Distributed Defense Scheme for Managing DNS Reflection Attack in Network Communication Systems

Dana Hasan Ahmed, Masnida Hussin, Azizol Abdullah, Raja Azlina Raja Mahmood
Department of Communication Technology & Networks,
Faculty of Computer Science & IT, University Putra Malaysia (UPM), Selangor, Malaysia.
danajaf@gmail.com

***Abstract*—Domain Name System (DNS) is based on client-server architecture and employed User Packet Diagram (UDP) protocol to transport requests and responses. Due to UDP supports unreliable connection, malicious users are able to fabricate spoofed DNS requests very easily. Such DNS problems in turn affect numerous other network services and critical in resource utilization. Delay in deploying secure DNS motivates the need for local networks to protect DNS infrastructure. DNS reflection attack for example takes advantage of the DNS response message and results substantially larger than DNS query messages. In this work, we propose a distributed defense scheme in DNS infrastructure to prevent from reflection attack. Our defense scheme aims to prevent spoofed addresses from getting any responses by applying a classification-based packet filtering strategy. Specifically, our local DNS server regularly checked DNS requests in its database in order to differentiate between legitimate and illegitimate requests. We invent validation phase in our filtering strategy by getting confirmation before the request stored in local side server. The key idea behind this is to ensure the local DNS database is merely stored legitimate requests and prevent the fake DNS request transferred to users. Our analysis and the corresponding experimental results show that the proposed scheme offers an effective defense solution while implicitly improving network communication traffic.**

***Keywords*—DNS Reflection Attack; Defense Scheme; Communication Traffic**

I. INTRODUCTION

From its creation over 30 years ago, Domain Name Service (DNS) has been successfully maturing to become the essential part of the Internet today. It is an important part of Internet infrastructure that translates domain names into Internet Protocol (IP) addresses and vice versa [1]. The inaccessibility of the naming service causes big impairment in the Internet and might leads to catastrophic results [2-6]. The significance of DNS in Internet services made it very (if not the most) fascinating target for malware criminals and hackers. It is because when the DNS server fails, the domain that it serves also went inaccessible that makes the probability of wide-scale disturbance.

Nowadays, the security community put the DNS protocol under their consideration because of the security shortcomings, cracks, and defects found in it in the last few years. On the Internet, the reflection-based attacks is the most familiar and nastiest one, which very expensive and high performance hardware and equipment required to counter on it [3]. In such attack scenario, the attacker forges a datagram with victim's IP address as the source address. The datagram then is sent to DNS server, amplifies it and sends it back to the victim, which brings about uninvited traffic. Due to UDP datagrams are connectionless scheme, it is very easy the attacker to destruct the IP address. Recently, attackers launched outbreaks with hundreds of Gb/s bandwidth of fake addresses and make the network traffic in heavy congestion [7]. When the attacker conquered the local DNS server it then become a reflection server; hence, users/victims cannot allocate their self into the absolute server. The main role of defense systems is to accurately detect such attacks and quickly respond to stop the illegal incoming requests. It is significant to recognize the legitimate traffic and differentiate it from attackers in order to avoid them for sharing the same communication traffic. Unfortunately, it is hard to trace such differentiation at a traffic attacker's structure. The factor clearly indicate that the DDoS problem required a distributed cooperative solution [4, 5, 8, 9]. Specifically, the traffic detection mechanism is needed at near to the users/victims. Meanwhile, the process of responsive and division of communication traffic between legitimate users and attackers is applied at the source communication channel. In addition, the communication traffic from either legitimate users or attackers can be achieved by enlisting the help of backbone routers for controlling the attack traffic.

