virtualization of SDN. On the opposite side, SDN impacts DDoS assault resistance in negative routes as we talked about ahead of schedule. The negative effect of SDN primarily originates from the proficiency of handling parcels utilizing programming, which may create new assault surface and prompt singlepoint disappointment. When planning a DDoS assault guard arrangement in SDN, one must take the calculation and correspondence overhead into the thought so that no new security powerlessness is presented. To entirety up, we trust SDN innovation will profit the DDoS assault barrier in distributed computing as along as the configuration could deliberately deal with the correspondence and calculation overhead.

2.6 Results of DARPA 1998 Evaluation

The 1998 DARPA Defense Advanced Research Projects Agency (DARPA) interruption location assessment was to a great degree fruitful in giving the principal extensive reasonable assessment of interruption recognition frameworks[6]. The outcomes were valuable in centering exploration and featuring the present abilities and later advances of existing interruption recognition frameworks. The techniques used to make the information also, create assaults functioned admirably and can without much of a stretch be reused to produce more information for future assessments. Scientists who partook in the assessment could see the solid and frail purposes of their own methodologies, and the entire interruption discovery look into group picked up a lot of knowledge about the qualities and shortcomings of current endeavors in the field in general.

Albeit full discourse of the Nitty gritty consequences of the assessment are out-

side of the extent of this proposal, there were a couple of wide perceptions that will be valuable in controlling future endeavors to assess interruption identification frameworks utilizing reenacted PC attacks.First, ebb and flow interruption recognition inquire about frameworks speak to a sensational change over more established watchword spotting frameworks. The best blend of 1998 assessment frameworks gives in excess of two requests of extent in false alert rate with incredibly enhanced discovery precision. Second, current Intrusion Detection Frameworks can effectively recognize more established, known assaults with a low false caution rate, however do not execute too while recognizing novel or new attacks.The rate of recognition is spoken to on the vertical hub furthermore, Results are separated into the two assault classifications that can be made stealthy (Tests and User to Root) along the even hub. Location rates for stealthy assaults were not altogether lower than for assaults performed free.

2.7 Feature extraction in a local analyzer

Despite the fact that an activity screen can create basic includes proficiently, these highlights may not be sufficient to identify assaults. Specifically, the parcel rate furthermore, information rate highlights may just be valuable for recognizing high-volume assaults; and SYN/FIN(RST) proportion has a substantial variety notwithstanding for typical movement and subsequently can't help precisely recognize ordinary organize conditions from arrange peculiarities[7]. To enhance location exactness, one can utilize a neighborhood analyzer to produce more refined highlights, for instance, the SYN/SYN-ACK proportion proposed in and the level of new IP addresses proposed.the SYN/SYN-ACK proportion and the level of new IP addresses either don't prompt great execution of indicators, or require high stockpiling/time multifaceted nature. To address these deficiencies, we propose another sort of highlight called 2D coordinating highlights, which can make particular highlights among typical and assault movement, in this way moving forward exactness of distinguishing assaults.

The inspiration of proposing 2D coordinating highlights is the accompanying. For most Internet applications, bundles are produced from the two has that are occupied with correspondence. Consequently, a few data conveyed by bundles on one course

should coordinate the relating data conveyed by bundles on the other course. For instance, if station A speaks with station B through TCP, at that point we can watch bundles with source A what's more, goal B on one bearing, and we can likewise watch parcel with source B and goal An on the other way. Then again, if a DDoS assailant produces source An on one heading, the reaction parcel may not achieve the connection on the turn around course (where a movement screen is set) in the event that the aggressor parodies its source IP address. Subsequently, we can use this component to distinguish DDoS assaults. To encourage the exchange, we have to indicate 'keys' that contain the data that might show up on the two headings of a connection. For example, the key for TCP SYN parcels on one heading can be characterized by hsrcIP; dstIP;srcPort; dstPort; Seqi what's more, the relating key for SYN-ACK bundles on the other way can be characterized by hdstIP;srcIP; dstPort;srcPort; Ack 1i:

Naturally, by inspecting whether a key matches the comparing key on the other way, we can recognize the SYN surge or SYN-ACK surge assaults. Since the proposed highlight is produced by coordinating the keys on two headings of a connection, we call it 2D coordinating component.

2.8 Distributed Denial of Service Using IRC Botnet

Internet Relay Chat (IRC) Botnets are a rising risk to all associations on the grounds that they can trade off a system and take imperative data also, convey malware[8]. Botnets consolidate individual noxious practices into a solitary stage by streamlining the activities should have been performed by clients to start refined assaults against PCs or systems around the globe. These practices incorporate facilitated examining, DDoS exercises, coordinate assaults, backhanded assaults and other tricky exercises taking put over the Internet. The principle objective of this situation is to perform dispersed assaults utilizing tainted has on the testbed. An Internet Relay Talk (IRC) bot organize enable clients to make open, private what's more, mystery channels. For this, we utilize a LOIC12, an IRC-based DDoS assault age instrument. The IRC frameworks have a few other huge points of interest for propelling DDoS assaults. Among the three essential advantages are (I) they bear the cost of a high level of secrecy, (ii) they are

hard to identify, and (iii) they give a solid, ensured conveyance framework. Besides, the assailant never again needs to keep up a rundown of operators, since he can essentially sign on to the IRC server and see a rundown of every single accessible operator. The IRC channels get correspondences from the operator programming with respect to the status of the specialists (i.e., up or down) and take an interest in telling the aggressors with respect to the status of the operators.

2.9 IMPROVEMENT OVER THE CLP-BASED SCHEME

We propose new techniques to supplant the CLP-based plan to accomplish fast tasks, e.g., 10 Gb/s. In the CLP-based conspire, a scorebook, an accumulation of each property estimation's score, is first produced in view of Bayesian CLP. The score related with each quality esteem is gotten from two histograms; one is the right now estimated and the other is the ostensible profile. The usage multifaceted nature emerges from the figuring of these two histograms for every bundle quality. Despite what might be expected, the LB-based plan does not have to ascertain a deliberate profile histogram, nor does it have to ascertain any sort of histogram continuously. Rather, it appoints a LB for each quality esteem and decides a score for each property esteem in light of the quantity of floods of the related LB. The scorebook can be promptly built by monitoring the flood check of each basin, which, thus, requires as it were standard LB memory access and refresh tasks. Another proposed scoring strategy, called AV, enhances the precision of bundle discarding,4 under all conditions, as analyzed with the CLP-and LB-based plans. This is accomplished by utilizing a quality esteem fluctuation rather than straightforward characteristic values as a LB edge. It is less intricate than the CLP yet more perplexing than the LB. The multifaceted nature originates from the need to ascertain the fluctuation for each trait esteem amid the ostensible profile. It is exceptionally testing to give a viable over-burden control at the point when a framework is under quick changing DDoS assaults. The beforehand proposed Packet-Score plot utilizes a CDF and a stack shedding calculation to produce the disposing of edge. Parcels with scores lower than the edge are disposed of. In any case, if an aggressor changes its assault write and power, the âĂŤ which was legitimate for a specific scope of scoresâĂŤwould likely wind up invalid, along these lines trading off the separation

limit, until a more sufficient is progressively set. This circumstance has a tendency to decline as the scores of an estimation period are utilized as a part of the following time frame, while the assaults keep on change.we have watched that the minute the assaults change, spikes of conceded activity show up (because of the edge refutation clarified above), some of the time going on for a generally extensive time of time. Indeed, even with visit limit refreshes in a little time of time, (the best way to revalidate the edge), the CLP plot still experiences this issue. To address this issue, we apply the traditional extent mix (P/I) conspire in charge hypothesis to decide the disposing of edge powerfully. This control-theoretic approach not just makes the framework more straightforward, yet additionally lessens the computational and memory necessities of the framework. The P/I plot additionally gives a higher level of freedom from the scores produced in the past period, and adjusts quicker to new assaults than the CDFbased stack shedding plan utilized as a part of the first Packet-Score conspire.

2.10 Taxonomy of DDoS Attack Tools and Their Comparison

Kind of Interface utilized: The interface utilized by the DDoS assault devices can be either summon line interface or then again graphical UI. Goldeneye, trinoo, shaft and so forth utilize summon line interface though hoic, User Datagram Protocol (UDP) udp flooder, xoic and so forth utilize graphical UI.

