KJAR
Kurdistan Journal of Applied Research

Preventing DNS misuse for Reflection/Amplification attacks with minimal computational overhead on the Internet

Rebeen R. Hama Amin
Network Department
Computer Science Institute
Sulaimani Polytechnique University
Sulaymaniyah, Iraq
rebeen.rebwar@spu.edu.iq

Dana Hasan
Computer Science Department
College of Science
University of Garmian
Kalar, Sulaymaniyah, Iraq
dana.hasan@garmian.edu.krd

Masnida Hussin
Department of Communication Technology and Network
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
Serdang, Selangor, Malaysia
masnida@upm.edu.my

Article Info

Volume 5 - Issue 2 - December 2020

DOI: 10.24017/science.2020.2.6

Article history:

Received: 01 November 2020
Accepted:03 December 2020

Keywords:

DNS,
DDM,
Reflection/amplification,
DDoS,
Amplification Factor.

ABSTRACT

DNS reflection/amplification attacks are types of Distributed Denial of Service (DDoS) attacks that take advantage of vulnerabilities in the Domain Name System (DNS) and use it as an attacking tool. This type of attack can quickly deplete the resources (i.e. computational and bandwidth) of the targeted system. Many defense mechanisms are proposed to mitigate the impact of this type of attack. However, these defense mechanisms are centralized-based and cannot deal with a distributed-based attack. Also, these defense mechanisms have a single point of deployment which leads to a lack of computational resources to handle an attack with a large magnitude. In this work, we presented a new distributed-based defense mechanism (DDM) to counter reflection/ amplification attacks. While operating, we calculated the CPU counters of the machines that we deployed our defense mechanism with which showed 19.9% computational improvement. On top of that, our defense mechanism showed that it can protect the attack path from exhaustion during reflection/amplification attacks without putting any significant traffic load on the network by eliminating every spoofed request from getting responses.

1. INTRODUCTION

Reflection/amplification attacks are from the Distributed Denial of Service (DDoS) attacks category that utilizes Domain Name Servers (DNS) as the attacking tool. In this genre of attacks, an attacker fabricates spoofed DNS requests packets with the Internet Protocol (IP) address of their victims. Since DNS utilizes User Datagram Protocol (UDP) which is a connectionless protocol with no handshaking mechanisms or techniques as transport layer protocol. Therefore, the DNS server responds to the victim instead of the source of the packet. On top of that, the attacker seeks response types that are many folds larger than the corresponding request which results in an amplified response. The ratio between response size to request size is called Amplification Factor (AF). Furthermore, introducing DNS Security Extension (DNSSEC) made the matter worse. It increases the size of DNS responses from 512 bytes to 4096 bytes which significantly increases the amplification factor. Though DNSSEC is important for protecting DNS servers from attacks such as cache poisoning. However, it makes an attacker's job easier when it utilizes it to generate larger traffic volume with minimal effort [1][2][3][4][5].

When the response arrives at the victim's machines it will be dropped without any use of it. Thus, a portion of the victim's resources (I.E. computational and bandwidth) of the victim is wasted. However, an attacker usually uses many machines to perform such attacks to magnify the effectiveness of these types of attacks; therefore, the victim machine would be bottlenecked and completely paralyzed [6][7]. Figure 1 shows how reflection/amplification attacks occur. Solving the problem with DNS reflection/amplification attack is not an easy task. First, DNS packets are allowed to pass through every security barrier. Second, DNS utilizes UDP without any built-in authentication mechanism. Third, previous defense mechanisms are centralized-based and they have a single point of deployment (i.e. source, intermediate-network, destination). To be able to confront a distributed based attack, we need to have a distributed-based defense mechanism that can be deployed at multiple nodes on the Internet. Also, distributed defense mechanisms can provide more resources while tackling distributed-based attacks which can lower the bottleneck on the defense mechanism [8][9].

