Honeynetbased Defensive mechanism Against DDoS Attacks

Abhinav Dahiya[1], Kamaldeep Joshi[1], Rainu Nandal[1], Rajkumar Yadav[2], Satinder Bal Gupta[2]

[1]Department of Computer Science and Engineering, University Institute of Engineering and Technology, Maharshi Dayanand University, Rohtak, Haryana, India

[2]Department of Computer Science and Engineering Indira Gandhi University, Meerpur, Rewari

Corresponding Author: *rajyadav76@rediffmail.com*

ORCID iD: 0000-0002-3096-5407, 0000-0002-3238-0234, 0000-0003-0350-0388, 0000-0003-0605-8759, 0000-0002-6056-1489

Research Paper Received: 05.02.2020 Revised: 08.05.2020 Accepted: 01.06.2020

Abstract—Internet we are using today is expanding faster than we could have imagined. Since the dawn of the Internet, there has been an exponential increase in the number of web sites and so the quantity of data on these websites. The hackers attack the web sites to collect useful information and also to make other legitimate users devoid of the information or services they required. Such problems and other similar types of attacks can be handled by “Honeypot” system which takes all attacks on itself and studies the attack patterns to detect similar kind of attacks in future.Honeypots allow all the attacks on itself and make attackers think that they have the access of real system and meanwhile honeypots will study all the attack pattern of attackers. The authors have created a network of various Honeypots to enhance the efficiency. Before honeypots, a filtering algorithm is used which with the help of pre-defined sink server will predict whether a given packet is malicious or not, here help of ISP service provider can also be taken if sink server doesn’t have any information about the sender of given data packets. Then to further enhance the capability of honeynet cloud, a various different type of services can be deployed at honeynet clouds like HTTP, CBR and FTP. Here, the authors have used NS2 simulator to run the proposed work and the results are taken in the form of graphs like throughput of all three different types of honeypots, bandwidth and packet loss of all services provided by destination servers. Detection rate of malicious packets are calculated and comparison has be done between different services provided by honeynet cloud.

Keywords—DDoS, Honeypot, Security, NS2, Sink-Server, Intruder, Honeynet-cloud

1. Introduction

The quantity of gadgets associated with PC systems and the Internet is developing quickly day by day. This has prompted an expansion in the quantity of system based assaults. Private data of each individual utilizing web like PAN (Permanent Account Number), AADHAAR number which is a 12 digit unique number provided by UIDAI (Unique Identification Authority of India) and CREDIT/DEBIT card details has been put away in extensive databases and so forth. Vital records, reports and photographs are put away on clouds and drives which keeps this information on an enormous database in centralizing form. This expansion is trailed by a surging amount of surveillance issues. Advanced dangers

and susceptibility are discovered each time, and networking system is a long way from being secure. Various latest advancements are used to stop these attacks. About 76% of Indian businesses were hit by cyber-attacks in 2018, the highest after Mexico and France, according to a study by Sophos, a global leader in network and endpoint security. India is the at first position among the countries facing dictionary attacks followed by China. Dictionary attack is a brute force method where an attacker uses all dictionary words to create a password. India holds the 7th position in comment spammer countries while China tops followed by US, Russia and Ukraine. Comment spam is a term referencing a wide category of spammer posting or spambot which use social sites, forums, blogs, wikis etc to post unsolicited things in through any media. We cannot rely wholly on IDS and firewalls to keep the data completely secured. Firewalls are regularly installed around the border of a network so as to stop unapproved approach by penetrating particular ports and data. They can easily block all incoming request in order to block illegal request but also block some genuine requests in this process. An intrusion detection system can likewise be utilized to dissect approaching request however because of its ”false alert issues”, they are very little in use in case they are the only layer of security [1].

With such a tremendous number of problems, we need a mechanism to identify these assaults. One such protection system is the utilization of honeypots. A honeypot is a crucial security entity used for sacrificing its asset to research unapproved accesses to so as to find potential vulnerabilities in operational frameworks and eliminate the danger. These are like traps for suspecting user. Honeypot is installed on a network to attract the attackers. It gathers the information from the trapped user and used for future attacks as shown in Figure 1.

A decent method to explore new dangers is to catch the noxious movement well-ordered as it enters a system. It merits seeing the answers attackers give in the deadlock like situation, for example, contacting other aggressors or transferring different trojan like rootkits. As time goes on, we get different Honeypots with specific functions, like shadow honeypots, honey farms and honey-tokens. These different forms will be discussed in the literature review later in section II.

The honeypot technique can be used in various areas like IDS that is industrial control system, which collects data and information from various attacks and smartphones. If it is mixed along with Intrusion Detection System (IDS) and firewall, it handles the false negative and false positive rate and also adds another layer of Security. Also, it is compatible with encryption or communication through IPV6, not like other securities.

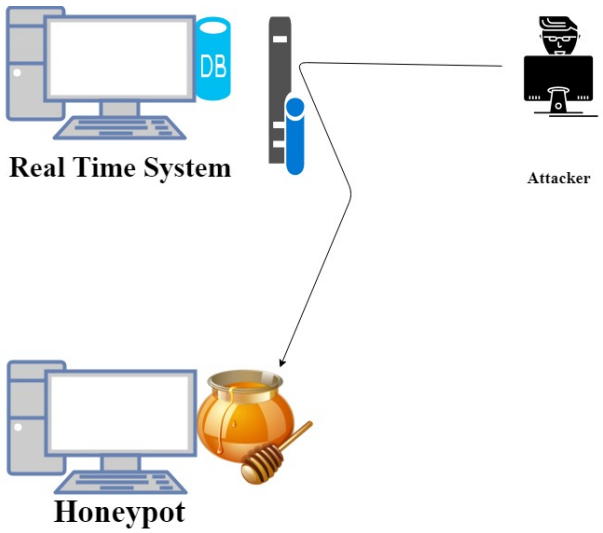

Figure 1. Basic structure of Honeypot

In the solutions proposed so far by researchers, work had been done in improving the detection and prevention rate. Architectures have been proposed to improve the efficiency as well as the overall throughput of the network. Honeypot is added as

