A Defence Mechanism: DNS based DDoS Attack

Arpita Narayan[1], UpendraKumar[2]

[1]Computer Science, BIT Mesra Deemed University, Ranchi, Jharkhand
[2]Faculty, Computer Science, BIT Mesra Deemed University, Ranchi, Jharkhand

Abstract-*Distributed Denial of Service (DDoS) attacks pose one of the most serious security threats to the Internet. In this work, we aimed to develop a collaborative defence framework against DNS based DDoS reflection and amplification attacks in networks. We focus on two main phases, which are victim detection and filtering of malicious traffic, to achieve a successful defence against DNS reflection attack and prevention against amplification attack. We propose an efficient server level approach to identify victim IP accurately and responsively by using unusual request count. Once the victim IP is confirmed, our approach is then to use HOP count i.e. number of router packets passes to reach destination, to filter out the entire illegitimate request.*

Keywords: *Distributed Denial of Service attacks (DDoS), Domain Name System (DNS), DNS message sequence, HOP count, Reflection attack, Amplification attack.*

1 Introduction

As DDoS attacks become one of the most threatening security issues, the need to detect this type of attack is increasing. Later DoS attack is launched for fun by some attackers, but later it become an way to show technical ability and converted into cyber war or for so called “hacktivist” groups [6]. Today Distributed Denial of Service (DDoS) attacks have become one of the biggest threats to the Internet's security and stability[4]. The threat toDDoS attacks keeps growing; the biggest DDoS ever of 300 Gbit/s in early 2013 has already been surpassed this year by an attack that was 25% larger in volume. The flows in the current Internet organizationthat facilitates DDoS attacks because the inability for a packet recipient to authenticate that packet’s claimed source IP address. In other words, an attacker can intentionally modify, or spoof, the source address of the packets it sends from a compromised host. Scenario for DDoS attacks that is based on IP address spoofing are:

- TCP SYN Flooding [9]:Normal TCP connection begins by sending TCP SYN packets from the server to the client sidePacket initiated TCP connectionIn this attack,[13][4] SYN packets contain spoofed source IP addresses, which cause the victim to waste resources that are allocated to half-open TCP connections which will never be completed by the attacker [6].
- Reflector Attack: In this attack described by Paxson [7],[4], the attacker try to make victim resource unavailable by overwalming with traffic. Intermediate servers are use to amplify the attacker’s bandwidth and or hide the attacker’s origin. DNS server based Reflection DDoS attacks, which use huge traffic to disable a victim server. An attacker who wants to evade sourceIP address based packet filtering will use source IP spoofing. However, as source IP address filtering mechanisms become widely deployed (e.g., the Pushback framework [7], [8]), it is likely that attackers will have to resort to source IP address spoofing to increase the effectiveness of their attacks.

Recent attack incidents verify the catastrophic outcomes of this class of attacks, when triggered against key Internet components like DNS servers. For example, as reported in [2], in October 2002 eight out of the thirteen root DNS servers were suffered a massive DoS attack. Many other similar attacks were triggered against DNS in 2003 and 2004 [3], [4]. In a recent study, the Distributed Denial of Service (DDoS)[32],[33] activity in the Internet was analyzed employing a method called “backscatter” [5]. The results of this study showed that nearly 4,000 DDoS attacks are released each week. In February 2006, name servers hosting Top Level Domain (TLD) zones were the frequent victims of enormous heavy traffic loads. In reflection based amplification attack DNS servers to amplify the attack traffic to a certain target victim. The open recursive servers are those that accept arbitrary DNS queries from any source and send the final response answers to the queries. Since the source IP address in the queries are spoofed as the victim’s and the elaborate DNS query could be much smaller than its associated response in size, the attack traffic is actually amplified while reflected to the victim server, which is called the DNS amplification attacks [28]. It is very difficult to deal with this kind of attack. For instance, in an ordinary DDoS attack, one can potentially block a bot instructed to launch a DDoS attack by blocking the bots IP address. Contrariwise, it is not so simple to block a DNS server without affecting and damaging the operation of a corporate network. The amplification factor in such recursive DNS attacks stems.