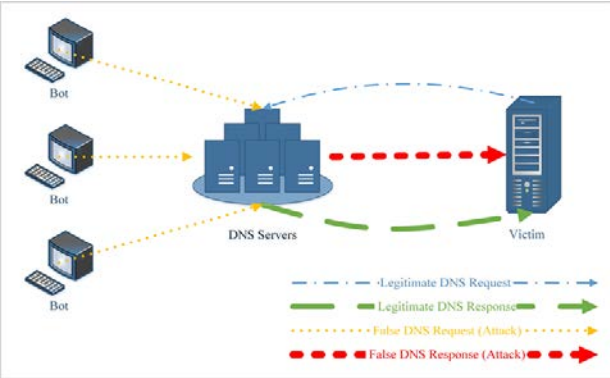

Figure1: Reflection/amplification attack

In this work, we present a distributed-based defense mechanism to tackle the DNS reflection/amplification attack with minimal consumption of computational resources at a single node. It is a hybrid of intermediate-network-based and destination-based defense mechanism. Our strategy is to provide a defense mechanism lower computational overhead during reflection/amplification attacks.

The rest of the paper is organized as follows. Section2 is about Types of defense mechanisms that discuss different types of different mechanisms based on their launching point on the network. Section 3, is about previous works proposed to protect DNS from the reflection/amplification attack. Section 4, is about our proposed mechanism and metrics. Section 5, discusses the results of our works. Section 6, concludes the outcome of this study, providing recommendations and future work of the study.

2. RELATED WORKS

The primary goal of defense mechanisms is detecting the attack's traffic as soon as possible and prevent them from causing any damages as close as possible to their sources. There are two main types of defense mechanisms used to detect and prevent any DDoS-based attacks, centralized, and distributed. This categorization had set based on the defense mechanism's deployment point on the network.

In centralized-based mechanisms, detecting and responding are mostly done in a very centralized manner either by one of the two model approaches which are the deployment points (e.g., source-based, destination-based) or by the responsible points model approach which is within the group of deployment points (e.g., network-based). However, these deployment points have no cooperation between them to tackle an attack volume. Source-based defense mechanisms are deployed closest to the sources of the attack to prevent network users from generating any sort of flooding attacks on DNS. The advantages of these mechanisms are first to detect and respond (i.e., filter) to the offensive or malicious traffic at the source and before wasting any more computing resources. However, the disadvantages are mostly related to the form that sources are distributed among different domains. Hence, accurately detecting and filtering attack flows is a very difficult task for each of the sources [8]. Detection and response in the destination-based defense mechanisms are mostly performed at the destination site of the attack (i.e., victim). The advantage of this kind of defense mechanism is that it is smooth and cost-effective compared to other attack detection techniques because of its ability to access the aggregate traffic adjacent to the destination nodes. The disadvantage is that it is not possible to detect and respond accurately to the attack in advance and when the attack reaches the victims it will waste computing resources on the paths toward the victim. Intermediate network-based mechanisms deployed internally inside networks and mainly on the routers of the Autonomous Systems (AS). The advantage is that it is possible to detect and respond to the attack traffic flow at the intermediate networks and also can be placed closest to the source. On the other hand, the disadvantage is that it causes high storage, processing, and computational overhead on the routers [8].

In contrast to centralized defense mechanisms, distributed defense mechanisms can be deployed and placed at multiple locations including source, destination, and intermediate networks with cooperation among the deployment points. The advantage of the distributed approach is that these mechanisms can robustly stand against flood attacks. Also, this will provide more computational resources at different levels (e.g., destination, source, and network) to prevent and tackle the attacks. The disadvantages of this type are the computational overhead and complexity due to the cooperation and communication among distributed components shared all over the Internet. Also, the lack of incentives for the service provider to collaborate and cooperate, which requires trustworthy communication among various distributed components to perform such actions [8].

In [10], the authors proposed moving to an alternative Internet architecture such as Content-Centric Networking (CCN) that could prevent and eliminate numerous numbers of existing Denial of Service (DoS) attacks. Specifically, CCN will eliminate the probability of DNS-based amplification attacks because there will be no need for DNS services anymore. In CCN, ISPs will be able to cache data based on system services and user requests by deploying content routers. However, CCN uses the request and response approach in the form of "Interest" and "Data" packets, and it is not yet confirmed to what level of resiliency this new network model belongs to be able to prevent amplification attacks or new approaches of DoS attacks proposed designs.
[11] proposed an approach that would reduce the efficiency of amplification attacks on the current Internet to modify and increase the size of requests specifically in vulnerable protocols.

This will reduce the amplification factor, but the growing amount of traffic across the Internet will not be as expected. Furthermore, suggesting that making all protocols amplification free is not logical. For instance, it is entirely sensible to expect that request sizes is smaller or less than the response, especially in large file cases. Moreover, DNS providers already started to minimize the use of the DNS "ANY" records to mitigate the amplification attacks' growing danger. CloudFlare, which is a DNS provider, recently started to resolve and responds with RCODE 4 "Not Implemented". However, the issues of time-consuming and latency still exist while identifying and disabling services with high request and response rates. Therefore, this can lead to the loss of functionality which is one main factor that applications attacks rely on.