2.11 Assault rate flow

Depending upon the assault rate flow, assault apparatuses can either create the consistent assault movement (no varieties in sending assault ask for) rate and variable assault rate (apparatus can fluctuate the assault rate to keep away from the discovery which can be the expanding rate and fluctuating rate). Working System Supported: various DDoS assault instruments are intended to help the different working frameworks like unix, linux, solaris or windows.

2.12 Assault show

DDoS assault devices can make utilization of either Operator Handler demonstrate or and IRC show. AgentHandler depends on ace slave connection while the IRC framework utilize open channels for propelling assaults.

2.13 Convention

The sort of convention indicates the sort of movement created by the assault devices for producing surge assaults, correspondence between the specialist handler, handler-customer and customer specialist. Surge assaults basically utilize UDP, (ICMP ECHO ask for and ICMP Resound answer), HTTP, (TCP-SYN, TCP-ACK what's more, RST-surge) conventions.

2.14 DDoS assault class

The outcome of a DDoS assault is the inaccessibility of the assets or data transmission of the casualty. Henceforth, the assailants utilize those assault apparatuses that can deplete target framework or system's assets and data transfer capacity. There are number of DDoS assault apparatuses accessible that can exhaust both the assets and the data transfer capacity in the blink of an eye.

2.15 Target regaion

DDoS assaults can either block the connection or end point. In this way, DDoS assault devices are ordinarily intended for the clog at the connection level(congestion at the casualty arrange) or toward the end point level(congestion at the casualty server). All the mainstream assault apparatuses are looked at on the premise of distinguished key highlights as appeared in Table:1. The key highlights incorporates the effect of assault which cause consumption either at data transfer capacity or asset level, extent of the

assault apparatus, the kind of assault propelled, support of working frameworks, usage dialect and so forth. Further, it has likewise been watched that all DDoS assault devices takes after the same summed up assault apparatus design as given by Lee.

2.16 Stachaldraht

Stacheldraht is the C-based DDoS apparatus that can make the ICMP surge, SYN surge, UDP surge and Smurf assault towards the objective. It has the ability to stuff the connection and farce the IP address. It can keep running on the Linux and the Solaris 2.1. It has order line based interface.

2.17 TFN

TFN (clan surge organize) can create a number of various types of assaults. It is additionally called the "Child Of Trinoo". It is the charge line based which executes on the windows, linux and so forth. It is composed in the C dialect and has the assault design like the handler-specialist show. It creates DDoS assault that has the ability to drain both asset and data transfer capacity of the objective.

2.18 Trinity

Trinity is the order line based assault instrument that can dispatch UDP, part, SYN, RST, irregular, banners and invalid surge asks for that prompts the endpoint asset fatigue and connection blockage. This device utilizes the scrambled organization and requires the Linux stage. The engineering model of the Trinity is the IRC-based.

2.19 Bubonic

A C-based assault device which can utilize Linux, Unix and Windows as the hidden stages for its execution. It is a DoS endeavor to abuse or deceive the windows2000

machine by haphazardly sending a colossal volume of the TCP parcels with the irregular settings to expand the heap on the machines which drives the machines to a crash. Arbitrary settings include the setting of arbitrary IP locations and irregular port number. Jar. A summon line based DoS assault instrument sends a expansive number of ICMP parcels with a specific end goal to focus on the casualty machine running on the windows 95 or NT with the goal that the casualty machine neglects to reassemble them for utilize. It execution dialect is C. Be that as it may, this sort of assault don't bring on any radical harm to the casualty framework, and the machine is still in the state, to be recuperated.

2.20 Mstream

A C-based and order line interface DDoS assault apparatus has capacity to manufacture the source addresses. It makes the TCP ACK surge and TCP RST surge solicitations to the objective server. It can create botnets and furthermore parody the ip locations of the aggressors while performing DDoS assaults. Both of these solicitations can debilitate the system assets and expends data transfer capacity of the casualty server .

2.21 Shaft

Shaft is the order line interface DDoS assault instrument that can debilitate the transfer speed and assets of the casualty server. It gives measurements to TCP, UDP and ICMP flooding assaults and helps the aggressors to recognize the casualty machine status (either totally down or alive) or to choose the end of zombies notwithstanding the assault. Its design show is Agent-Handler based .

2.22 Targa

Targa is the C-based assault device which can exhaust the transfer speed and assets. It is the DoS assault device which is the accumulation of the 16 unique projects of DoS.

These assaults can be propelled exclusively and additionally in the gathering moreover. It has the capacity to parody the ip addresses and requires the linux stages.

2.23 Irinoo

Trinoo is the DDoS assault instrument, that uses an ace have and a few communicate has. Ace host trains the different communicate hosts to dispatch the assault. An Application layer assault apparatus that has the ability to exhaust the assets and influences the data transfer capacity of the casualty arrange. It is charge line based and its design demonstrate is the AgentHandler based.

2.24 Blast

Blast20 is the DOS assault apparatus is called as the TCP benefit pressure apparatus can recognize the potential shortcomings in the system servers in a split second.

2.25 DDoS Prevention Using Optimisez Traffic Matrix

Inconsistency location plans can basically be separated into the accompanying specialized classes; rate restricting, information mining, what's more, factual examination strategies. At to begin with, rate restricting systems identify bizarre association conduct in view of the preface that a contaminated host will attempt to interface with a wide range of machines in a brief timeframe. It identifies portscans by putting new associations surpassing a specific edge in a line. An alert is raised when the line length surpasses a limit. The rate restricting methods are straightforward and actualize also.

In any case, they are excessively straightforward, making it impossible to distinguish complex interruptions and it is difficult to set up appropriate limit esteems. Next, information mining procedures are utilized to assemble an identification display (classifier) that can find profile of system highlights. Lee and Stolfo constructed a characterization model to recognize oddities. They made a sensible progress as far as arranging

ordinary and interruption information and lessened misclassification rates by utilizing extra measurable highlights. A meta-identification show [6] was proposed to enhance their past approach.

It utilized consolidated numerous discovery models to expand recognition precision yet numerous models certainly made calculation more intricate. At last, numerous discovery strategies have been proposed in a measurable investigation field. A few measurable examination based discovery models, specifically those depending on checking IP qualities of entry bundles were proposed. Talpade et al. proposed NOMAD which is a versatile and uninvolved system observing framework. It can recognize assaults by dissecting IP parcel header data, for example, an opportunity to live (TTL) field, parcel postpone variety and activity stream. It doesn't bolster making the classifier for high-data transfer capacity movement that is amassed from conveyed sources [3]. Peng et al. [13] proposed a straightforward location conspire called Source IP address Monitoring (SIM) to identify high transfer speed assaults. The model screens entry rates of new source

IP addresses and identifies changes of them utilizing a non-parametric Cumulative Sum (CUSUM) calculation which is more appropriate for investigating a mind boggling system condition than a parametric calculation. Their approach appeared high recognition exactness with low computational overheads. Assaults including subnet caricature IP tends to can be additionally distinguished by this model. Yet, their exploratory outcomes demonstrated that the recognition delay was in the vicinity of 10 and 127.3 seconds which isn't tasteful as far as the recognition delay for an ongoing location framework. Feinstein proposed a factual discovery model to distinguish DDoS assaults by processing entropy and recurrence arranged dispersions of

particular IP properties. The entropy could be figured through various successive bundles called a sliding window of a settled width. They executed an entropy demonstrate as a module for Snort and performed tests to approve it in different system follows. Be that as it may, the extent of a sliding window, a tunable parameter was not advanced.

They set it from their experiential information. In addition, they were not worried about subnet ridiculed assaults in the tests. The previously mentioned approaches don't

fulfill a few noteworthy necessities which ought to be accomplished in DDoS identification methodologies, for example, low handling overheads, a short identification postponement and high location rates. Our model can be worked in an ongoing system condition as a result of an upgraded movement framework, which is developed by a lightweight hash work. Additionally, a parcel based variable time window empowers our model to recognize DDoS assaults inside a brief period of time. The parameters for identification are advanced by GA and it ensures high discovery rates. Furthermore, our proposed model can distinguish DDoS assaults including subnet ridiculed and moderately low transmission capacity assaults. The points of interest of our proposed recognition demonstrate are displayed in the following area.