an additional security layer along with the Intrusion Detection System (IDS). There is not much work present where a honeynet cloud is explored with different kinds of services. In the proposed scheme, honeynet cloud is employed with multiple subnets which deal with different kinds of traffic flows like HTTP, FTP and DNS etc. In this paper, the authors aim to deal with DDoS attacks by utilizing honeypots and honeynet cloud. The authors have contributed in the existing state of the art of this domain by adding a new service which will provide the sink server more information about the attackers and their attacking method. Further sections are arranged as follows: section II will discuss some significant paper in this domain which makes us work in this direction followed by section III which contains the research work done. Section III will describe experimentation and result from analysis followed by section IV which finally will conclude the paper.

1.1. Research Gap/Problem statement

In previous papers, work had been done in improving the detection and prevention rate. Architectures have been proposed to improve the efficiency as well as the overall throughput of the network. Honeypot is added as an additional security layer along with the Intrusion Detection System(IDS). Researches related to various services within in honeynet are very limited. For example, various servers at honeynet to handle attacks related to various services like HTTP, FTP etc can be added. Their bandwidth along with throughput comparisons are very limited.

1.2. Objectives

The specific aim of a honeypot system is to limit the DDoS attacks by transferring all attacks towards itself so that no attacker will not be able to access the destination servers. This paper studies the various services at honeypot and in order to do so following are its objectives:

1. To study the previous literature work related to honeypot, honeytoken and honeynet.
2. To find the detection rate of attacks at the system.
3. To add a new service which will be paid in nature which will provide the sink server with additional information about the attackers and their attacking method.
4. To compare the bandwidth and throughput of all three services at Honeynet that is HTTP, FTP and CBR.

2. Literature Survey

In this section, the authors have discussed some of the important schemes related to their work. i.e. the researches which had use honeypot technology to combat DDoS attacks.

In [2], Brown et. al. have carried out a study considering different cloud platforms, for example, Amazon EC2, Windows Azure, IBM SmartCloud along with honeypot to analyze different attack packets. USA and China are the countries which carry out HTTP and SSH based DDoS attacks predominantly. But this study was bounded to EC2 and Azure. Low interaction honeypot was mainly focused on the proposed approach.

In [3], Buvaneswariet. al. have utilized IHoneycol as an incentive provider to local ISPs in order to combat DDoS attacks effectively. This whole framework consists of Firecol-IPS and Honeypot-IPS that diverted DDoS attack traffic near to source and destination respectively. Twin attack and ping of death attacks are efficiently handled by the proposed approach. This protocol is lightweight but has high

computational overhead.

In [4], an intrusion detection system has been designed for a cloud environment using honeypot technology for reducing the false alarm ratio. Implementation rules have been built by this technique where brokers accessed the data send by cloudlets. The attack is detected by the honey gateway that is deployed at each echelon of cloud nodes. The main disadvantage of this scheme is that implementation at each OSI layer, its speed gets decreased.

In [5], ADTRVH (Ant-based DDoS detection technique using roaming virtual honeypots) has been proposed by the authors. Authors have used ant colony optimization algorithm to control the traffic flows in the victim network. Pheromone deposit has been used to detect the frequency of DDoS attacks and tracing back the attackers to their actual IP addresses. Database of attack signatures and log files are regularly updated as the new attack signatures have been found in the network.

In [6], authors have analysed the incoming and outgoing data traffic from a particular network. They have utilized the inbound and outbound ratio to find any discrepancy in the data traffic flows. Some rule-based defensive mechanisms have been permitted to estimate the occurrence of DDoS attack. Monitoring tools deployed by the proposed scheme results in the generation of some mitigation rules. Conjunction and disjunction of data packets parameters have been used to filter out the malicious data traffic from the intended traffic towards a specific server. This fact leads to inefficiency of the proposed approach as it cannot be fitted into every network state.

In [7] Zargar et al. figures out various insider threats in any enterprise using raw log s and traffic.This solution is not fully comprehensive which only detects the deviation of attacker from normal behaviour, and then create a system generated alert. This particular way can be used with high rate along with network session. This method has a defiancefor very large number of flows of network from many hosts to single host.

Bercovitch.et.al [8] gives a honeytoken generating tool which is automatic in nature to create fake data elements in the database having three phases: Rule extraction fetches the rules from databases so that the fake data items look real, then in second phase different honeytokenwere generated on the basis of rule extracrionphase known as Honeytoken generation phase and after that rating is given to newly generated honeytoken on the basis of its similarity to real data items.

In [9] Asaf.et.al carried out the study in two phases. In the first phase,generic methods are used to create the honeytoken which are quite similar to real life dataset. In the second phase, the authors carried out the study to show that although the honeytoken are implanted in databases but nature of user doesn't change.