In this paper, we propose classification-based mechanism for detecting spoofed addresses and preventing from getting any responses from the local server. Such processes are applied in both authoritative and recursive DNS servers. Our defense mechanism able to accurately distinguish between legitimate DNS packets and the fake ones. It also implicitly helps to increase system reliability in communication network. This paper organizes as follows. Section 2 introduced an overview of Domain Name Service from security perspective. In Section 3, we discussed about DNS reflection attack. Section 4 described our defense mechanism. Experiment and results are presented in Section 5 and, finally Section 6 concluded the paper.

II. Overview of Domain Name System (DNS) from Security Perspective

The name of communication servers can be divided into two main categories are authoritative and recursive (resolver) servers. The authorization server provides response (answer) to request queries that based on communication protocol in the system configuration. Meanwhile, the recursive (resolver) server forwarded a query that does not have any match from its record to the higher level of servers in the DNS hierarchy system. It aims to find the answer for the query. In some cases, if the answer is not found means that there is no matching in DNS record, and then it will be directed to the DNS root server that represent as ".", the top of the DNS hierarch.

DNS is a client-server service based system. It employs a constitution of a distributed database that takes advantage of a hierarchical tree structure to organize the domain name space into zones [10]. For each zone, the authoritative name server (ANS) responses for each incoming request that gets help from DNS Resource Record (RR). Each RR outlines the zone resources to its analogous domain name. RR is dispatched the queries to a predefined recursive name server (RNS) when there is an application on a given host needs the Internet Protocol (IP) address of a domain. The RNS then traverses the DNS hierarchy and try to find and match the appropriate answer. Moreover, for memory performance reasons, the RNS maintains a cache memory of RR for storing the current received query. It also aims to minimize searching time of subsequent similar requests that arrived from other users.

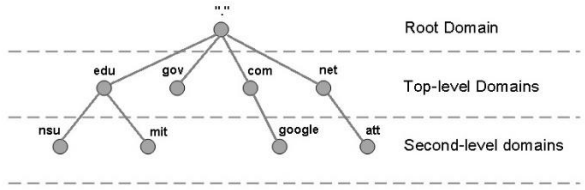

Figure 1: Domain Name Structure

An open DNS resolver server is another component in DNS system. It role is to resolve the recursive queries for both local and non-local users [11]. The open resolver servers are a necessary element in DNS amplification attacks. Usually, DNS servers should reject queries that arrived from anonymous networks. It only answers to those queries that originate from the trusted networks. In some circumstances, it may force some hosts or companies to make their DNS server become as open DNS resolver. It aims to serve their employees and clients that are traveling around the world and need trusted DNS servers. However, the DNS open resolvers answer all incoming DNS queries. There are many public DNS open resolvers such as OPENDNS (208.67.222.222 or 208.67.220.220) and Google's DNS (8.8.8.8). According to Open Resolver Project in Cisco Systems [12], since October 2013 there exist about 28 million operating open DNS resolvers in the world have facing with risky and great threats. Therefore, in order to avoid mishandlings, most of the known public DNS servers have equipped with some security features and very strong DNS policy.

Botnets are prevailing mechanisms for the facilitation of the distributed denial of service (DDoS) attacks on computer networks or applications [8, 13]. Botnets consist of networked collections of compromised machines called robots or 'bots' for short that is controlled by nodes called 'Botmasters' or 'Botherders'.

In DNS reflection/amplification attacks, botnets query DNS open resolvers with spoofed requests and return much larger DNS responses to the victim (the spoofed IP address) [13]. Botnets are networks that promoted the distributed denial of service (DDoS) attacks on computer networks or applications. The high amplification factor will be achieved by the attacker when DNS server answers "ANY" requests. It is because "ANY" request returns all records of the entire name server that recognized in the RR. The network firewall can be used to block all "ANY" requests, unfortunately it also will probably be blocked the legitimate traffic as well. There are other some services that normally used by the attackers such as "RRSIG", "DNSKEY" and "TXT"(*Abbasi, 2014*) to cause large amplification. There are few filtering mechanisms in DNS that able to provide a complete separation and identification of legitimate and fraud requests. This leads to bogus traffic that flooded in the communication path between the attack and users *(Douglas C. MacFarland, Craig A. Shue, 2015*).There is also lack of defense mechanism that able to help the users/victims during communication attack [6]. Therefore, DNS queries (i.e., request and response) needed reliable mechanism to prevent their queries from misleading by illegal request or attackers.