2.26 ICMP

Internet Control Message Protocol (ICMP) Internet Control Message Protocol (ICMP) bundles are intended for organize administration highlights, for example, finding system hardware and deciding the number of bounces or round-trip-time to get from the source area to the goal. For example, ICMP$_E CHO_R EPLY bundles(" ping")enablethe client to send a de man$

2.27 Amplification Attacks

A DDoS enhancement assault is gone for utilizing the communicated IP address include found on most switches to intensify and mirror the assault . This element permits a sending framework to indicate a communicate IP address as the goal address instead of a particular address. This trains the switches overhauling the bundles inside the system to copy the parcels and send them to all the IP addresses inside the communicate address extend. For this kind of DDoS assault, the aggressor can send the communicate message straightforwardly, or the assailant can utilize the operators to send the communicate message to expand the volume of assaulting movement. In the event that the assailant chooses to send the communicate message straightforwardly, this assault gives the aggressor with the capacity to utilize the frameworks inside the communicate arrange as zombies without expecting to invade them or introduce any operator pro-

gramming. We additionally recognize two kinds of enhancement assaults, Smurf and Fraggle assaults.

2.28 Smurf Attacks

In a DDoS Smurf attacks, the assailant sends bundles to a system intensifier (a framework supporting communicate tending to), with the arrival deliver caricature to the casualty's IP address. The assaulting parcels are commonly ICMP ECHO REQUESTs, which are bundles (like a "ping") that demand the recipient to create an ICMP ECHO REPLY bundle. The enhancer sends the ICMP ECHO REQUEST bundles to the greater part of the frameworks inside the communicate address go, and each of these frameworks will restore an ICMP ECHO REPLY to the objective casualty's IP address. This compose of assault opens up the first parcel tens or many circumstances.

2.29 Fraggle attack

A DDoS Fraggle attack is like a Smurf attack in that the assailant sends parcels to a system speaker. Fraggle is unique in relation to Smurf in that Fraggle employments UDP ECHO parcels rather than ICMP ECHO parcels. There is a variety of the Fraggle assault where the UDP ECHO parcels are sent to the port that backings character age (chargen, port 19 in Unix frameworks), with the return deliver satirize to the casualty's reverberate benefit (reverberate, port 7 in Unix frameworks) making an endless circle. The UDP Fraggle bundle will focus on the character generator in the frameworks came to by the communicate address. These frameworks each create a character to send to the resound benefit in the casualty framework, which will resend a reverberate parcel back to the character generator, and the procedure rehashes. This assault produces significantly more awful movement and can make significantly more harming impacts than only a Smurf assault.

2.30 DDoS Agent Setup

There are both dynamic and aloof techniques that assailants use to introduce malignant code onto a optional casualty framework with a specific end goal to execute a DDoS assault in either the Specialist Handler or the IRC-Based DDoS assault models. Dynamic strategies commonly include the aggressor checking the organize for frameworks with known vulnerabilities. After recognizing such powerless frameworks, the aggressor runs contents to break into the framework. Once the aggressor has broken into the framework, he can stealthily introduce the DDoS Agent programming. In this manner the framework is traded off as a optional casualty, which can be utilized as a zombie in a future DDoS assault. Detached techniques normally include the assailant sharing degenerate records or building sites that exploit known vulnerabilities in an optional casualty's web program. Upon getting to a document or site with an installed DDoS Agent, the auxiliary casualty framework is bargained, and the DDoS operator code might be introduced.

2.31 Active DDoS Installation

Before propelling a DDoS assault, aggressors should first set up the DDoS assault organize. They frequently run a checking device to recognize potential optional casualty frameworks. One basic instrument aggressors use to check for ports is a product program called Nmap. Assailants can download Nmap from different areas on the web (for instance www.insecure.org/nmap/). This instrument permits aggressors to choose scopes of IP delivers to filter. The apparatus will then continue to look through the Web for every one of these IP addresses. Nmap restores the data that every IP address is broadcasting, for example, TCP and UDP ports that are open, and the particular OS of the filtered framework. An aggressor would then be able to look at this rundown for potential auxiliary casualty frameworks. Another device for examining the system discovers irregular IP addresses with a known weakness. This furnishes the aggressor with a rundown of casualty frameworks that all offer the same OS Supported Unix Linux Solaris Windows Lie and Pause Effectively Survey DDoS Software Instrument

Specialist Initiation Technique Assault Network Correspondence ClientHandler HandlerAgent None Handler âĂŞ Agent IRC Based Truly, Private or Mystery Channel No, Open Channel No Rootkit Cradle Flood Trojan Stallion Program Dynamic Scanning Programming/ Indirect access Weakness Specialist Setup Irritated Site Debased Document Establishment Detached Yes Convention TCP UDP ICMP Encryption normal helplessness. One case of this kind of helplessness filter device is Nessus

2.32 Attack Network Communication

2.33 Protocols Used

The DDoS specialists and handlers can impart to each other by means of TCP, UDP, and additionally ICMP. DDoS handlers and customers can likewise speak with each other utilizing a similar convention alternatives.

2.34 Encrypted Communication

Some DDoS assault apparatuses have likewise been created with help for encoded correspondence inside the DDoS assault arrange. Operator handler DDoS assaults may utilize an encoded channel either between the customer and the handlers, or between the handlers and the operators. The strategy for encryption for operator handler DDoS assaults will be subject to the correspondence convention utilized by the DDoS device. IRC-based DDoS assaults may utilize either an open, private, or mystery channel to impart between the specialists and the handlers. Both private and mystery IRC channels give encryption, however private channels (not the information or clients) show up in the IRC server's channel rundown and mystery directs don't show up in the IRC server's channel list.

2.35 Agent Activation

There are two key techniques for the DDoS specialists to end up dynamic. In some DDoS instruments, the specialists effectively survey the handlers or IRC channel for guidelines, though in different DDoS apparatuses, the specialists will lie and sit tight for correspondence from either the handler or the IRC channel.

2.36 OS Supported

DDoS assault instruments are commonly intended to be perfect with various working frameworks (OS). Any OS framework, (for example, Unix, Linux, Solaris, or Windows) may have DDoS operator or handler code intended to chip away at it. Ordinarily, the handler code is intended to help an OS that would be situated on a server or workstation at either a corporate or ISP site. This generally prompts the decision of Unix, Linux, or Solaris. For the operator code, it is likewise regular for it to be good with Linux or Solaris with the expansion of Windows. Numerous aggressors target private Internet clients with DSL and link modems (for higher assaulting data transmission) and these clients ordinarily utilize Windows. Specialist and handler programming can be composed for various working frameworks.

2.37 DDoS Attack Software Commands

Each DDoS assault device has various charges. These summons are intended for both the handler and operator programming bundles. For most DDoS operators, particular summons are entered by means of a charge line interface. In spite of the fact that the particular summons vary for each DDoS assault apparatus.

2.38 System Service Providers

One system right now being talked about is for suppliers and arrange overseers to add dynamic estimating to their system use, to empower auxiliary casualties to end up more dynamic in keeping themselves from ending up some portion of a DDoS assault. On the off chance that suppliers charged contrastingly for the utilization of various assets, they could charge for access to specific administrations inside their systems. This would enable the suppliers to just permit authentic clients on to their systems. This framework would make it simpler to avert assailants from entering the system [35]. By modifying the estimating of administrations, auxiliary casualties who might be charged for getting to the Internet may turn out to be more aware of the activity they send into the system and henceforth may complete a superior occupation of policing themselves to confirm that they are most certainly not partaking in a DDoS assault.

2.39 Detect and Neutralize Handlers

One vital strategy for ceasing DDoS assaults is to identify and kill handlers. Since the operator handler DDoS assault instruments require the handler as a go-between for the assailant to start assaults, finding and ceasing the handlers is a fast technique to closure the assault. This can perhaps be finished by concentrate the correspondence conventions and activity designs between handlers what's more, customers or handlers and specialists keeping in mind the end goal to distinguish organize hubs that may be tainted with a handler. Additionally, there are typically far less DDoS handlers conveyed than there are operators, so closing down a couple of handlers can render various specialists futile. Since operators frame the center of the assailants capacity to wage an assault, killing the handlers to keep the aggressor from utilizing them is a compelling system to ruining DDoS attacks.