[12] proposed a method to configure DNS servers to respond to a specific set of requests from each IP address within a pre-configured time interval. When using newer versions of Bind, Response Rate Limiting (RRL) support can be easily applied. However, RRL protects against the abuse of a single amplifier. An attacker can easily search for a significant number of different amplifiers at a low request rate to attack his/her victim in a very short period.

Reflection and amplification attacks are mostly sending requests using a spoofed IP address of the targeted node. One of the most effective methods to stand and countermeasure against amplification and other DoS attacks is filtering spoofed packets [13][14]. However, recent studies stated that approximately 25% of Authoritative Name Servers (ANS) are not filtering spoofed IP packets which will give the ability to launch amplification attacks by the attackers, and this indicates that the issue still exists. Moreover, several studies have stated that spoofed IP packets still traverse and cross from almost 20% of the entire Internet.

In [15] the authors implemented a detection and mitigation sketch technique to impact and prevent amplification attacks. The sketching technique enables a method to detect and work with large-volume traffic during amplification attacks. They used Chinese Reminder Theorem-based Reversible Sketch to gather and collect the network traffic directly and track any disruption in one-to-one mapping between network transactions (i.e. requests and responses) to recognize amplification attack traffic. Since the detection mechanism does not require every complicated traffic features of amplification attacks, it can be considered an efficient technique.

The authors in [16] used a strategy in Software Defined Networks (SDN)-based networks to mitigate the effects of amplification attacks in DNS. They used the central management feature of SDN networks to take action in reducing the impact of amplification attacks without disturbing any legitimate packets. Their strategy is to monitor both the Time-to-Live (TTL) header and amplification factor and put them into time-series databases. When an anomaly appears, an alarm is initiated which starts the mitigation process. Their experiments showed that their strategy can be used reactively with SDN-based networks. Their results also showed that this method can be used to prevent other UDP-based reflection/amplification attacks as well. However, this method can only be used with SDN networks, and using it in legacy networks is not practical, unlike DDM which proposes a practical distributed solution for this sort of attack, because in the existent of distributed attacks it logical to have a distributed defense system instead of centralized defense mechanisms.

3. PROPOSED MECHANISM

The distributed-based defense mechanism (DDM) in this work is proposed to prevent the DNS reflection/amplification attack by applying the DNS authentication mechanism with classification filtering. The authentication mechanism confirms the legitimacy of incoming DNS responses. Then, classification filtering removes every packet which cannot be confirmed by the authentication mechanism.

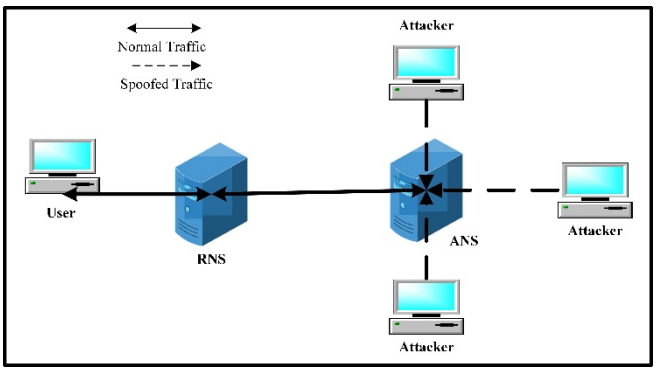

Figure 2: Experimental setup

As figure 2 shows, Recursive Name Server (RNS) is connected to the user's machine, Authoritative Name Server (ANS), and it provides name resolution for the User. ANS contains a website and the server itself is an authoritative server that can provide name resolution for RNS based on its pre-defined configuration. The user's machine asks for name resolution from RNS. Attackers' machines try to attack RNS sends spoofed packets to ANS with the IP address of RNS.

The experiment is done using a Lenovo Ideapad Y50-70 with a 2.2 GHz Core i7 processor, 16 GB RAM, 512 GB Samsung 2.5 SATA SSD, and Microsoft Windows 10 as the hosting operating system and virtual Windows Server 2016 as the DNS servers residing on VMWare workstation 15. We used Java programming language to develop DDM. Furthermore, we used C language to develop DNS Flooder to formulate our attack, Microsoft SQL Server 2008 as our database software. We also developed Packet Capture tool to capture the network traffic and put it inside our databases. To collect the data from CPU counters, we utilized Microsoft Performance Monitor which is a built-in tool inside Microsoft windows.