There is a group of attacks that overload targets with packets taking up massive amount of bandwidth and processing power in the hope of making the target server unavailable for genuine users and DNS server is unaware of any kind of attack

We propose the detection and filtering mechanism for reflection and amplification attack on DNS, a new detection and filtering scheme based on request count monitoring and HOP count assessment. In the first experiment, we quantify the effectiveness of the DNS attack detection and victim identification threshold filter. The false positiveand false negativerates, which represent the rate at which legitimate user's packets are consider as attack packets and the rate at which attacker packets are consider as Non-attack packets, respectively. Our scheme almost completely detects the reflection attack target at the good threshold selection. We propose algorithm for filtering mechanism on the fact that the HOP-count [31]information is indirectly reflected in the Time-to-Live (TTL) field of the IP header since each intermediate router decrements the TTL value by one. Although an attacker can forge any field in the IP header, he cannot falsify the number of hops an IP packet takes to reach its destination, which is solely determined by the Internet routing infrastructure. All packets of victim determined by detection stage forwarded for filtering and actual HOP count of victim is compared with all the victim packet HOP count. Then, legitimate packet will be filtered out.

2Related Works

Geva et al. [24] discuss DDoS attacks in a general sense. They performed simulations that illustrate the damage that attacks can cause and describe a spectrum of different DDoS attack mechanisms and various mitigation strategies against these attacks. They conclude that many defense mechanisms are problematic to deploy and that they may struggle to protect against the increasing threat level of today. A similar, but more practical and elaborate overview of DDoS attacks and defense mechanisms is given by Zargaret al. [25].

Independent of the application layer protocol, Gu and Strayer [29] propose botnet detection methods based on network flow characteristics. However, protocol-independent approaches will likely fail to detect DNS C&C as they expose neither chat-style characteristics nor necessarily spatial-temporal correlated behaviour. For example, Feeder bot does neither exhibit periodicity nor synchronized transactions among different bot executions – effectively exploiting the gap of existing detection approaches.

Casalicchio et al. [14] describe a highly detailed reference architecture for measuring the stability and security of the DNS. They argue that the DNS is the most important infrastructure underpinning the Internet and that there is a great need to assess the health of the DNS on a continuous basis.

The DDoS[34] activities owes much to the pioneer work carried out by Moore et al. in [34] that was revisited in [10]. The key observation behind the authors ‘technique is that attackers, before executing a DDoSattack[13], spoof their addresses using random IPs. Hence, once the attackers executed, all the victims’ replies are bounced back to the fake IP addresses [20], which could be in the monitored dark net space. Numerous research works has been performed on such data to analyzeDDoS activities. The majority focus on implementing new detection techniques to infer DDoS attacks [16], tracing back the sources of attacks [17], investigating spoofed attacks [18],[20],[31] and visualizing attacks [19]. In DNS Spoofing attack causes reflection attack.

The DDoS attacks can also be divided as bandwidth attacks and resources attacks in terms of the target of DDoS attacks. For the bandwidth attack, there are usually two types of DDoS attacks, namely, denial of edge service and denial of network service attacks [28],For the former type, the attackers usually try to saturate the ingress bandwidth of the victim side. The reflector attack belongs to the former type, which can render normal users not able to receive responses from the server on time. Methods for analyzing the characteristics of network traffic behaviour have been described in a large number of prior studies. Of particular relevance to our work are prior studies that describe techniques for classifying network traffic including [11][15],[16],[17],[18],[19]. These methods have been shown to be highly accurate, and we consider the information that they produce to be complementary to what is produced by our DNS query analysis. However, to obtain more specific information about DNS traffic it is necessary to store all important DNS packet fields such as source and destination addresses, queried domain name, or response data. This approach is used by [20],[21],[22],[23] for the detection of botnets based on the same DNS behaviour of devices, abnormal DNS traffic or malicious domain usage. This type of data can be also used for an intrusion detection system based on DNS traffic monitoring which was introduced in [18],[34][35].Our work uses the approach of DNS traffic flow analysis for differentiate request traffic from other DNS traffic flow for the purpose of detecting reflection attack and to identify target.

3 Problem Definition

In this paper we discuss the possibilities to detect reflection attacks on different threshold values and identify victim by inspecting DNS network traffic stored in flow-records. After that filtering of legitimate request is done on the basis of the fact that although an attacker can forge any field in the IP header, he cannot falsify the number of hops an IP packet takes to reach its destination. In particular, we will focus on the following research questions:

What are the flow-data fingerprints of DNS-based attacks?

How can the fingerprints of DNS-based attacks be recognized?

How the threshold should be chosen for maximum accuracy in attack detection?

How can victim of DNS-based reflection amplification attack be identified?

How the legitimate query can be filtered so that amplification attack may be prevented?