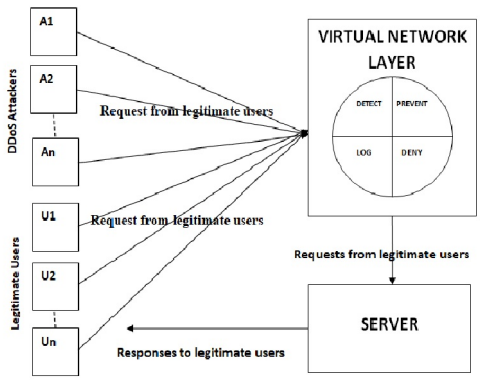
CHAPTER 3

OVERVIEW OF THE PROPOSED SYSTEM

3.1 DDoS Prevention Mechanism

The overview of proposed mechanism is shown in Fig. 3.1. A virtual network layer is created on a server to prevent the DDoS attack. Any interactions with the server have to pass through virtual network layer. The attack detection mechanism is plugged in the virtual network layer to detect the incoming DDoS attack depending upon the threshold limit allotted to each user. In this paper we consider three different way of spamming the service requests that are 1. Attacker uses its own bots/machines to spam service requests 2. Attacker uses a fake IP addresses generated by multiple bots to spam the service requests. 3. Attacker steals the IP address of a legitimate user to spam the service requests. Similar to detection mechanism, the prevention mechanism is also integrated with the virtual node. The prevention mechanism cuts the connection between the virtual node and server to prevent any data being stolen or lost or any resource being exhausted. The connection is temporarily terminated and the connection is refreshed after a time period. Even the connection is refreshed after a time period, the IP address is marked in blacklist node

Figure 3.1: Overview Of Proposed System



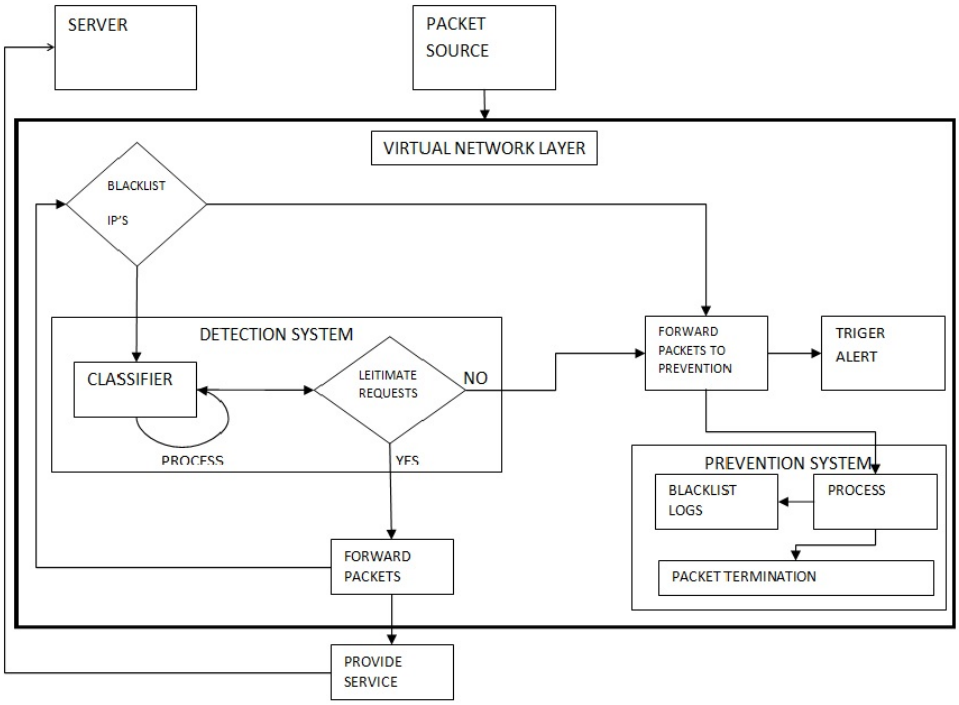
CHAPTER 4

SYSTEM DESIGN

4.1 Proposed System

The architecture of the proposed system is shown in Fig 4.1. The major process of the proposed system is Detection, Prevention and Recovery phase which prevent the damage caused by the DDos TCP FLOOD attacks. With respect to the different way of DDoS attack mentioned in section 3, we propose the solutions for the design requirements listed below. 1. The victim node doesnâĂŹt know which one of the ways is used by the attacker to spam the service request.. 2. If the attacker spoofs the IP address of legitimate user, the victim cannot ignore the particulare IP address because it beongs to legitimate user. Instead it can be banned for a limited time. Banning an IP address that belong to a legitimate user for a limited time would not be a noticable issue comparable with damage caused by the attackers. 3. The victim cannot affort the network being attacked. The following steps are followed in order to implement the detection, prevention and recovery pahse of the proposed system that considers the above mentioned design requirements. 1. Create a virutal network layer : A virtual network layer is created that would eventually act as a decoy to prevent the DDos attacks. Any interactions with the real system will have to pass through the virtual layer, which would give us an instinct when the actual system would come under attack. 2. Integrate a DDos Detection mechanism with the virtual network layer : In this stage, we plug in a Detection mechanism in the virtual network the was created. This part of the network is responsible to detect the incoming attacks depending upon the threshold limit alloted to each user. This would further allow the system to take preventive measures. 3. Taking a preventive measure against the DDos attacks: Once the attack is detected by the virtual network, immediately the preventive measures are toggled by the system. There are more than one way in which this could work since there are 3 different cases that an attacker might spam requests.

Figure 4.1: Proposed System



Similar to the detection mechanism, this is also intergrated with the virtual network.Since we would not be able to predict on which one of the above mentioned cases the attacker would initiate, we would block any requests that exceeds a threshold limits. Considering the possiblity that the attacker uses a legitimate users IP address, it would be impossible to block a address permanently. This leads us to set a ban time on a pirticular set of IP addresses, later sending an notification that such suspecious activity was found from their respective IP addresses. This would enable the user to tighten their security and possible give a good insight on the attacker of further invistigation was processed. On this point, the connection between the virtual network and the Actual netowrk is cut preventing any data being stolen or data loss. However, this is a case that is highly unlikely to happen and in case this happens the priority of keeping the data safe than that of running the service without an interruption comes first. Here, we realise that there might be cases that the attack is going out of hands and the prevention mechanish set in the virtual network can no longer with hold the pressure. Thus the connection between the virtual and the Actual network would eventually be cut until the virtual network is refreshed.

4.2 Detection Phase

During the detection phase, the detection sub-system collects the incoming packets within a time frame, for example 60 seconds. The collected packets are subjected to a blacklist check to test whether their sources are blacklisted as attackers of the cloud system. If the packet source is listed in the attacker blacklist, the detection system will send the packets directly to the prevention sub-system without further processing. If the packet source is not blacklisted, the incoming packet will be passed to the classier to decide whether the packets are normal (originating from a client) or abnormal (originating from an attacker). A packet is considered to be an attacking one if the source requests connections to the same destination more frequently than an assumed threshold. The threshold can be manually adjusted by the system administrator to cater for the varying requirements of a particular network. If a packet is considered to be normal, the detection system will send it to its destination (the cloud service provider). Otherwise, the detection sub-system will send the packet to the prevention sub-system.

4.3 Prevention Phase

When the packets reach the prevention system, they are considered to be attacking packets by the detection sub-system. The prevention subsystem alerts the system administrator of the attacks. Then, the prevention sub-system will add the attacking source address to the attacker blacklist used by the detection sub-system, if it is not already on the list. Finally, the attacking packet will be dropped. By the detection sub-system, if it is not already on the list. Finally, the attacking packet will be dropped.

4.4 Recovery Phase

The connection between the virtual network and the Actual netowrk is terminated preventing any data being stolen or data loss, which also prevents any damage to the hardware caused by the Attack.However, this is a case that is highly unlikely to happen and in case this happens the priority of keeping the data safe than that of running the

service without an interruption comes first. The termination of connection between the server and the virtual layer occurs only when the attack is going out of hands and the prevention mechanish set in the virtual network can no longer with hold the pressure and protect the system. Thus the connection between the virtual and the Actual network would eventually be cut until the virtual network is refreshed. This would result in a temperory delay in service but the hardware and data are safe and enough time is bought to counter further incoming attacks.

4.5 Algorithm Used To Detect/Prevent DDoS

1. Create a log file in which the number of the incoming requests say N, which is to be compared to the attacks detected in the previous stage where the data are stored in DDoS Capture Attack Pattern (DCAP) DCAP (DDos captured attack pattern database), that logs the source IP(SIP), destination IP(DIP) and the time.

2. Let Fn = f1, f2 , âĂęâĂęâĂę, fn be the flow identifier, and fn be the request flow form the host, where n = 1, 2, 3âĂęâĂęâĂę. The flow fn may have N number of requests to the server.