III. DNS-Reflection Attack

DNS reflection is a method used to perform Distributed Denial of Service or commonly called as DDoS attack. This method used DNS server as its base to slow down the communication network infrastructure and resources [8]. In such attack, the attacker sent DNS request to the DNS resolver server that consist the spoofed address as the target's address. The attackers then submit as many requests as possible to maximize the amplification effect. There are three main attributes in order to identify DNS amplification attack [3, 6]. First, the DNS amplification attack used port 53 and UDP protocol. Second, there is large volume of UDP packets that passed through the communication channel within short time period. The last characteristic of DNS amplification attack is that incoming and outgoing ports are do not match to each other's.

In order to launch the DNS reflection/amplification attack, the attacker must accomplish two procedures [6]. Firstly, the attacker must spoof the victim's IP address. Through this tactic, the attacker achieved the traffic reflection that cause all the responses from the DNS server then be directed to the victim's server. Second procedure is the attacker creates the responses that are several times larger than the request. The authors in [7] proposed the defense mechanism to reduce the amplification factor by lowering the amplification factor. This is achieved by increasing requests' sizes and disabling response to some records, (i.e., "ANY" record). The advantage of this mechanism is it can save the communication channel's bandwidth when the attack occurs. The drawback of such mechanism is it will be enlarged the communication traffic on the network at most of

the times. The authors in [9] proposed the defense method by storing information of every outgoing DNS packets in DNS Resource Record (RR) in order to distinguish between legitimate and fake packets to counter DNS reflection. Their defense method checked every incoming response in the DNS RR and only be accepted if the response existed in the record. Otherwise, it considered as suspicious and the response then be discarded. However, it is does not concerned on protecting the traffic from flooding by the attackers' responses. The response rate limiting (RRL) is defense method that proposed in [14] to mitigate DNS reflection attacks. RRL reduced the reflection attack rate by warning the authoritative servers of the high volumes of malicious queries. However, it is only applicable for authoritative name servers where it might decrease the server performance when the attacks get more sophisticated and dynamic.

When the flooding attack is detected, the computing system will be disconnected the victim from the network and manually fix the problem. All of the flooding attacks waste many resources (e.g., processing time, space, etc.). Hence, the critical goal of defense mechanism is to dynamically detect the attacks as soon as possible and stop/block them as near as possible to their sources. The first criterion for classification is the location where the defense mechanism implemented. There are two main types of defense mechanisms are centralized and hybrid [8]. In the centralized defense mechanism, the detection and response is mostly control centrally either by each of network deployment points (e.g., source-based mechanisms) or by some responsible points within the group of deployment points (e.g., network-based mechanisms). As opposed to centralized defense mechanisms, hybrid defense mechanisms are deployed at (or their components are distributed over) multiple locations such as source, destination or intermediate networks and there is usually some cooperation agreement among the network deployment points [4]. The advantage of such defense mechanism is more robust against flood attacks. Due to there are more resources at various levels (e.g., destination, source, and network), the authorization and authentication easy to monitor and implement. However, it might be increased the processing complexity and overhead because of the cooperation and communication among distributed components scattered all over the Internet. The hybrid defense mechanism also needs trusted communication among various distributed components in order to cooperate/collaborate in dynamic networks. Our defense mechanism in this work is explicitly taking into account distributed computing environment while classifying the DNS requests of legitimate and fake requests.