4Flow-based DNS Traffic Monitoring

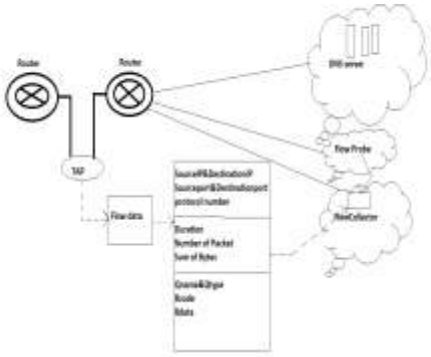

Figure1. Flow Monitoring Architecture of DNS Flow Record

There are two basic criteria for monitoring large and high-speed networks such as campuses or ISPs. First, monitoring tools must provide near real-time data analysis and, second, the tools must not demand large storage space. To fulfil these requirements, the concept of network flow is used. A flow is defined in RFC 7011 as “a set of IP packets [11] passing an observation point in the network during a certain time interval, such that all packets belonging to a particular flow have a set of common properties”. The standard flow record is a vector:
F= ($IPsrc$, $IPdst$, $Psrc$, $Pdst$, $Prot$, $Tstart$, $Tdur$, $Pckts$, $Octs$, $Flags$), where the flow is defined by the source and destination IP addresses $IPsrc$and$IPdst$, source and destination ports $Psrc$and $Pdst$, protocol $Prot$and the starttime $Tstart$with duration $Tdur$. The field’s $Pckts$and $Octs$represent the numberof transferred pack etc and octets, and $Flags$TCP flags.The flow exporter aggregates packets with common properties into one flowuntil the flow is terminated. This termination can be caused by the expiration offlow cache entry (active time-out, idle time-out or resource constraints), naturalexpiration based on packet flags indicating connection end, emergency expirationor cache flush [30]. In networks with a large volume of traffic, it is necessary tohave sufficiently large and free flow cache to avoid emergency expiration or cacheflush, which may cause unwanted flow records split.
Flow acquisition can be done by common network devices that support flow record export, such as routers, or by specialized network probes [30] which provides greater data accuracy and are able to effectively process a large volume of traffic. Figure 1depicts a monitored network with the probes installed at the local network uplink and also inside the network. The probe aggregates packets and export them as flow records to the flow collector that provides tools for basic flow processing and analysis. Although flow records do not contain information about application protocols, it is still possible to use them for monitoring DNS traffic. A DNS flow canbe distinguished from others by port-based protocol identification that relies on the fact that the TCP and UDP port number 53 is assigned to the DNS protocol by IANA. This port number is by default used by DNS resolvers which listen to this port. DNS monitoring using standard flow records can reveal anomalies that affect the volume characteristics of transferred data. However, anomalies connected to DNS application data remain undetected.
To understand how a reflection DDoS attack appears inNetFlow data, a short introduction will be given in theoperations of NetFlow. NetFlow is a network protocol that collects IP traffic information and aggregates this information as NetFlow records[37].
When a regular host sends DNS requests to a DNS server,
it will send each request from a random port number. This
results in flow-records consisting of one packet. Thus, for the
detection of a DNS reflection DDoS attack, the following
information is available [36]:

IPaddress of the host sending the DNS query.
IPaddressof the DNS server.
Time of the DNS request.
Time of the DNS reply.
Size of the DNS request

5. Detection and Filtering Algorithm

In this section we will discuss the model used for the identification of reflection attacks and prevention of amplification attack. After that, our algorithm for detecting targets of a reflection attack and filtering legitimate request to the targets is presented and then we will discuss windowing used for our algorithm. The last subsection will validate the thresholds of this algorithm in terms of accuracy, detection rate and FAR.

5.1 Model and Method

This section presents a new attack detection and filtering system for DDoS against DNS, which uses request count classifier to detect the attack and TTL field to filter the legitimate packet.

Reflection attacks will often make use of a script generating a high amount of requests with the same content, as it knows this specific request will result in a large response, amplifying the attack. Regular traffic request count size will vary with attack traffic, and thus this consistency separates an attack from legitimate requests.

Detectionphase, all packets are assumed to be analyzed by the server, using some detection algorithm. In other words, the server is given the power to differentiate between attack and legitimate packets that arrive in that window. Thus, the server is able to generate victim IP, suspicious IP, and normal traffic.

As Shown in Figure2 Our detection Architecture first select the DNS request traffic on the basis of destination port no 53.After that , matrices of source IP and its consequent request count was created as shown in dataset1 and dataset2.In the second phase of architecture, traffic separation is done on the basis of threshold value.

Case1shows that if the request count of particular IP is greater or equal to the threshold value then it is consider as reflection attack, and this IP is marked as victim in the detection phase. All the packets from victim IP is forwarded for filtering phase before sending the response message to that IP.