3. Research Work

In this section, the authors have discussed the proposed honeypot based defensive framework against DDoS attack. A two layer defensive framework utilizing honeypot has been proposed in this paper. DDoS attack is a cumulative malicious attempt by a network of compromised machines to make an online service completely unavailable for legitimate users. Consequences of a DDoS attack could be very devastating to an organization as an attacker doesn't need many resources to perform a DDoS attack. But a few hours of downtime can make a huge loss to a victim. Honeypots have been used for years by researchers to combat DDoS attacks. Honeypots are nothing but the potential victims which can offer more vulnerabilities and loopholes to an attacker and can act as trap to lure cyber attackers. Honeypots provide information to the defenders about

Table 1
Evolution of Honeypot Against DDoS attacks

YEAR	INNOVATION	TECHNOLOGY USED
1990-91	"HONEYPOT" term coined	Honeypot term was made in public domain for the first time through two books "The Cuckoo's Egg" [10] and," An Evening with Berferd in Which a Cracker Is Lured, Endured, and studied" [11].
1997	First ever public honeypot i.e. "Deception toolkit"	First, ever public honeypot was created by Fred Cohnen [12].
1998	Honeypot by the administration of US	Honeypot created by Martin Rash for the U.S. administration [13]
	CybercopSting	First commercial honeypot [13].
	BackOfficer Friendly	The launch of Backoffice Friendly Honeypot [14]
1999	Honeynet Project	Lance Spitzner along with a team of 50 members founded a no profit research group [15],
2000-2001	Adopted by organizations.	Honeypot was adopted by various big organizations to tackle the attackers and many worms [13].
2002	Solaris Honeypot	Exploits like dtspcd were detected by Solaris Honeypot [13]
	Honeyd	Nugatory daemon honeypot [16] formed.
	Honeynet	High level research interaction honeypot [17], [18].
	Honeynet against DDoS	Honeynet technology against Distributed DoS [19].
2003	Honeytoken	A new concept of honeytoken was coined [20].
	Eyeore	A new Honeywall CDROM Eyeore [21].
	Mirage	Coequal to snort against DDos [22].
2004	Roaming Honeypot	Introduced by Khattab to curb DDoS [23].
2005	Roo	Honeywall CDROM Roo.
2006	Hybrid protean honeypot	Combination of hybrid and protean based honeypot to handle DDoS [24].
2007	Server-client honeypot	Creation of honeypots for client-server architecture [25].
2008	HTTP based attack	Honeypots for attacks based on HTTP [14].
2010	Glastopf	Creation of dynamic and low interaction honeypot [19].
	Architecture having two level	Created by Sardana to curb dual level architecture [26].
2011	Honydroid	A high interaction honeypot [27].
2012	Cloud environment honeypots	Brown study the employment of honeypots on cloud computing like windows azure and Amazon EC2 [2].
2013	IHoneycol	Combination of honeypot and firecol. [3]
	Hostage	A low interaction honeypot for mobiles [28].
	Nomadic	A mobile phone honeypot concept [29].
	Labsac	A virtual honeypot network for Android [30].
2014	Attack detection in cloud	Honeypot use in Cloud environment.
2015	Ampot	Ampot used against DDoS [31].
	Honeymesh	Prevention of DDoS attacks in virtualized honeypot [32].
	Shadow honeypot for wireless access point	Here shadow honeypot has the extra advantage of anomaly detection [33]
2016	DDoS detection on Ant base	Detects DDoS attacks using roaming virtual Honeypots [5].
	Honeyphy	Construct honeypots for cyber physical system [34].
	Honeymix	An intelligent honeypot [35].
	IAAS infrastructure cloud	Combination of honeywall, honeycomb and honeyd in IAAS [36].
2017	Privacy issues in honeynet and honeypot.	EU laws were used to check to protections of information used by honeypots and honeynet [37].
	URL redirections	Honeypots that checks each URL redirection [38].

the number and type of attempts an attacker has made in order to infiltrate a network. A honeypot is configured in such a way that it always seems exploitable to an attacker. In the following section, various components of the proposed architecture will be discussed along with their functionalities.

3.1. Proposed Model's Architecture

In this section, various communicating units and their functionalities have been illustrated. Figure 2 shows the proposed architecture of an organization that want to defend its interests against DDoS attacks. Each module and its sub modules are discuusedin details in the following sub-sections.

3.1..1 Ingress Filtering Module

Ingress filtering module is deployed at the edge router of the network. Filtering module tries to filter malicious packets from the data traffic flow before entering into the organization's network. Basically, filtering module tries to stop DDoS attacks at the boundary of the network. Filtering module consists of a data log which constantly and timely been updated with new attack signatures or patterns. Every data packet requested by a user is compared against the information stored in data log. Ingress filtering is a defensive technique which is used to handle IP spoofing. Ingress filtering is used to ingoing data flow traffic (that tends to enter the victim network from other networks) IP spoofing is a technique used by attacker to forge the IP address of the data packet to hide the original source or to impersonate other sender. IP spoofing enables attackers to not being detected by the defenders and to confuse target servers between malicious data packets and legitimate ones. Using IP spoofing for carrying out DDoS attack is a very old technique to bypass the security systems deployed by the organization. Moreover, IP spoofing along with hierarchical architecture of the botnet makes very difficult for organizations to detect the original attacker. Figure 2 shows the architecture of proposed scheme. Filtering module consists of sub-modules like fetching module, analysis module, traffic rerouting module and behavioural analysis module which will be discussed in the following sub sections.

a) **Fetching Module**
Fetching module is the first sub module of the filtering mechanism. It tends to fetch the IP headers of the incoming data packets. IP header of every packets contains very important information about the sender or sender's network. Attackers need to forge only the IP header fields in order to spoof its IP address. Therefore, in this module IP headers of data packets are fetched and given as input to the next sub module.

b) **Collating Module**
Collating module works on the IP headers fetched in the previous module. It collects important information from the IP address like source's IP address, port number, offset field, destination IP address, port number, packet size etc. This information is used to detect new attack signatures and log server can be constantly and timely updated according to this information. With new advancements in technology and techniques attackers are always one step ahead of us. They are highly incentivized to carry out new types of DDoS attacks. Solutions proposed so far by researchers can detect DDoS attack that already exists. They are not trained to detect new types of DDoS attacks.