IV. Defense Mechanisms Towards DNS Reflection Attack

In this work, the defense mechanism proposed by applying a classification-based packet filtering strategy. It aims to prevent DNS reflection/amplification attack to be flooded into the communication network. In this work, we used two type of additional packets are validation and confirmation packets. We distinguished DNS requests between legitimate and fake requests and drop the fake request queries before it can congest/harm the communication path towards the users/victims. Specifically, every new request from the recognized domain is be given validation packet before its detail is stored in the DNS RR at LRS. After the packet's detail kept at the LRS, the ANS forwarded the confirmation packet to acknowledge the sender authenticates by the domain server already. Our validation and confirmation packets are created per request basis. That means the request must be has same information for every incoming and outgoing processes. It able to eliminate the fake DNS request stores in the system because the DNS amplification attack always change its information.

Our classified-based defense mechanism consists of Local Recursive Server (LRS), Authoritative Name Servers (ANS) and several processing machines that used by network end users (i.e., user and attacker) as in Figure 1.

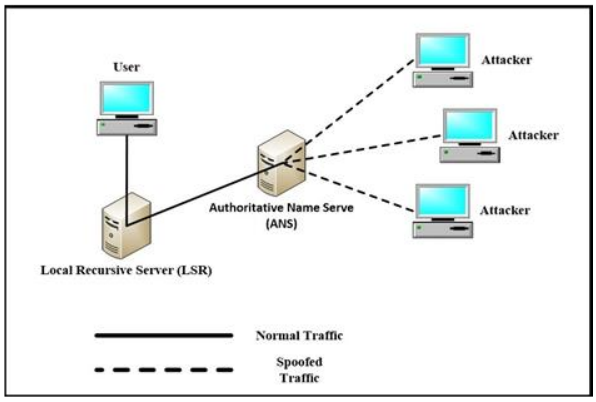

Figure 2: System Model

Every component in the system is running as virtual machines. We used VMware Workstation 12.0 X64 for designing the computing system. For both authoritative name server (ANS) and local recursive server (LRS) we utilized Debian 8.0 32-bit. Meanwhile, the user machine works on Microsoft Windows 7 32-bit, and Kali Linux 1.1.c acts as the attacker that starts the bot to attack the system. Attackers send spoofed, fake requests to Authoritative Name servers (ANS) using DNS flooder 1.1. Information about DNS request, validation request, validation response, and DNS response query are stored in My SQL tables within both LRS and ANS. Moreover, all the communication traffic requests/responses are stored using packet capturing and analyzing tool. In this work, we used IPtraf as the capturing tool.

The attacks formulated and commenced when ANS receives a large number of requests from attackers. The entire requests' sources IP addresses spoofed, where ANS then replicates them to the user/victim through the LRS. The communication path to the LRS is flooded with bogus traffic after the attack reflected by the ANS. At the endpoint, the Denial of Service received by the user/victim due to unknown DNS response that can be matched in the record at LRS.

The attackers' data is created by using DNS Flooder 1.1. The program is written in C language that developed by PLXSert [15]. PLXSert has observed the release and deployment of new DNS reflection tool. It contains new popular method of constructing large DNS resource records; i.e., contains until more than 4,000 bytes of responses, especially when the DNS queries with an "ANY" requests from the spoofed IP addresses. The responses to these "ANY" requests resulted in amplified

attack payload which may reach until 50 times larger of the size of requesting query [15].

Our classification-based defense mechanism will be verified the request packets within two approval stages. Specifically, when the user machine requests a particular website that its address is not in the cache records, the Local Recursive Server (LRS) updates the record with the new request packet (i.e., the information are Source IP, Destination IP, Source port, Destination port). It then forwards the information to the Authoritative Name Server (ANS). The ANS sends a validation request to the source of the request packet. If the packet is from the legitimate source, then the LRS checks for validation packet of the corresponding request. It then sends back the confirmation packet and informs the ANS about the request packet (i.e., legitimate request). However, if there is no validation packet, LRS sent a false response to ANS. When ANS received the false response from LRS, it drops the request and all other requests that acts like that one.