Case2shows that if the request packet of particular IP is greater than or equal to 100 then it will check the next condition that its count is greater than or equal to half of the threshold value or not. IP match the condition then marked as suspicious. Response is not forwarded till next window check. In the next window the particular IP again come under the above describe condition then marked as attack.

Case3: Response is forwarded to the entire request from the normal traffic.

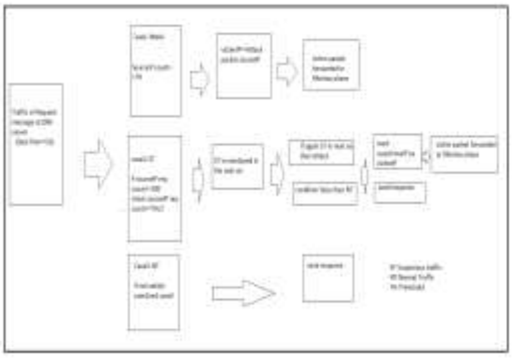

Figure2. Detection Model

In the second phase, the *filtering phase*, the server has to differentiate between victim IP actual request and spoofed request. For this actual HOP count of victim is calculatedHv and compared with the all the large number of request packet of the victim IP i.e. Hc. Hop Count Filter (HCF) is based on this TTL. Packets having same HOP cou are considered as legitimate and server sends response to only those packets, discards those IP packets with mismatching HOP counts.

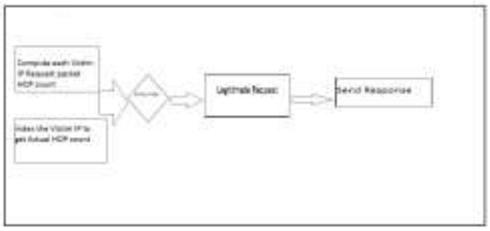
Compute each Victim IP Request packet HOP count
Index the Victim IP to get Actual HOP count
Legitimate Request
Send Response

Figure3. Filtering Model

5.2 Algorithm

While our detection algorithm desirable works in real time, a certain amount of data is required to calculate designed to work with ten thousand data, as a trade off between the amounts of data required the metrics of data such as the consistency of the request count. Because of this, our detection algorithm is and the preference to work in real-time.

Our detection approach is based on the metrics discussed in the previous section and is split up into two parts. The first part focuses on detecting victim IP addresses on the basis of large count DNS requests through the use of a threshold for the amount of requests generated by an IP address. The second part focuses on detecting IP addresses receiving suspicious DNS request, also through the use of a threshold for the amount of request received. In two consecutive sliding windows the same IP is marked as suspicious then it will be consider as victim IP. Thresholds used in this detection approach are discussed in the next section.

Condition1 Detecting IPs generating unusual requests

```
1.For separating request messages:
if(destination port == 0035)
reportflowrequest messages(Accept)
ws=getAggregatedflowRequestsFromSourceIPAddr()

2. For collecting individual srcIPAddr count in sliding window(ws) :
for(i=0;i<=ws;i++)
    Report flowrequest _count.SrcIPAddr

3. For identification of victimIP:

ifcount.SrcIPAddr>= Th.count
report Attack detected
markSrcIPAddr (Victim IP)
4. For identification of suspiciousIP:

elseifcount.Pkts>= 100
ifrequests_Count.SrcIPAddr> (Th.count/2) :
report to block the requests from SrcIPAddr till next ws
markSrcIPAddr(Suspicious)
```

condition 2 Filtering the Victim IP legitimate request:

```
For each packet of victimIP:
Extract the final TTL Tf
Infer the initial TTL Ti
Compute the Hop Count Hc= Ti – Tf
Index Victim IP to get stored hop count Hv
If(Hc==Hv)
the packet is legitimate;
send response message
else
the packet is spoofed;
discard the packet and send no reply;
```

The pseudo-code for the first part of the detection algorithm is listed in Program 1. On rule 1 this part starts by filtering the dataset on requests (i.e. collect-records with destination port 53) and then combines all requests count with the same source IP address. This result in an aggregated flow-record for each IP addresses containing all requests originating from that address. The next step is to check how many packets each of these aggregated flow-records contain, and thus how many requests were sent. If more than or equal to the threshold requests were sent, then it can be deduced that this IP address sends more requests than a legitimate DNS and marked as victim.

If the amount of requests sent by an IP address is less than threshold but more than 100, the aggregated flow-records of requests count from this IP address are retrieved to calculate whether it is more than or equal to the half of the threshold request send, then it can be deduced that block the requests till next sliding window check and mark as suspicious. Again in next check if same IP is marked as suspicious then this IP is considering as victim.