c) **Traffic Rerouting Module**
Traffic rerouting module diverts the legitimate traffic to the intended destination while illegitimate ones are diverted towards the honeynet

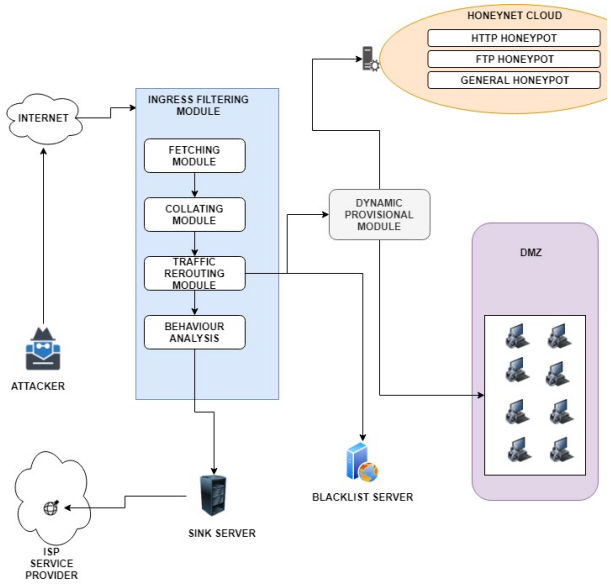

Figure 2. Architecture of proposed scheme

and sink node. Honeynet is deployed by the organization to lure attackers and consists of a number of open vulnerabilities. Different traffic load levels have been embedded with a wide range of threshold values. If the threshold value is within the control of defensive mechanism than all the traffic is diverted towards the server. If the threshold is in betweenSceptical and Alert level, then the traffic is diverted towards the honeynet. Threshold values must be set granularly as it can make a system oscillate between different states often resulting into instability of the defensive mechanism.

– **Threshold Value**: The process of calculating threshold values is done by monitoring data traffic during non-attack period and with the help of ISP. There could be many parameters of calculating threshold value but here in this paper two parameters have been considered :"Frequency of data packet sent" and "Size of data payloads". Average value of frequency with which request for data is sent and payload of data request packet, is calculated at non-attack period with the help of ISP. To further make it robust, threshold is divided in three level : **Normal**, **Sceptical** and **Alert**. If the frequency or payload(size) of data packets sent is considerably high, then its threshold value will be put under Sceptical level. But if the value of both frequency and size of data packets is high then its threshold value will be put under Alert level. If threshold value of incoming packet starts getting Sceptical, it will further investigate the request with the help of ISP. If on further investigation, gets on Alert level, it will be marked as illegitimate and will be sent to Honeynet Cloud and blacklist and sink servers will be updated.

d) **Behavioral Analysis**
Values of parameters that have been considered by an organization to handle DDoS attacks are calculated by this module. These values are then stored in the log server for future references and further mitigation.

3.2. Sink Node

Sink node handles the data traffic marked as illegitimate which cross the maximum threshold value set and cannot be handled by the server. All the traffic irrespective of malicious and non-malicious will be send towards sink node. A time window has been set by the sink node and stores all the information about the malicious packets. A blacklisted server has been used to store all the information. All the malicious IP addresses are stored in the blacklisted server. This blacklisted server is constantly updated as the data packets arrived at the sink node.

3.3. Honeynet Cloud

Honeynet cloud is composed of heterogeneous honeynets that can handle various TCP, UDP and HTTP data packets. Different numbers of honeypots are deployed to construct a honeynet. The data request from user passed through various sub modules has reached to honeynet through data request mapping module. IP address of each honeypot is constantly changed by the dynamic provisioning module to make it more real to the attacker and to avoid being detected by attacker. Fingerprinting is the major problem caused due to static IP address of the honeypot. It is very important to change IP address of the honeypot regularly to confuse the attacker and make it appear more real to the attacker. Data traffic is diverted towards the honeynet if the threshold value lies within the Sceptical and Alertlevel.

3.4. Data Request Mapping Module

Data request mapping module is deployed between filtering module and honeynet. It is the responsibility of this module to send data requests from users to their right destination according to the algorithm discussed below. In this algorithm, the matrix value is used. If its value is 0 then it means it illegitimate packet and if 1 then legitimate packet. All packets are made to gone through this algorithm , this algorithm check the source IP address from packets with database on sink server which has history of all data attacks. This module decides which honeynet sub network is going to receive a particular data packet or a packet will be sent to server or sink node.

3.5. Dynamic Provisioning Module

Dynamic provisioning module has functionality to constantly change the IP addresses of each honeypot present in honeynet cloud. This is being done to befool the attacker and make honeypots appear more real systems to attackers. This technique is employed to overcome the fingerprinting problem which is caused due to static IP addresses of honeypots. It is very necessary to change the IP addresses regularly to perplex the attacker and diverts his attention from the real server.

3.6. Demilitarized Zone (DMZ)

It is the best location for a Destination or other servers to be deployed in a network. Internal and external placements of honeypots have their own respective disadvantages. This is the placement point which shields the disadvantages of both kinds of placement. Every organization consists of this zone. It listen service request from the rest of the Internet. A proved legitimate packet is only headed towards this zone for further processing.

3.7. ISP service provider

ISP service provider is attached to the sink server. When collating module collects the IP address from the packet, algorithm checks it in the sink server, if it not found as illegitimate IP address, sink server may consult ISP for double check. This service will be no free of cost, as we are asking ISP to check into its database for the history of IP address of incoming packets.