3. Let R be the total number of requests obtained from all the flow identifiers within the threshold time T.

4. Let Ts be the threshold value per second which determines the maximum number of requests that can be handled by the server Attack Qualifier Condition (AQC) AQC = R(1/TS)

5. If the AQC (Attack qualifier condition) âL'ě 1 then all the flow will be considered as a suspicious attack flow, update the logfile for suspicious flow.

6. Generate Feature Vector, which contines the time interval (ti âĂŞ ti-1) and the number of requests within the time interval (ti âĂŞ ti-1) 1.e. Feature Vector (FV) FV= (ti âĂŞ t1-1), NR where NR is the number of requests.

7. Create a general attack vector (GAV) General Attack Vector (GAV) from a database DCAP, which contines time interval and the number of requests within the

time interval.

8. Cclculate the similarity measure E using normalized absolute distance between the GAV and FV using, Threshold (Ts)

E=[(GAV)-(FV)]/FV

9. If E>T$_G AV$,

where T is the maximum threshold lying between 0 to 1, then the flow contains DDOS attack. Else there is no DDos attack in the flow.

10. Update the logfile by setting the ADS(attacks detected). If ADS is set to Y then the SIP(source ip) are considered blacklist.

11. If E<1 then do not terminate the program, else if E>1 terminate the program, ban the connection for the source ip(SIP) for temporary time (Tt), wherer T$_G AV = (0to1)$.

12. Create a logfile of the entire activity, and input new data() for the upcoming process.

Figure 4.2: Prevention Phase

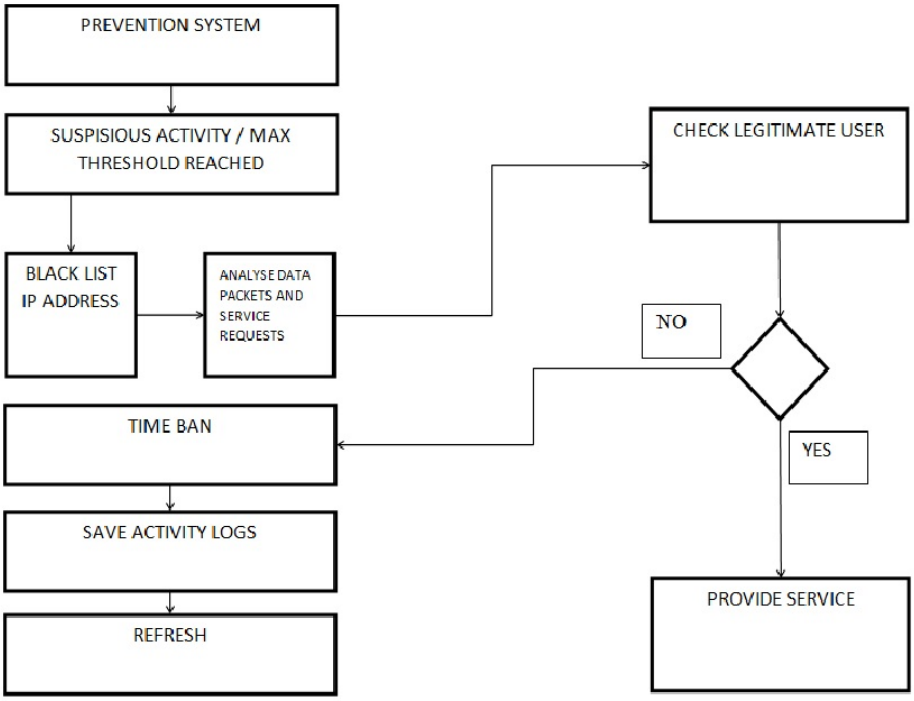
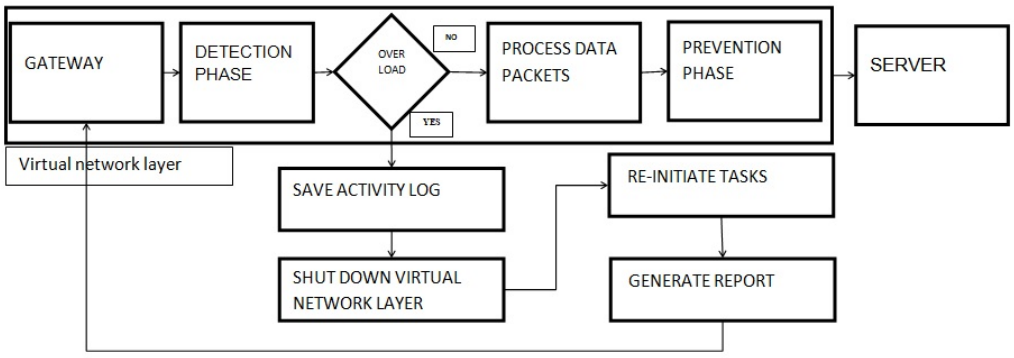


Figure 4.3: Recovery Phase



CHAPTER 5

CODING, TESTING

5.1 Log File Creation

```
<html>
    <head>
    </head>
    <body bgcolor="b3d9ff">
    <?php
    include_once("dbconnect.php");
    $db = new Database();
    $db-> connect();
    $sql1 = "delete from logs";
    $res1 =db->deleteData($sql1);
    if ($fh = fopen('log1.txt','r'))
    //$i = 0;
    while (!feof($fh))
    $line = fgets(fh);
    //echo $line." <br> ";
    $words = explode("",line);
    //echo $words[1];
    $f = 0;
```

foreach($wordsas$w)

if ($f == 1$)

//$sourceip$ =w;

$f = 0$;

$sip = explode($" : ",w);

$SourceIP$ =sip[0];

//echo "
Source IP: " . $SourceIP$;

$localIP = getHostByName(getHostName())$;

//echo $localIP$;

$sql =$ "$insertintolog_svalues($'SourceIP', '$localIP'$,'YES',''$)";

//echo sql;

//exit();

$result$ =db->insertData(sql);

if ($w ==$ "$from$")

$f = 1$;

//i =i+1;

//if ($i == 15$)

//break;

fclose(fh);

//mysql$_close$(conn); ?>

<h1>Prevention of DDoS TCP Flood Attack Using Virtual Network Layer</h1>

<hr>

```
<br><br>
<h2>Suspicious Log Created Successfully!!!</h2>
<h2>
<br><br><br>
<a href="index.html"> Back to main page</a><br><br>
</h2>
<br><br><br><br><br><br><hr>
<b><i>Developed by Prasanna V Balaji, Beniel Dennyson</i></b>
<hr>
</center>
</body>
</html>
```

5.2 Blacklist Check

```
<html> <head> </head> <body bgcolor="b3d9ff">
<center> <br> <h1>Prevention of DDoS TCP Flood Attack Using Virtual Network Layer</h1>
<br>
<hr>
<h2>Blacklist Log Check</h2>
<h2>
</center>
<br>
<a href="index.html"> « Back</a><br>
```

</h2>

<?php

include$_once("dbconnect.php");$

$db = new Database();$

$db->connect();$

$sql = "SELECT * FROM log_s";$

$result =$db->selectData($sql);$

$count = mysqli_num_rows($result);$

if ($count == 0$)

echo "<h3 style='color:ff0000'>No incoming request!!!</h3>";

else if($count > 0$)

echo " <table cellspacing='3' cellpadding='3' border='1px'

style='border-collapse:collapse;'>";//border: bbd2d0;

echo "<tr align='center'><th>Source IP</th><th>Destination

IP</th><th>Blacklist</th><th>Status</th></tr>";

$slno = 0;$

while($row = mysqli_fetch_assoc($result)$)

// $DutyArr = array();$

$slno =$slno+1;

//echo "<tr align='center'>";

//echo "<td>". $DutyArr['did']." </td> ";

//echo "<td>". $slno." </td> ";

//echo "<td>". $row['Sen_Order']." </td> ";

// echo $row['SourceIP']." : ".row['DestinationIP']. "
";

//echo "<td align='left'>". $row['Staff_Name']." < /td > ";

$sql1 = "SELECT * FROM blacklist where SourceIP =' ".row['SourceIP']."' and DestinationIP='".$row['DestinationIP']."'";

//echo $sql1;

$res1 =db->selectData($sql1);