In Program 2 the filtering algorithm can be seen. Only the victim IP packets go for the filtering phase. This part focuses on identifying legitimate request packet from the spoofed packets. Since HOP-count[31] information is not directly stored in the IP header, one has to compute it based on the final TTL value. TTL is an 8-bit field in the IP header, originally introduced to specify the maximum lifetime of each packet in the Internet. Each intermediate router decrements the TTL value of an in-transit IP packet by one before forwarding it to the next-hop. The final TTL value when a packet reaches its destination is, therefore, the initial TTL decreased by the number of intermediate hops (or simply hop-count). The challenge in hop-count computation is that a destination only sees the final TTL value.

By sending a packet to the claimed host that will cause a reply we can check to see if the TTL[31] in the reply is the same as the packet being checked. If they are of the same protocol, they generally have the same TTL. Because different protocols use different initial TTLs, when the probe packet is of a different protocol, we must infer the actual hop count. Only a few initial TTL values are commonly used. For TCP/UDP, 64 and 128 are most commonly seen. ICMP commonly uses 128 and 255 as the initial value. By subtracting the observed TTL from the supposed initial value we can estimate the number of hops. For example, for an ICMP packet with an observed TTL of 241, we get 255-241 or 14 as the estimated number of hops. If we are checking a TCP packet with an observed TTL of 50, we get 128-50=78 and 64-50=14. Because 14 is the expected value, we can assume the packet was not spoofed. If we knew the actual initial TTL for the host this would be more certain.DNS server wants to assess the authenticity of victim packets then it initiates the verification modules. For indexing victim IP hop-count, initially source wants to communicate with the destinations node then it checks its routing table. Here source is DNS server and destination is victim node .If the entry is found then TTL field is updated in initial message. If the entry is not found then it as sends the Multicast Probe RREQ message to destination. Destinations reply with its IP Address, mapping and required details in Probe RREP message. This entry of multicast route is getting updated in routing table. Total number of hops is the number of devices traversed during this data communications.

Now the hop-count of each request packet of victim is compared with victim actual hop-count. If the match found then it is legitimate packet send by victim and DNS server will send response message to only these packets. In this way amplification attack is prevented by the DNS server.

5.3Windowing

[5] Windows based on packet count, occurring for every n network packets .The important fact that drives the choice among the two is the fact that analysis during learning should help analysis to be done real time. Packet windows are considered a better choice of the two for the following reasons:

Packet windows provide smaller reaction times during an attack situation, because of the fact that the system may haveto wait for the time window to complete before deciding to flag an attack(or anomaly)Packet windows may provide for more accurate modelling, since the number of possible events to be considered is bound to be limited, whereas the number of events to be considered may be large in time windows, since there is no control on how many packets could be received within a time window.In case of packet window the periodicity of computations may not be known, since one doesn’t know when n packets would be available for computing. This occur particularly when there is high traffic rate at one time, and low traffic rate at another. But the server can easily handle low volume traffic and related computations in any case. In our experiment window size is chosen as ten thousand.

5.4Data Set

Data collection is usually a basic requirement for any IDS .The type of data that we should collect is based on the type of IDS. When accessing to a real environment for traffic simulation is hard. According to our knowledge, there are no available generated dataset for DDoS attacks against DNS. Therefore, the required data for our experiments was generated randomly using a packet study of cisco.

The network packet detail of DNS request is required for our work. There are two types of traffic generated in the network which are legitimate traffic and attack traffic. In our work we collect traffic request packet count of both legitimate client and attack client in consecutive window. The attacker is expected to flood the target name server with excess traffic. We chose source IP 192.168.2.3 and 198.16.7.3 as target for reflection attack in dataset1 and dataset2 respectively. Thus, for the detection of a DNS reflection DDoS attack, the following information is used in this paper: SourceIP, Source port name, Source IP packet count, Destination port name, Window size(ws),Threshold(Th), HOP count

Table1. Dataset 1 of DNS Request table

SourceIP	S_PortNo.	DestinationIP	D_portNo	Packet count
192.168.2.6	092A	192.168.60.163	0035	500
192.168.2.2	0951	192.168.60.163	0035	1500
192.168.2.3	**0352**	**192.163.60.100**	**0035**	**4300**
192.168.2.4	072A	192.163.60.15	0035	2500
192.168.2.5	0735	192.163.60.163	0035	1200
192.168.2.6	0734	192.163.60.212	0045	560