3.1. Distributed-based defense mechanism (DDM)

A distributed-based defense mechanism should be deployed at multiple locations on the network. These nodes should cooperate to prevent reflection/amplification attacks from causing harm to any part of the system. Our strategy is to fortify DNS with security layers by redesigning the querying structure of DNS. RNS contains a table that is designed to store information about every outgoing DNS request. ANS also contains a table that contains information about every incoming DNS packet. The information which is necessary for both tables to have is the packets' source IP, destination IP, Source Port, and Destination port.

When RNS receives a DNS request from a User, it stores the source IP, Destination IP, Source Port, and Destination port of the packet into its table. Then, it sends the packet to ANS. When the packet arrives at the ANS server, it stores the packet's information in its table as well. Before, sending the response back to RNS, it needs to check for the legitimacy of the packet. For that reason, with help of a custom-made Java program, ANS fabricates a UDP packet with no more than 21 bytes of payload to authenticate the incoming DNS request. It sends the packet to the source IP of the incoming request (Which is RNS). RNS receives the packet and compares it with the content of the table. We call this operation DNS Checkpoint. If the packet exists, then that particular record is deleted from its table. Like ANS, RNS also uses a Java custom-made program to send a message back to ANS with a 1-byte payload confirming the legitimacy of the packet. When ANS receives the packet, it will send back the DNS response to RNS. Upon the arrival of the DNS response, RNS saves it into its cache for later use, then sends the response back to the User. However, when a spoofed packet arrives at ANS, it sends an authentication

packet to RNS. When RNS receives the packet, it compares the ANS authentication request to its table. Since no request corresponds to the authentication request sent by ANS, RNS replies a message that falsifies the DNS request that ANS received, figure 3 shows the sequence diagram of how DDM works.