Due to the size of the response is considerably larger than the request; the attacker increased the amount of traffic that connected directly to the victim. In this work, we used two key issues for measuring severity of the attack. It based on amplification factor and attack ability. The amplification factor is the ratio between the traffic volume of response and request packets that implicitly represent the resource cost ratio between attackers and victims. The amplification factor computed as follows:

$$Amplification\ Factor = \frac{Response\ size}{Request\ size} \quad (1)$$

Meanwhile, the attack ability is the absolute amount of traffic that launched by attacker to the victim; used in [12][13]. We used these key issues in order to study accuracy of our classification-based defense mechanism.

V. Experimental Results

In our experiment setting, the ANS database contains both DNS requests (i.e., legitimate/real and spoofed requests). The sample size used in the experiment is 10000 requests. Meanwhile, in the LRS database contains 100 legitimate/real requests. The response size is given as summation of authentication request and response. It used to calculate amplification factor in our study. Both ANS and LRS run on Debian Jessie 8.0. Our sample user machine runs on Microsoft Window 7 Home Premium while the attacker machine using Kali Linux 1.1c.

We first study on how the performance of our classification-based defense mechanism (so-called *CBD*) influenced by a percentage of true-positive ratios of packets. Such ratios represent in four different ways as follows:

True positive = spoofed packet that detects as spoofed
True negative = real packet that detects as real
False positive = spoofed packets that detects as real
False negative = real packets that detects as spoofed

We measured accuracy of our mechanism based on the amplification factor that given by:

$$\alpha = \frac{\beta}{\gamma} \times 100 \quad (2)$$

where β refers to number of spoofed packet and γ is number of packet arrives in the system, respectively. The CBD shows higher α; it means that better protection made against DNS reflection attacks.

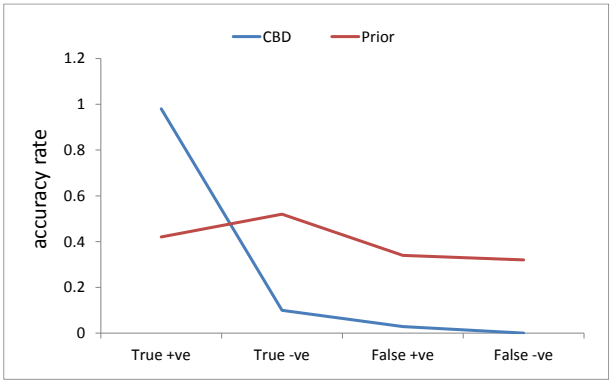

Figure 3: Accuracy in True-Positive Scenarios

The pattern of accuracy in *CBD* (Figure 3) significantly differs compared to *Prior* (i.e., without confirmation status from LRS). There is a tendency of reliability growth towards variability on the communication links when there is verifying procedure applied.

We then measured the effectiveness of CBD for protecting the bandwidth from flooded by bogus traffic. It is indicate as below:

$$X = \frac{Y\ x\ S_{attack}}{Z\ x\ S_{DDS}} \quad (3)$$

where S_{attack} is the size of incoming packet, S_{DDS} is the size of authentication packets, Y refers to the number of spoofed packets that went through the communication channel and Z represents the number of validation and confirmation packets that used for each packet in the defense mechanism, respectively.