Table2. Dataset 2 of DNS Request table

SourceIP	S_PortNo.	DestinationIP	D_portNo	Packet count
192.168.10.1	301A	123.60.10.1	0035	100
192.168.10.2	301B	123.60.10.3	0035	200
128.15.10.6	301C	123.60.10.1	0035	2000
198.16.7.3	**301D**	**123.60.10.1**	**0035**	**3200**
128.15.12.7	301E	123.60.10.1	0035	100
192.168.2.4	301F	123.60.10.1	0035	2700
128.13.5.8	301G	123.60.10.1	0035	500

5.5 Thresholds

In this section, the deduction of thresholds used in our detection approach is discussed. The validation of victim identification and attack detection calculation all depend on the possibility to calculate a good threshold. Thus we have tochoose thresholds that give a high accuracy in attack detection. For this we vary the threshold and find the accuracy of attack detection at different level. The thresholds that need to be calculated give a high accuracy in attack detection

In the first experiment, we quantify the effectiveness of the DNS attack detection and victim identification threshold filter. The *false positive and false negative* rates, which represent the rate at which legitimate user's packets are consider as attack packets and the rate at which attacker packets are consider as Non-attack packets, respectively. For the purpose of our evaluation, we refer to the following metrics: the *accuracy, detection rate and false alarm rate(FAS);* which is discussed in detail. Threshold (Th) level with respect to the window size (ws).In the next section threshold validation has been done for maximum accuracy and less FAR.

5.6 Results and Discussions

In this section, the performance of each classifier in terms compared. For better understanding of results comparison, we introduce these criteria:

Accuracy, which refers to the proportion of data of detection rate, false alarm rate, and accuracy, was classified as accurate type in the total data. Accurate situations are True Positive (TP) and True Negative (TN), while false detected situations are False Positive (FP) and False Negative (FN).[] Accuracy of the system is calculated by the following equation:

$$\text{Accuracy} = \frac{TP}{TP+TN+FP+FN} *100\%$$

Detection rate, which refers to the proportion of each type of DoS attack detected among all the same type of attack and is calculated by the following equation:

$$\text{Detection Rate} = \frac{TP}{TP+FN} *100\%$$

- False Alarm Rate (FAR), which is defined as the percentage of the network traffic that is misclassified by the classifier. It can be calculated using the following equation:

$$\text{FAR} = \frac{FP}{FP+TN} *100\%$$

Threshold →	5%	10%	15%	20%	25%	30%	35%	40%	45%
Dataset1	43	48	60	75	75	100	100	100	57
Dataset2	36	41	53	53	100	68	68	68	68

Table3. Percentage Accuracy in detection of attack at different threshold Level w.r.t. ws

Threshold →	5%	10%	15%	20%	25%	30%	35%	40%	45%
Dataset1	100	100	100	100	100	100	100	100	0
Dataset2	100	100	100	100	100	100	0	0	0

Table4. Percentage Detection Rate at different threshold level w.r.t.ws

Threshold →	5%	10%	15%	20%	25%	30%	35%	40%	45%
Data set1	100	91.22	70.17	43.38	0	0	0	0	0
Data set2	92.75	86.75	69.11	39.70	0	0	0	0	0

Table5. FAR at different threshold level w.r.t ws

	Dataset1	**Dataset2**
Victim IP	192.168.60.163	198.16.7.3
Suspicious IP	192.60.163 **192.168.2.4**	198.15.10.6 **192.168.2.4**

Table6. In case of 30% threshold VictimIP and SuspiciousIP

Table 3 presents the accuracy at 30% threshold is 100% in both dataset .Table 4 presents detection rate 100% till 40% threshold in dataset1 and till 30% threshold in dataset 2 after that True Negative rate increases very harshly and detection rate become 0.Table 5 presents FAR it is high at low threshold choice and decreases to 0 at good quality threshold. Table 6 presents the Victim IP and suspicious IP at 30% threshold when accuracy is 100% .Suspicious IP 192.168.2.4 will consider as attack on the base of proposed model.

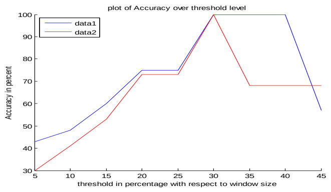

Figure4. Accuracy Vs threshold Graph using Matlab Tool

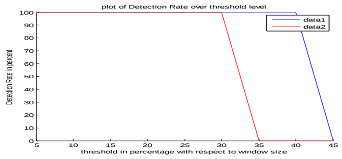

Figure5. Detection Vs Threshold Graph using Matlab Tool

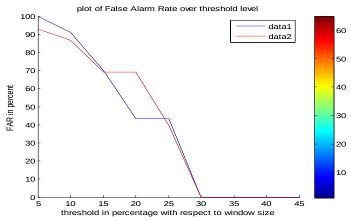

Figure6. FPR Vs Threshold Graph using Matlab Tool

6 Conclusion and Future Work

In this paper, we presented a detection approach for DNS reflection attacks that is capable of detecting the target using only the requests flow-records of the attack. The approach is based on thresholds that separate legitimate DNS traffic from reflection attacks. After identifying target IP from the detection approach All the request packet is forwarded for filtering phase.