4. Experimentations and Results

In this section, the authors have discussed the simulation performed and results obtained. For that, first of all study of the tool on which the simulation have been performed is discussed. Then, the parameters that have been considered while

Ingress Filtering Algorithm

```
Input: PACKET P_K
#AT INGRESS FETCHING MODULE
BEGIN:
MATRIX[P_K] == 0  //set value for packet Metric as null or 0
HEADER OF PACKET P_K WILL BE FETCHED USING FETCHING MODULE
FETCH (P_K.HEADER);

#AT SINK SERVER
If (P_K(SOURCE_IP_ADDRESS) || P_K (MAC_ADDRESS) == SUSPICIOUS_IP || SUSPICIOUS_MAC)
//If source IP address or mac address of a system belongs to blacklist IP address in SINK SERVER and ISP SERVICE PROVIDER

    MATRIX[P_K]==0; // illegitimate
Else
    MATRIX[P_K]==1;  // Not illegitimate

If(P_K.(SOURCE_IP_ADDRESS) || P_K (MAC_ADDRESS) € SUSPICIOUS_IP || SUSPICIOUS_MAC && P_K_SINK_VALUE == 0)
MoveTo_HONEYNET(P_K);
Else
MoveTo_DMZ (P_K);
End
MOVE_HONEYNET (P_K)
    If (HEADER_HTTP_i[k] ==true)
MOVE_P_K HTTP_HONEYPOT;
    Else if (HEADER_FTP[k] ==true)
MOVE_ P_K FTP_HONEYPOT;
    Else if (HEADER_UDP[k] ==true)
MOVE_P_K GENERAL_HONEYPOT;
```

performing simulation and after that whole simulation process is discussed. The authors have used i.e. Ns2 for performing simulation. Ns2 stands for network simulator version 2. Ns2 is an event driven open source simulating tool used for research in communication network domain specifically. It used to study dynamic behaviour of the computer and communication networks. NAM is network animator which is GUI of Ns2. In the proposed model, the authors deals with volume based DDoS attacks. There does not exist any standards or benchmarks to guide researchers on how to take parameters for analysing performance of a defensive model against volume based DDoS attack. There are two factors which form the basis of parameters considered i.e. low variation in attack traffic or high variation in attack traffic. Choosing suitable parameters to evaluate the performance of any scheme is a very important task. Legitimate data traffic values must be collected during non-attack period to maintain the repository of log files and to update attack signatures. These log files can be used further to have comparative analysis during the attack or after the attack period. So, following parameters have been considered while analysing the performance of the model:

a) Illegitimate packet drop rate: A DDoS attack defense mechanism tends to reduce the number of illegitimate packets from a traffic flow by discriminating them from the legitimate packets selectively. It is defined as the ratio of total illegitimate packets dropped to the total traffic flow.
b) Benign Packet drop rate: We want defense mechanism to not only defend against DDoS attack but provide QoS to the benign users. It is defined as the ratio of total legitimate packets reached to destination to the total traffic flow.
c) Throughput: It reflects performance of the system. It the total amount of data packets transmitted in a unit time.
d) Failure rate: This metric is defined at the application layer which is defined as ratio of number service requests unattended (not processed) by the victim server to the total number of service requests sent by sender. While performing the simulation, the authors have taken total 15 nodes as shown in Figure 3. Each of the nodes represents either the clients or servers. Of the total 15 nodes there are 6 client nodes, 3 are legitimate nodes and 3 are attacker nodes. Attacker nodes are represented with respective

attacker label and legitimate nodes are labelled as legitimate nodes. The 3 legitimate nodes shown are for different applications protocols like HTTP or FTP etc. Sixth node is ingress filtering modulewhich filters the incoming data flow from respective legitimate nodes and attacker nodes.

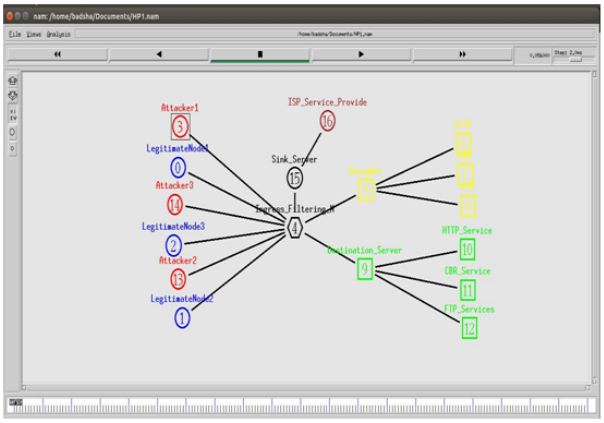

Figure 3. Initial topology of proposed scheme

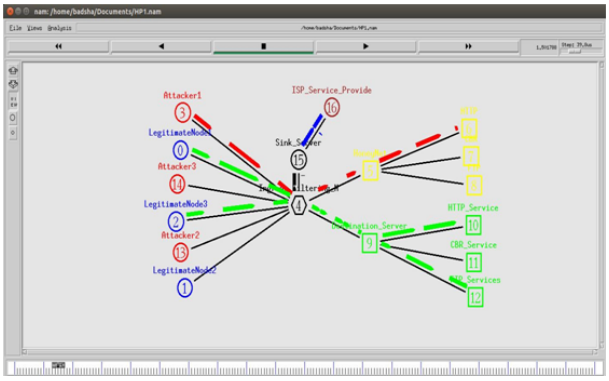

Figure 4. System state of proposed scheme when attacker node also requests for HTTP service.