$count1 = mysqli_num_rows(res1);

if ($count1! = 0)

echo "<tr><td>". $row['SourceIP']." < /td >< td > ".row['DestinationIP']. "</td><td>YES</td><td>Access Denied</td></tr>";

else

echo "<tr><td>". $row['SourceIP']." < /td >< td > ".row['DestinationIP']. "</td><td>No</td><td>Access Allowed</td></tr>";

//mysql_close(conn);

echo "</table> ";

?>

</body>

</html>

5.3 Attack Analysis

<html>

<head>

</head>

<body bgcolor="b3d9ff">

<h1>Prevention of DDoS TCP Flood Attack Using Virtual Network Layer</h1>

<hr>

<h2>Attack Analysis</h2>

<h2>

 « Back

</h2>

<?php

include$_once("dbconnect.php");$

$db = new Database();$

$db-> connect();$

$sql = "SELECT * FROM log_s";$

$result =$db->selectData($sql);$

$count = mysqli_num_rows($result);$

if ($count == 0$)

echo "<h3 style='color:ff0000'>No incoming request!!!</h3>";

else if($count > 0$)

echo " <table cellspacing='3' cellpadding='3' border='1px' style='border-collapse:collapse;'>";//border: bbd2d0;

echo "<tr align='center'><th>Source IP</th><th>Destination IP</th><th>Attacker</th><th>Statu

$slno = 0;$

while($row = mysqli_fetch_assoc$(result))

// $DutyArr = array();

$slno =$slno+1;

//echo "<tr align='center'>";

//echo "<td>". $DutyArr['did']." < /td > ";

//echo "<td>". $slno." < /td > ";

//echo "<td>". $row['Sen_Order']." < /td > ";

// echo $row['SourceIP']." : ".row['DestinationIP']. "
";

//echo "<td align='left'>". $row['Staff_Name']." < /td > ";

$sql1 = "SELECT*FROMddos_datasetwhereSourceIP =' ".row['SourceIP']."
' and DestinationIP='".$row['DestinationIP']."'";

//echo $sql1;

$res1 =db->selectData($sql1);

$count1 = mysqli_num_rows(res1);

if ($count1! = 0)

echo "<tr><td>". $row['SourceIP']." < /td >< td > ".row['DestinationIP'].
"</td><td>YES</td><td>Access Denied, Blacklist Log Updated</td></tr>";

$sql2 = "insertintoblacklistvalues('".row['SourceIP']."','" . $row['DestinationIP']."')";

$res2 =db->insertData($sql2);

else

echo "<tr><td>". $row['SourceIP']." < /td >< td > ".row['DestinationIP'].
"</td><td>No</td><td>Access Allowed</td></tr>";

//mysql_close(conn);

echo "</table> ";

?>

</body>

</html>

5.4 Data set Comparison

<?php

//$host = "localhost"; //Hostname$

//$username = "root"; //Mysqlusername$

//$password = ""; //Mysqlpassword$

//$db_name = "dbexammaster"; //Databasename$

//// Connect to server and select databse.

//mysql$_connect$("host", "$username$","password")or die("cannot connect");

//mysql$_select_db$("db$_name$")$or die$("$cannot select DB$");

?>

<?php this script contains the class for database connection and passing queries ?>

<?php

class Database

public $host$;

public $user$;

public $password$;

public $database$;

public $connection$;

public id;

public id_no;

public row;

function $_{c}onstruct()$

$this- > host = "localhost"$;

$this- > user = "root"$;

$this- > password = "server"$;

$this- > database = "ddos"$;

//function to connect to database

function connect()

// $this- > connection = mysqli_{c}onnect$(this->host,$this- > user$,this->password,$this- > database)ordie(mysql_{e}rror())$;

// //mysql$_{s}elect_{d}b()ordie(mysql_{e}rror())$;

$this- > connection = mysqli_{c}onnect$(this->host, $this- > user$,this->password,$this- > database)ordie("cannotconnect")$;

//mysql$_{s}elect_{d}b$(this->database) or die("cannot select DB");

//function to insert data in database

function insertData($insertQuery$)

return mysqli$_{q}uery$(this->connection,$insertQuery$);

function updateData($updateQuery$)

return mysqli$_{q}uery$(this->connection,$updateQuery$);

//function to select data from database

function selectData($selectQuery$)

return mysqli$_{q}uery$(this->connection,$selectQuery$);

function deleteData($deleteQuery$)

return mysqli$_{q}uery$(this->connection,$deleteQuery$);

function beginTrans()

mysqli$_{q}uery$(this->connection,"SET AUTOCOMMIT=0");

mysqli$_{q}uery$(this->connection,"START TRANSACTION");

function commitTrans()

mysqli$_{q}uery$(this->connection,"COMMIT");

function rollbackTrans()

mysqli$_{q}uery$(this->connection,"ROLLBACK");

//function to disconnect the current connection from database

function disconnect()

mysqli$_{c}lose$(this->connection);

?>

5.5 DDoS Database

– phpMyAdmin SQL Dump

– version 4.1.12

– http://www.phpmyadmin.net – – Host: 127.0.0.1

– Generation Time: Apr 23, 2018 at 12:25 PM

– Server version: 5.6.16

– PHP Version: 5.5.11

SET SQL$_{M}ODE = "NO_{A}UTO_{V}ALUE_{O}N_{Z}ERO"$;

SET time$_{z}one = " + 00 : 00"$;

/*!40101 SET @OLD$_{C}HARACTER_{S}ET_{C}LIENT = @@CHARACTER_{S}ET_{C}LIENT*$
/;

/*!40101 SET @OLD$_{C}HARACTER_{S}ET_{R}ESULTS = @@CHARACTER_{S}ET_{R}ESULTS*$
/;

/*!40101 SET @OLD$_{C}OLLATION_{C}ONNECTION = @@COLLATION_{C}ONNECTION$
/;

/*!40101 SET NAMES utf8 */;

–

– Database: ‘ddos‘

–

– ————————————————————

–

– Table structure for table ‘blacklist‘

–

CREATE TABLE IF NOT EXISTS ‘blacklist‘ (

‘SourceIP‘ varchar(30) NOT NULL,

‘DestinationIP‘ varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

–

– Dumping data for table ‘blacklist‘

–

INSERT INTO ‘blacklist‘ (‘SourceIP‘, ‘DestinationIP‘) VALUES

(’10.1.107.217’, ’10.1.107.39’),

– ————————————————————

–

– Table structure for table ‘ddos$_{d}ataset$‘

–

CREATE TABLE IF NOT EXISTS ‘ddos$_dataset$‘(

‘SourceIP‘ varchar(30) NOT NULL,

‘DestinationIP‘ varchar(30) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

–

– Dumping data for table ‘ddos$_dataset$‘

–

INSERT INTO ‘ddos$_dataset$‘(‘$SourceIP$‘, ‘$DestinationIP$‘)$VALUES$

(’10.1.107.217’, ’10.1.107.39’);

– ————————————————————

–

– Table structure for table ‘log$_s$‘

–

CREATE TABLE IF NOT EXISTS ‘log$_s$‘(

‘SourceIP‘ varchar(30) NOT NULL,

‘DestinationIP‘ varchar(30) NOT NULL,

‘Suspicious‘ varchar(10) NOT NULL,

‘Attacker‘ varchar(10) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

–

– Dumping data for table ‘log$_s$‘

–

INSERT INTO ‘log$_s$‘(‘$SourceIP$‘, ‘$DestinationIP$‘, ‘$Suspicious$‘, ‘$Attacker$‘)$VALUES$('10.1.