we have presented a hop-count-based filtering scheme that detects .Thisalso propose to discards spoofed IP packets to conserve system resources. Our scheme inspects the hop-count of incoming packets to validate their legitimate packets by doing some comparisons. Using only a moderate amount of storage, HCF constructs an accurate IP to HC mapping table via IP address aggregation and hop-count clustering. Once spoofed DDoS traffic is detected, target IP packets switches to the *filtering* state and discards most of the spoofed packets.

The threshold values were varied at different level for finding high accuracy, detection rate and less false alarm rate. Validation of the presented detection algorithm has shown that the approach can accurately separate targets of reflection attacks from other traffic. It is found to have 100% accuracy when we have chosen appropriate thresholds in the range of 30%. However, it was found that sometime targets exist that cannot be detected with this approach, as we have taken threshold very high, so that the detection rate decreases very fast. This validation was done using assumed dataset.

There are several issues that warrant further research. First, We are in the process of analyzing the effectiveness of the HOP count filtering methods discussed above as well as determining estimations of the false-positive and false negative rates. Second, to install the detection and filtering system at a server site for practical use, we need a systematic procedure for setting the parameters of detection and filtering schema, such as the sliding window size. We would like to build and deploy in various high-profile DNS server sites to see how effective it is against real DNS based DDoS traffic which causes reflection and amplification attack.

References

[1] J. Mirkovic and P. Reiher, "Ataxonomy of DDoS attack and DDoSdefense mechanisms," ACM SIGCOMM Computer Communication Review, vol. 34, pp. 39-53, 2004.

[2]Gibson, S., “DRDoS Distributed Reflection Denial of Service”,http://grc.com/dos/drdos.htm, 2002.

[3]Guo, F., Chen, J., and Chiueh, T., “Spoof Detection for Preventing DoS Attacks against DNS Servers”, In Proceedings of the 26th IEEE international Conference on Distributed Computing Systems , July 2006

[4]Chandramouli, R. and Rose, S. “An Integrity Verification Scheme for DNS Zone file based on Security Impact Analysis”, In Proceedings of the 21st Annual Computer Security

[5] Atkins, D., Austein, R., “Threat Analysis of the Domain Name System (DNS)”, RFC 3833, Aug. 2004

[6] K. Lakshminarayanan, D. Adkins, A. Perrig, and I. Stoica.Taming IPPacket Flooding Attacks.In *Proceedings of ACM HotNets-II*, pages45–50, November 2003.

[7] Vern Paxson. An Analysis of Using Reflectors for Distributed Denial-of- Service Attacks.*Computer Communication Review*, 31(3):38–47, 2001.

[8] John Ioannidis and Steven M. Bellovin. Implementing Pushback: Router-Based DefenseAgainstDDoS Attacks. In*Proceedings of the Symposiumon Network and Distributed Systems Security (NDSS 2002)*, San Diego, CA, February 2002

[9] W. Chen and D.-Y.Yeung, “Defending Against SYN Flooding Attacks Under Different Types of IP Spoofing”, ICN/ICONS/MCL ’06, IEEE Computer Society, pp. 38-44, April 2006

[10] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, andS. Savage, “Inferring internet denial-of-service activity,”ACM Transactions on Computer Systems (TOCS), 2006.

[11] Z. M. Fadlullah, T. Taleb, A. V. Vasilakos, M. Guizani, and N. Kato, “Dtrab: combating against attacks on encrypted protocols through traffic-feature analysis,” IEEE/ACM Transactions on Networking (TON), vol. 18, no. 4, pp. 1234–1247, 2010

[12] G. Yao, J. Bi, and Z. Zhou, “Passive iptraceback: capturingthe origin of anonymous traffic through networktelescopes,” SIGCOMM Comput. Commun. Rev., vol. 41, no. 4, Aug. 2010.

[13] J. Bi, P. Hu, and P. Li, “Study on classification andcharacteristics of source address spoofing attacks in theinternet,” in Proceedings of the 2010 Ninth International Conference on Networks. Washington, DC, USA: IEEE Computer Society, 2010

[14] E. Casalicchio, M Caselli, and AColetta. Measuring the global domain name system. IEEE Network,27(1):25{31, January 2013

[15] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy. Transport Layer Identification of P2P Traffic. In Proceedings of ACM Internet Measurement Conference (IMC ’04), Taormina, Italy, October 2004.