Seventh node is Sink server which is attached to ingress filtering module which helps it to differentiate between attack and legitimate traffic. Eighth node is destination node which is further attached to HTTP server node, FTP server node and CBR server node. Further, honeynet cloud is attached to different types of honeypot i.e. HTTP honeypot etc.All nodes are connected by wired connections. All connections are duplex in nature where two way communications is possible at a time.

In the simulation, the authors have used distance vector (DV) Routing protocol. This routing protocol is used in wired communication system. In DV, each node sends periodic route updates for every 2 seconds. The simulation has 18 seconds duration. The complete analysis and output will be within the timeframe of 18 seconds. Within this simulation time both attacker node and legitimate node will send data and will be handled by the proposed scheme.

In Figure 4, node 3 (attacker1) also started sending data packets. Ingress filtering module again will ask to Sink server which will check its database filled with recent attack history. Sink server with the help of ISP will send ACK as negative, then filtering module will transfer the attacker node 3 data packets towards honeynet where honeynet will give the data packet to honeypot dealing with HTTP requests and it will start analysing it. During this complete duration, data packet from legitimate node 0will keep communicating with HTTP service without any problem. In Figure 5, queue at link between node 4 to node 5 start getting overwhelmed. Node 4 is receiving data packets more than it could handle. Only attacker packets are getting dropped at this moment as node 4 to node 5 is for only attack packets which are being sent to honeynet clouds for attack packet analysis. At this point a very small data packet drop can be seen on link node 4 to 9.

4.1. Graphs Description

After the simulation gets complete, the results of the simulation in the form of graphs have been discussed.

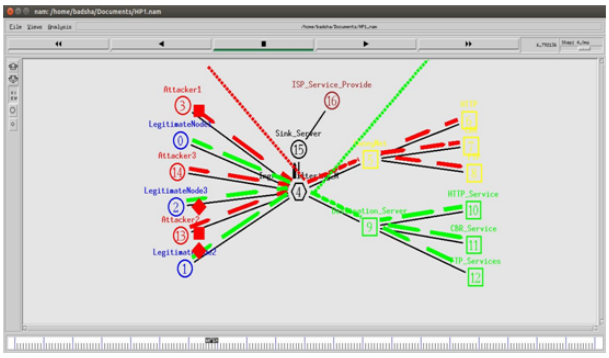

Figure 5. System state of proposed scheme when attack packet drop is significantly higher for HTTP service.

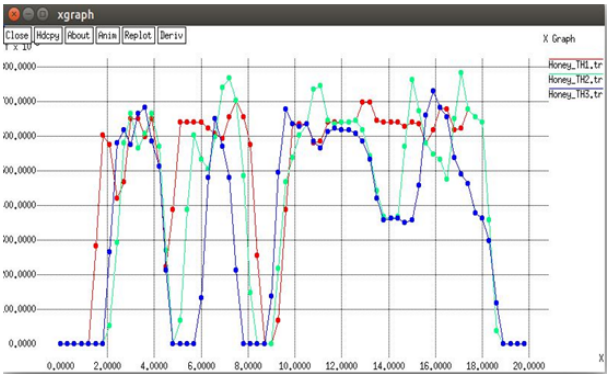

Figure 6. Throughput of all three honeypots in honeynet.

In Figure 6, net throughput of all three honeypot in the honeynet are shown. Graph curve HoneyTH1.tr of red color shows the throughput of node number 6 which is a Honeypot for HTTP services. Graph curve HoneyTH2.tr of green color shows the throughput of node 7, a Honeypot for CBR services. Graph curve HoneyTH3.tr of blue color shows the throughput of node number 8 which is a Honeypot for FTP services.

In Figure 7, net bandwidth of all three services at the destination servers are shown. Graph curve HP1.tr of red color shows the bandwidth of node number 10, a server for HTTP services. Graph curve HP2.tr of green color shows the bandwidth of node

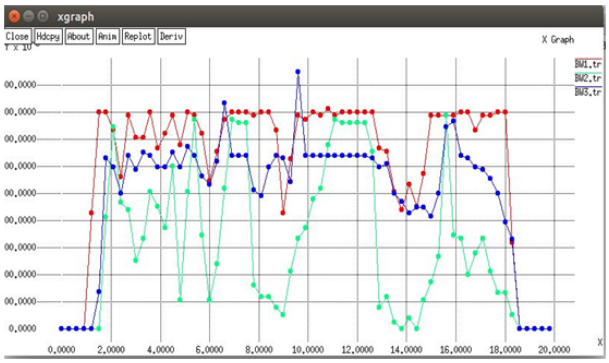

Figure 7. Bandwidth of all three services at the destination server.

number 11, a server for CBR services. Graph curve HP3.tr of blue color shows the bandwidth of node number 12, a server for FTP services.

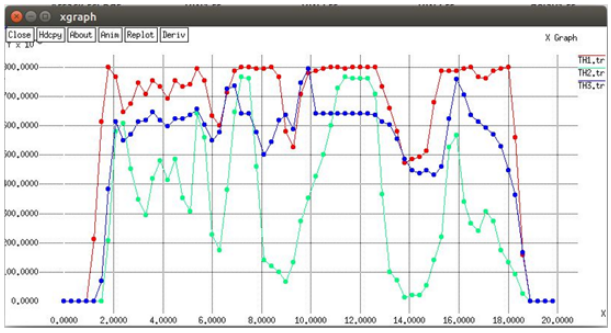

Figure 8. Net throughput of all three services at the destination server.

In Figure 8, net throughput of all three services at the destination servers are shown. Graph curve TH1.tr of red color shows the throughput of node number 10 which is a server for HTTP services. Graph curve TH2.tr of green color shows the throughput of node number 11, a server for CBR services. Graph curve TH3.tr of blue color shows the throughput of node number 12, a server for FTP services.