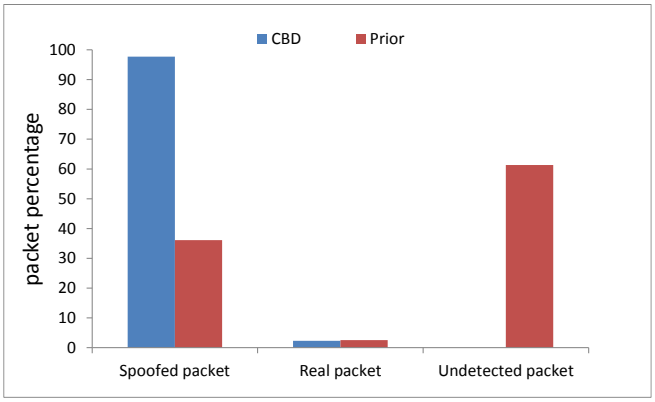

Figure 4: Number of Packet Detected

The effectiveness of *CBD* in gaining better performance presented in Figure 4. Interestingly, the classification method enables more packets to be accurately checked in the network especially during flooded attack. In addition, apparently there is huge number of packet not being detected by *Prior* that implicitly leads to insecure communication link.

VI. CONCLUSION

The domain name can be maliciously used as DNS reflection/amplification attacks. Such attacks flooded the users' machines with large number of incoming DNS responses, then paralyze it. In this work, we presented a distributed defense scheme that aims to effectively detect the DNS reflection occurrence. Specifically, our defense scheme detected the spoofed responses through high amplification factor before they can reach the users' machines. Our detection strategy developed based on classification-based filtering mechanism that implicitly leads to improve the system accuracy. By filtering and discarding the spoofed responses, our defense scheme allows only the legitimate requests to get a (right) response. Our experimental results confirm that the detection strategy through distributed mechanism helps to increase system reliability in communication network. In near future, we aim to analyze communication complexity when the classification-based filtering mechanism used in DNS system.

REFERENCES

[1] Bilge, L., et al. *EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis*. in *NDSS*. 2011.

[2] Anagnostopoulos, M., et al., *DNS amplification attack revisited*. Computers & Security, 39: p. 475-485, 2013.

[3] Di Paola, S. and D. Lombardo, *Protecting against DNS reflection attacks with Bloom filters*, in *Detection of Intrusions and Malware, and Vulnerability Assessment (LNAI)*, Springer. p. 1-16, 2011.

[4] Chow, S.T., D. Wiemer, and J.-M. Robert, *Distributed defence against DDoS attacks*, Google Patents, 2007.

[5] Rafiee, H., M. von Löwis, and C. Meinel, *Challenges and Solutions for DNS Security in IPv6.* Architectures and Protocols for Secure Information Technology Infrastructures: p. 160, 2013.

[6] Marrison, C., *DNS as an attack vector–and how businesses can keep it secure.* Network Security, 2014(6): p. 17-20, 2014.

[7] Rossow, C. *Amplification Hell: Revisiting Network Protocols for DDoS Abuse*. in *NDSS*. 2014.

[8] Colella, A. and C.M. Colombini, *Amplification DDoS Attacks: Emerging Threats and Defense Strategies*, in *Availability, Reliability, and Security in Information Systems*, Springer. p. 298-310, 2014.

[9] Ye, X. and Y. Ye, *A practical mechanism to counteract DNS amplification DDoS attacks.* Journal of Computational Information Systems, **9**(1): p. 265-272,2013.

[10] Krishnan, S. and F. Monrose. *DNS prefetching and its privacy implications: when good things go bad*. in *Proceedings of the 3rd USENIX conference on Large-scale exploits and emergent threats: botnets, spyware, worms, and more*. USENIX Association, 2010.

[11] Schomp, K., et al. *On measuring the client-side DNS infrastructure*. in *Proceedings of the 2013 conference on Internet measurement conference*. ACM, 2013.

[12] Chambers, J.T., *Cisco Systems 2014 Annual Report*, 2014.

[13] Zargar, S.T., J. Joshi, and D. Tipper, *A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks.* Communications Surveys & Tutorials, IEEE, 15(4): p. 2046-2069, 2013.

[14] Vixie, P. and V. Schryver, *Dns response rate limiting (dns rrl).* URL: http://ss. vix. su/~ vixie/isc-tn-2012-1. txt, 2012.

[15] Mansfield-Devine, S., *The growth and evolution of DDoS.* Network Security, 2015(10): p. 13-20, 2015.