/*!40101 SET CHARACTER$_SET_CLIENT = @OLD_CHARACTER_SET_CLIENT*
/;

/*!40101 SET CHARACTER$_SET_RESULTS = @OLD_CHARACTER_SET_RESULTS*
/;

/*!40101 SET COLLATION$_CONNECTION = @OLD_COLLATION_CONNECTION*
/;

5.6 Blacklist Refresh

<html>

<head>

</head>

<body bgcolor="b3d9ff">

<?php

include$_once(”dbconnect.php”);$

$db = newDatabase();$

$db- > connect();$

$sql1 = ”deletefromblacklist”;$

$res1 =$db->deleteData($sql1);$

?>

<h1>Prevention of DDoS TCP Flood Attack Using Virtual Network Layer</h1>

<hr>


```
<h2>Blacklist Log Cleared!!!</h2>

<h2>

<br><br><br>

<a href="index.html"> Back to main page</a><br><br>

</h2>

<br><br><br><br><br><br><hr>

<b><i>Developed by Prasanna V Balaji, Beniel Dennyson</i></b>

<hr>

</center>

</body>

</html>
```

CHAPTER 6

PERFORMANCE ANALYSIS

Figure 6.1: Home Page

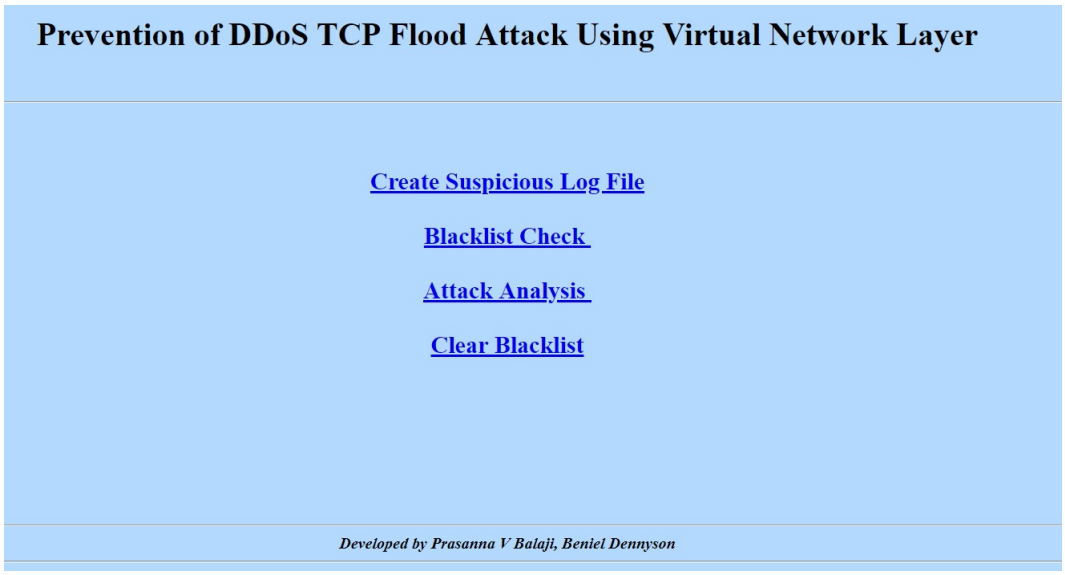


Figure 6.2: Create Suspicious Log File

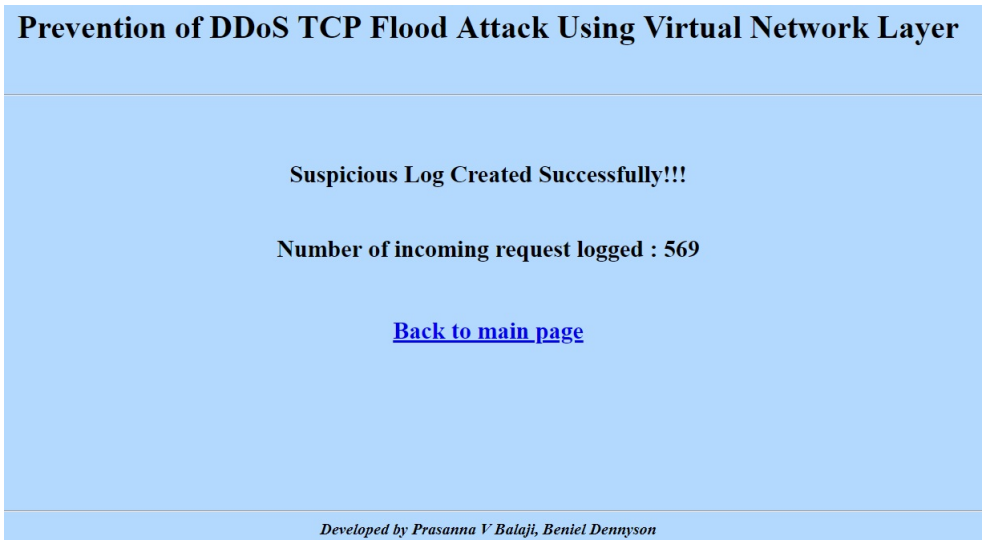
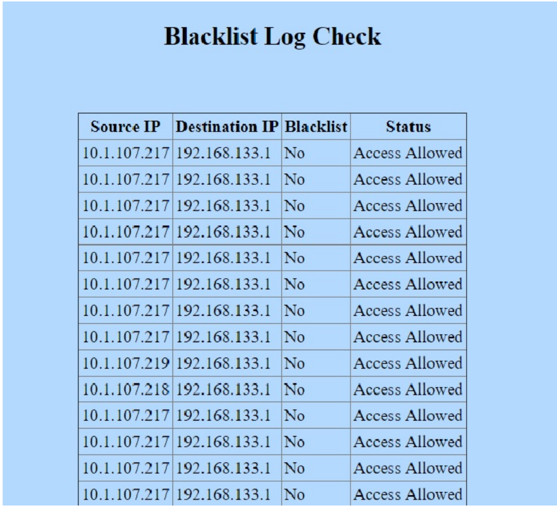


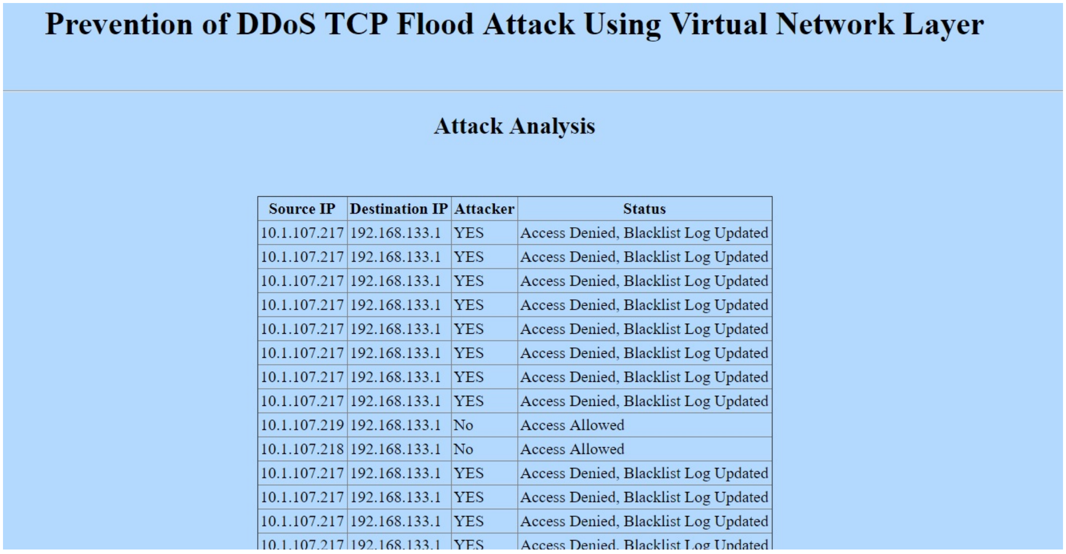
Figure 6.3: Blacklist Check



Blacklist Log Check

Source IP	Destination IP	Blacklist	Status
10.1.107.217	192.168.133.1	No	Access Allowed
10.1.107.217	192.168.133.1	No	Access Allowed
10.1.107.217	192.168.133.1	No	Access Allowed
10.1.107.217	192.168.133.1	No	Access Allowed
10.1.107.217	192.168.133.1	No	Access Allowed
10.1.107.217	192.168.133.1	No	Access Allowed
10.1.107.217	192.168.133.1	No	Access Allowed
10.1.107.217	192.168.133.1	No	Access Allowed
10.1.107.219	192.168.133.1	No	Access Allowed
10.1.107.218	192.168.133.1	No	Access Allowed
10.1.107.217	192.168.133.1	No	Access Allowed
10.1.107.217	192.168.133.1	No	Access Allowed
10.1.107.217	192.168.133.1	No	Access Allowed
10.1.107.217	192.168.133.1	No	Access Allowed