In Figure 9, net packet loss of all three services in the destination servers are shown. Graph curve Loss1.tr of red color shows the packet loss of node number 10 which is a server for HTTP services.

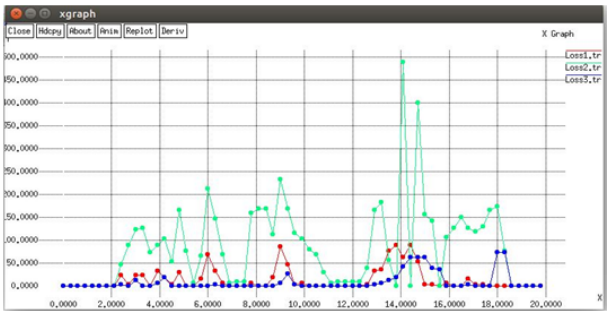

Figure 9. Packet loss rate of all three services at the destination servers

Graph curve Loss2.tr of green color shows the packet loss of node number 11, a server for CBR services. Graph curve Loss3.tr of blue in color shows the packet loss of node number 12, a server for FTP services.

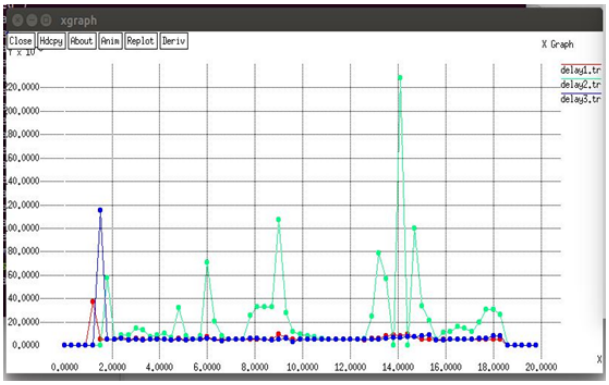

Figure 10. Packet delay rate of all three services at the destination servers.

In Figure 10, packet delay rate of all three services at the destination servers are shown. Graph curve delay1.tr of red color shows the packet delay rate of node number 10 which is a server for HTTP services. Graph curve delay2.tr of green color shows the packet delay rate of node number 11, a server for CBR services. Graph curve delay3.tr of blue color shows the packet delay rate of node number 12, a server for FTP services.

5. Conclusion

In order to evaluate the results of the emulation, the authors use ns2 simulations. It is performed to calculate the detection rate of attacks and comparison of various services provided by destination servers in the form of graphs in different time windows. Simulation time is taken 18 seconds here. There is a heterogenous network of honeynet cloud as well as of destination servers. Three types of services are provided by the honeynet cloud. In the simulation, only one node is taken for each of the services. Detection rate of the simulation can be calculated using trace file tracefile.tr which is an output file for recording the data of the simulation. Here detection rate is for the illegitimate packets sent by attacker to create the DDos and to curtail the Quality of service. It shows the ingress filtering module provides us with the detection rate of around 76% and when the proposed method is implemented without the heterogenous network of honeynet cloud, it happens to be 63%. Detection rate is improved due to introduction of new component in the proposed architecture that is ISP.

References

[1] G.C. Tjhai, C. Gina, M. Papadaki, S. M. Furnell, L. Nathan. “Investigating the Problem of IDS False Alarms: An Experimental Study using Snort”, *In IFIP International Information Security Conference, Springer*, Boston, MA, pp. 253-267. 2008 October 1949.

[2] S. Brown, R. Lam, S. Parsad, S. Ramasubramanian, J. Slauson. “Honeypots in the Cloud”, *University of Wisconsin-Madison*, Vol.11, 2012.

[3] M. Buvaneswari, T. Subha. “Ihoneycol: a collaborative technique for mitigation of DdoS attack”, *International Journal of Emerging Technology and Advanced Engineering*, Vol.3, pp. 176-179, January 2013.

[4] R. Meghani, S. Sharma. “Security from various Intrusion Attacks using honeypots in cloud”, *International Journal of Emerging Technology and Advanced Engineering*, Vol.4, pp. 468-473, May 2014.

[5] S. Rajalakshmi, V.M. Kuthadi, T. Marwala. "Ant-based distributed denial of service detection technique using roaming virtual Honeypots", *IET Communications*, Vol.10, pp. 929-935, 19 May 2016.

[6] M. Aupetit, Y. Zhauniarovich, G. Vasiliadis, M. Dacier, Y. Boshmaf. "Visualization of actionable knowledge to mitigate DRDoS attacks", *In 2016 IEEE Symposium on Visualization for Cyber Security (VizSec)*, pp. 1-8, Octomber 2016.

[7] A. Zargar, A. Nowroozi, R. Jalili. "XABA: A zero-knowledge anomaly-based behavioral analysis method to detect insider threats", *In 2016 13th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC)*, pp. 26-31, September 2004.

[8] M. Bercovitch, M. Renford, L. Hasson, A. Shabtai, L. Rokach, and Y. Elovici. "HoneyGen: An automated honeytokens generator", *Proceedings of 2011 IEEE International Conference on Intelligence and Security Informatics*, pp.131–136, 2011.

[9] A. Shabtai, M. Bercovitch, L. Rokach, Y. Gal, Y. Elovici, and E. Shmueli. "Behavioral study of users when interacting with active honeytokens", *ACM Transactions on Information and System Security (TISSEC)*, Vol.18, No.3, pp. 1-21, 2016.

[10] C. Stoll. "The cuckoo's egg: tracking a spy through the maze of computer espionage", *New York: Pocket Books Nonfiction*, 2005.

[11] B. Cheswick."An evening with Berferd in which a cracker is lured, endured, and studied", *AT&T Bell Laboratories*, 1991.