[16] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: Multilevel Traffic Classication in the Dark. In Proceedings of ACM SIGCOMM ’05, Philadelphia,
PA, August 2005.

[17] “UltrDNSDDoS Attack, Washington Post,” May 2005, http://blog.washingtonpost.com/securityfix/2006/05/ blue security surrenders but s.html.

[18] J. Erman, M. Arlitt, and A. Mahanti.Traffic Classification Using Clustering Algorithms. In Proceedings of ACM SIGCOMM Workshop on Mining Network Data (MineNet ’06), Pisa, Italy, September 2006.

[19] A. McGregor, M. Hall, P. Lorier, and J. Brunskill.Flow Clustering Using Machine Learning Techniques.

[20] Choi, H., Lee, H.: Identifying botnets by capturing group activities in dns traffic. Comput.Netw. 56(1), 20–33 (Jan 2012)

[21] Marchal, S., Francois, J.,Wagner, C., State, R., Dulaunoy, A., Engel, T., Festor, O.: DNSSM: A Large Scale Passive DNS Security Monitoring Framework. In: Network Operations and Management Symposium (NOMS), 2012 IEEE. pp. 988–993 (Apr 2012)

[22] Ellens, W., Żuraniewski, P., Sperotto, A., Schotanus, H., Mandjes, M., Meeuwissen, E.: FlowBbased Detection of DNS Tunnels. In: Emerging Management Mechanisms for the Future Internet, pp. 124–135. Springer (2013)

[23]E. Wustrow, M. Karir, M. Bailey, F. Jahanian, and G. Huston, “Internet background radiation revisited,” in Proceedings of the 10th annual conference on Internet measurement. ACM, 2010, pp. 62–74

[24] M. Geva, A. Herzberg, and Y. Gev. Bandwidth Distributed Denial of Service: Attacks and Defenses.
IEEE Security & Privacy, 2013.

[25] S.T. Zargar, J. Joshi, and D. Tipper. A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks. IEEE Communications Surveys & Tutorials,

[26] T. Peng, C. Leckie and K. Rammamohanarao, “Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems”, ACM Computing Surveys, Vol. 39, Issue 1. 2007.

[27] L. Garber. Denial-of-service attacks rip the Internet.Computer, 33(4):12{17, April 2000.[10] R. Vaughn and G. Evron, “DNS amplification attacks,” 2006. [Online].Available:
http://www.isotf.org/news/DNS-Amplification-Attacks.pdf

[28] Xin Liu, Xiaowei Yang, and Yanbin Lu. To filter or to authorize: Network-layer DoSdefense against multimillion-node botnets. In *Proceedings of ACM SIGCOMM*, 2008.

[29] W. T. Strayer, D. E. Lapsley, R. Walsh, and C. Livadas, “Botnet detection based on network behavior,” in Botnet Detection, ser. Advances in Information Security, W. Lee, C. Wang, and D. Dagon, Eds. Springer, 2008

[30] Karasaridis, A., Meier-Hellstern, K., Hoeflin, D.: Detection of DNS Anomaliesusing Flow Data Analysis. In: Global Telecommunications Conference, 2006.GLOBECOM’06. IEEE. pp. 1–6. IEEE (2006)

[31] Haining ,Cheng Jin, and Kang G. Shin” Defense Against Spoofed IP Traffic
Using Hop-Count Filtering” IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 15, NO. 1, FEBRUARY 2007

[32] L. Garber. Denial-of-service attacks rip the Internet. In IEEE Computer, volume 33, April 2000.

[33] D. Moore, G. Voelker, and S. Savage, “Inferring internetdenial-of-service activity,” in In Proceedings of the 10thUsenix Security Symposium, 2001

[34] B. Irwin and N. Pilkington, “High level internet scaletraffic visualization using hilbert curve mapping,” inVizSEC 2007.Springer, 2008.

[35] Perdisci, R., Corona, I., Giacinto, G.: Early Detection of Malicious Flux Networks via Large-Scale Passive DNS Traffic Aanalysis. Dependable and
Secure Computing, IEEE Transactions on 9(5), 714–726 (2012)

[36] V. K. cek. Inspecting DNS Flow Tra_c for Purposes
of Botnet Detection. GEANT3 JRA2 T4 Interal
Deliverable, 2011

[37] B. Claise. Cisco Systems NetFlow Services Export
Version 9. RFC 3954, IETF, October 2004.