Figure 6.4: Attack Analysis



Prevention of DDoS TCP Flood Attack Using Virtual Network Layer

Attack Analysis

Source IP	Destination IP	Attacker	Status
10.1.107.217	192.168.133.1	YES	Access Denied, Blacklist Log Updated
10.1.107.217	192.168.133.1	YES	Access Denied, Blacklist Log Updated
10.1.107.217	192.168.133.1	YES	Access Denied, Blacklist Log Updated
10.1.107.217	192.168.133.1	YES	Access Denied, Blacklist Log Updated
10.1.107.217	192.168.133.1	YES	Access Denied, Blacklist Log Updated
10.1.107.217	192.168.133.1	YES	Access Denied, Blacklist Log Updated
10.1.107.217	192.168.133.1	YES	Access Denied, Blacklist Log Updated
10.1.107.217	192.168.133.1	YES	Access Denied, Blacklist Log Updated
10.1.107.219	192.168.133.1	No	Access Allowed
10.1.107.218	192.168.133.1	No	Access Allowed
10.1.107.217	192.168.133.1	YES	Access Denied, Blacklist Log Updated
10.1.107.217	192.168.133.1	YES	Access Denied, Blacklist Log Updated
10.1.107.217	192.168.133.1	YES	Access Denied, Blacklist Log Updated
10.1.107.217	192.168.133.1	YES	Access Denied, Blacklist Log Updated

Figure 6.5: Clear Blacklist

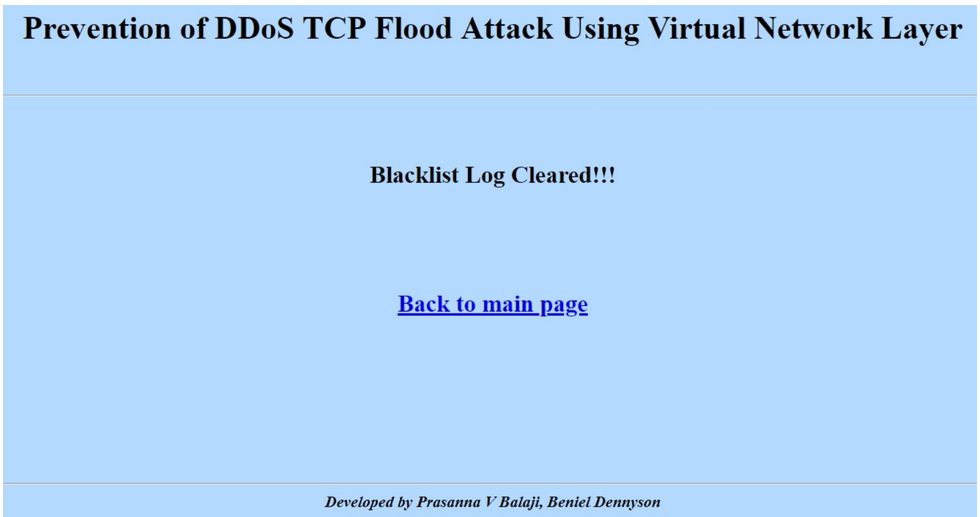
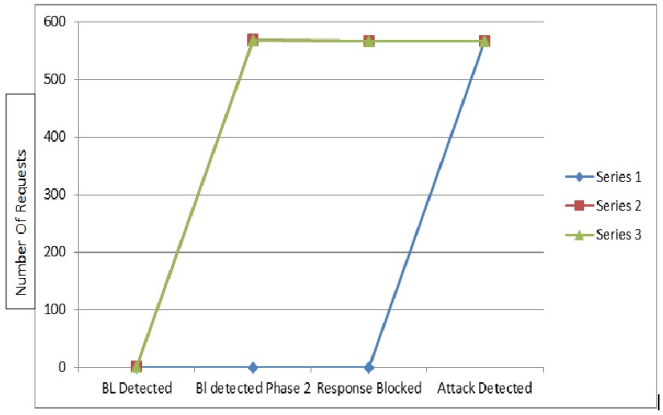


Figure 6.6: Graph against Blocked Requests



CHAPTER 7

CONCLUSION

This paper proposes a mechanism that prevents the DDoS attack and increases the availability of services and resources to the legitimate nodes. Since all the requests generated by the legitimate nodes and the attackers are analyzed by the virtual machine set on the victim node and the only the service requests from the legitimate nodes are allowed to pass to the victim node, the damages caused by DDoS attack is reduced.

CHAPTER 8

FUTURE ENHANCEMENT

Due to the growth of Internet of Things (IOT) and Wireless Sensor Network (WSN) the usage of electronic devices for basic needs have been increased drastically. This leads those devices to be vulnurable to various types of attacks such as DDoS.

Implementation of DDoS Prevention mechanism in every communication devices providing Defence against such attacks would be a major priority in Future.

CHAPTER 9

REFERENCE

1. AQEEL SAHI, DAVID LAI, YAN LI AND MOHAMMED, An Efficient DDoS TCP Flood Attack Detection and Prevention System in a Cloud Environment (DIYKH)

2. B. Wang, Y. Zheng, W. Lou, and Y. T. Hou, "DDoS attack protection in the era of cloud computing and software-de ned networking," Comput. Netw., vol. 81, pp. 308 319, Apr. 2015.

3. (DEFCON, The SHMOO Group, 2011. (http://cctf. shmoo.com/)

4. K. Kendall, A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems, MasterâĂŹs The-sis, MIT, 1999.

5. Kejie Lu, Dapeng Wu, Jieyan Fan, Sinisa Todorovic, Antonio Nucci, âĂIJRobust and efficient detection of DDoS attacks for large-scale internetâĂİ, Computer Networks, Vol. 51 (2007), pp. 5036âĂŞ5056.

6. LORI L. DELOOZE, Applying soft computing techniques to intrusion detection, Applying soft computing techniques to intrusion detection, University of Colorado at Colorado Springs Colorado Springs, CO, USA Âl'2005

7. Monowar H. Bhuyan, Dhruba K. Bhattacharyya, and Jugal K. Kalita,Towards generating real life Datasets for network intrusion detection May 9, 2015 (International Journal of Network Security

8. Nathalie Weiler. âĂIJHoneypots for Distributed Denial of ServiceâĂİ, Enabling Technologies: Infrastructure for Collaborative Enterprises, 2002. WET ICE 2002. Proceedings. Eleventh IEEE International Workshops, 2002. pp. 109-114. 2002.

9. Neeta sharma , Mayank singh , Anuranjan Mirsha, Prevention against DDos attack using triple pass filter 2016 (International Conference on Computing for Sustainable Global Development)

10. Paul J. Criscuolo. âĂIJDistributed Denial of Service Trin00, Tribe Flood Network, Tribe Flood Network 2000, And Stacheldraht CIAC-2319âĂİ. Department of Energy Computer Incident Advisory Capability (CIAC), UCRL-ID-136939, Rev. 1., Lawrence Livermore National Laboratory, February 14, 2000

11. P. E. Ayres, H. Sun, H. J. Chao, and W. C. Lau, "ALPi: A DDoS defense system for high-speed networks," IEEE J. Sel. Areas Commun., vol. 24, no. 10, pp. 1864 1876, Oct. 2006.

12. Sunny Behal, Krishan Kumar,Trends in Validations of Ddos Research, International Conference on Computational Modeling and Security (CMS 2016)

13. Saman Taghavi Zargar, Member, IEEE, James Joshi, Member, IEEE, and David Tipper, Senior Member, IEEE, A Survey of Defense Mechanisms Against Distributed Denial of Service (Ddos) Flooding attacks.

14. Sang Min Lee a , Dong Seong Kimb,c,âĹŮ , Je Hak Lee a , Jong Sou Park,Detection of Ddos attacks using optimized traffic matrix (science direct 2012)

15. Sang Min Lee, Dong Seong Kimb, Je Hak Lee, Jong Sou Park, âĂIJDetection of DDoS attacks using optimized traffic matrixâĂİ, Computers and Mathematics with Applications, Vol. 63 (2012), pp. 501âĂŞ510

16. Stephen Specht. Ruby Lee sspecht@princeton.edu rblee@princeton.edu. Department of Electrical Engineering. Taxonomies of Distributed Denial of Service. Networks, Attacks, Tools, and Countermeasures. Princeton Architecture Laboratory for Multimedia and Security. Technical Report CE-L2003-03.

17. V. Jacobson, C. Leres, and S. McCanne, âĂIJThe tcpdump manual page,âĂİ Lawrence Berkeley Laboratory, Berkeley, CA, 1989.

18. Wei Zhou, Weijia Jia, Sheng Wen, Yang Xiang, Wanlei Zhou, âĂIJDetection and defense of application-layer DDoS attacks in backbone web trafficâĂİ, Future generation computer system, Vol. 38(2014), pp. 36-46.