[12] http://www.all.net/dtk. "Deception toolkit", *Open Sourse*, Latest access time for the website is 20 March 2019.

[13] L. Spitzner. "Honeypots: tracking hackers", *Addison-Wesley*,Vol.1,2003.

[14] A. Ahmad, M. Ali, and J. Mustafa. "Benefits of honeypots in education sector", *International Journal of Computer Science and Network Security*,Vol.11, pp.24-28, October 2011.

[15] https://www.honeynet.org/blog/4. "The Honeynet project", *Open Sourse Project*, Latest access time for the website is 20 March 2019.

[16] http://www.citi.umich.edu/u/provos/honeyd/. "Honeyd-Network", *Open Sourse Project*, Latest access time for the website is 20 March 2019.

[17] A. Chuvakin. "Honeynets: High Value Security Data": Analysis of real attacks launched at a honeypot", *Network Security*,Vol.2003, pp.11-15, 2003.

[18] J.K. Jones, G.W. Romney. "Honeynets: an educational resource for IT security", *Proceedings of the 5th conference on Information technology education, ACM*,pp.24-28, 2004.

[19] N. Weiler. "Honeypots for distributed denial-of-service attacks", *Proceedings. Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*,pp.109-114, 2002.

[20] A. B. Petruni, Robert "Honeytokens as active defense", *EC-Council Cyber Research*,Vol.11, No. 10,pp.1-14, 2011.

[21] http://old.honeynet.org/papers/cdrom/eeyore/. "Know Your Enemy:Honeywall CDROM Eeyore", *Open Sourse*, Latest access time for the website is 20 March 2019.

[22] S. Yeldi, S. Gupta, T. Ganacharya, S. Doshi, D. Bahirat, R. Ingle, A. Roychowdhary. "Enhancing network intrusion detection system with honeypot", *TENCON 2003. Conference on Convergent Technologies for Asia-Pacific Region, IEEE*, Vol. 4, pp.1521-1526, 2003.

[23] S.M. Khattab, C. Sangpachatanaruk, D. Moss, R. Melhem, T. Znati. "Roaming honeypots for mitigating service-level denial-of-service attacks", *24th International Conference on Distributed Computing Systems, Proceedings, IEEE*, pp.328-337, 2004.

[24] H. Artail, H. Safa, M. Sraj, I. Kuwatly, Z. Al-Masri. "A hybrid honeypot framework for improving intrusion detection systems in protecting organizational networks", *computers & security*,Vol.25, No.4,pp.274–288, 2006.

[25] C. Seifert, I. Welch, P. Komisarczuk. "Honeyc-the low-interaction client honeypot", *Proceedings of the 2007 NZCSRCS, Waikato University, Hamilton, New Zealand*,Vol.6, 2007.

[26] A. Sardana, R. Joshi. "An auto-responsive honeypot architecture for dynamic resource allocation and QoS adaptation in DDoS attacked networks", *Computer Communications*,Vol.32, No. 12,pp.1384-1399, 2009.

[27] C. Mulliner, S. Liebergeld, M. Lange. "Poster: Honeydroid-creating a smartphone honeypot", *IEEE Symposium on Security and Privacy*,Vol, pp.1-2, 2011.

[28] E. Vasilomanolakis, S. Karuppayah, M. Fischer, M. Fischer, M. Muhlhauser, M. Plasoianu, L.Pandikow, W. Pfeier. "This network is infected: Hostage-a low-interaction honeypot for mobile devices", *Proceedings of the Third ACM workshop on Security and privacy in smartphones & mobile devices*, pp.43–48, 2013.

[29] S. Liebergeld, M. Lange, C. Mulliner. "Nomadic honeypots: A novel concept for smartphone honeypots", *Proc. W'shop on Mobile Security Technologies (MoST'13), together with 34th IEEE Symp. on Security and Privacy*, Vol.4,pp.1-4, 2013.

[30] V. B. Oliveira, Z. Abdelouahab, D. Lopes, M.H. Santos, V.P. Fernandes. "Honeypotlabsac: a virtual honeypot framework for android ", *International Journal of Computer Networks & Communications*,Vol.5, pp.159-172, 2013.

[31] L. Krämer, J. Krupp, D. Makita, T. Nishizoe, T. Koide, K. Yoshioka, C. Rossow. "Amppot: Monitoring and defending against amplification ddos attacks", *International Symposium on Recent Advances in Intrusion Detection*,pp.615–636, 2015.

[32] H. A. Deshpande, "Honeymesh: Preventing distributed denial of service attacks using virtualized honeypots", *IJERT*,Vol.4, No. 8,pp.263-267, 2015.

[33] N. Agrawal, S. Tapaswi. "Wireless rogue access point detection using shadow honeynet", *Wireless Personal Communications*,Vol.83, No.1,pp.551–570, 2015.

[34] S. Litchfield, D. Formby, J. Rogers, S. Meliopoulos, and R. Beyah. "Rethinking the honeypot for cyber-physical systems", *IEEE Internet Computing*,Vol.20, No.5,pp.9–17, 2016.

[35] W. Han, Z. Zhao, A. Doupé, G.J. Ahn. "Honeymix: Toward sdn-based intelligent honeynet", *Proceedings of the 2016 ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*,pp.1–6, 2016.

[36] C. Saadi, H. Chaoui. “Cloud computing security using ids-am-clust, honeyd, honeywall and honeycomb”, *Procedia Computer Science*, Vol.85, pp. 433-442, 2016.

[37] P. Sokol, J. Míšek, M. Husák. “Honeypots and honeynets: issues of privacy”, *EURASIPJournal on Information Security*, Vol.4, pp.1-9, February 2017.