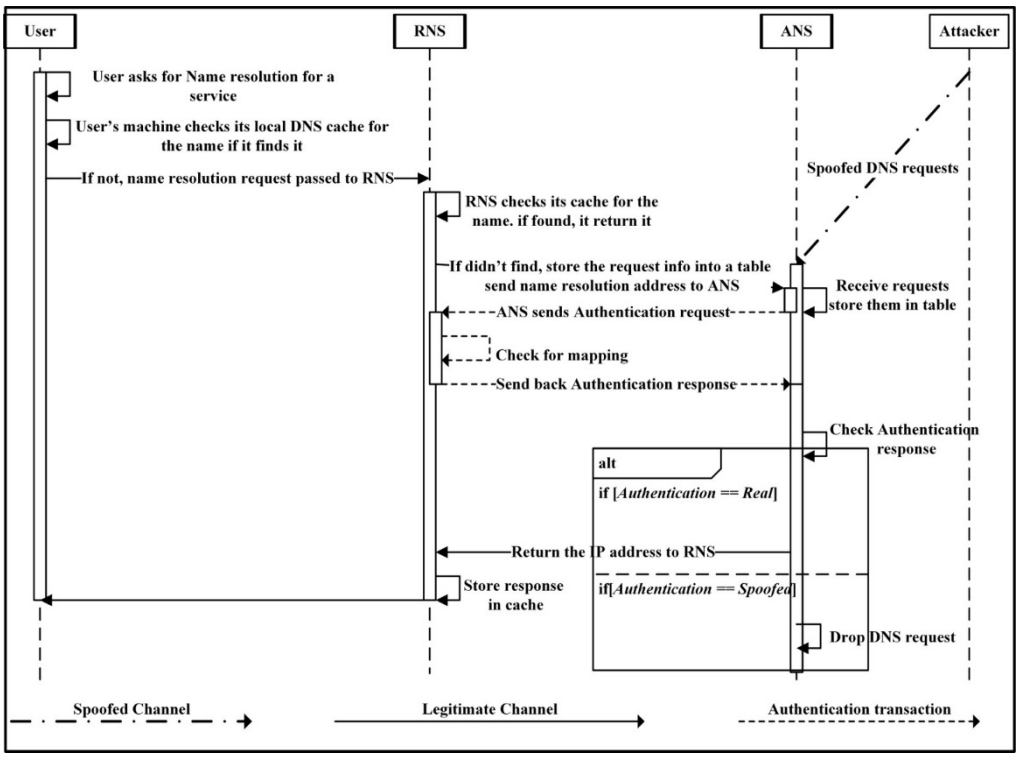

Figure 3: DDM Sequence Diagram

3.2. Performance Metrics

The metric that we considered in this work is CPU counters. The CPU counter which we are interested in is Context/Switching. It occurs when a CPU core switches from executing code on one thread to executing code on a different thread or going to idle. Context/Switching time is overhead; the system does no useful work while switching (15). To collect the data from the CPUs, we utilized Performance Monitor which is a built-in tool inside the Microsoft Windows operating system. Then the data is analyzed and benchmarked with a previous centralized-based defense mechanism to show the difference in their resource consumption during an attack.

We also measured the amount of traffic during each experiment to calculate the volume of traffic during an attack. We benchmarked our results with results from Detecting DNS Amplification Attacks (DDAA) which is a destination-based defense mechanism and Response Rate Limiting (RRL) which is an intermediate-network-based defense mechanism implemented since the release of Bind 9.10.

3.3. Attack formulation

DNS Flooder is used to launch a high volume of traffic with great dynamicity. It is a very powerful tool built using C language with a very complex traffic signature. It can unleash a massive number of spoofed requests using "ANY" records which results in a large response size. The tool is operating at Attacker machines and sends spoofed packets with the address of RNS to ANS. Then, ANS reflects and amplifies the response back to RNS.

4. EXPERIMENTS AND RESULTS

4.1. Experiments

After operating every machine, testing the links between them, and checking whether the databases are working properly, we start the simulation. The User sends DNS requests to RNS to perform name resolution and RNS stores them in its database, then sends them to ANS. Also, Attackers send spoofed requests to the ANS machine to be reflected in the RNS machine. Upon receiving DNS requests, ANS start sending authentication messages to RNS, and RNS replies to them. The arrival of authentication responses triggers the classification filtering directly. ANS drops any DNS request with a falsified authentication response based on the authentication response from RNS. Also, ANS sends back DNS responses to RNS for every successfully authenticated request, figure 4 shows a sample of DNS requests stored in the database.

Source IP	Destination IP	Source Port	Destination Port	Status
192.168.1.1	192.168.1.100	20484	53	Legitimate
192.168.1.1	192.168.1.100	20484	53	Spoofed
192.168.1.1	192.168.1.100	30894	53	Spoofed
192.168.1.1	192.168.1.100	30894	53	Spoofed
192.168.1.1	192.168.1.100	30894	53	Spoofed
192.168.1.1	192.168.1.100	30894	53	Spoofed
192.168.1.1	192.168.1.100	30894	53	Spoofed
192.168.1.1	192.168.1.100	30894	53	Spoofed
192.168.1.1	192.168.1.100	13022	53	Legitimate

Figure 4: DNS requests stored in databases

4.2. Context-Switching

The results of our simulation are collected using the Performance Monitor tool in Microsoft windows. Since most of the actions take place on the ANS machine; therefore, we collected the CPU counter data from the ANS machine, specifically CPU context/Switching. The data are extracted and analyzed using Microsoft Excel 2016. The analyzed results are benchmarked with a centralized-based defense mechanism. Figure 4, shows the results of CPU context/switching.

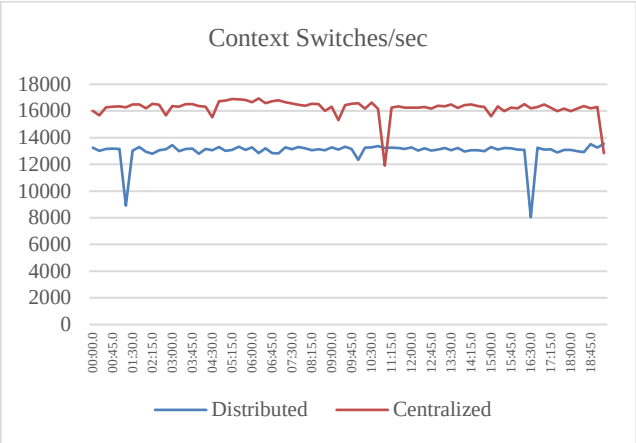

Figure 5: Context Switching per second

As figure 5 shows, the Distributed mechanism shows lower context/switching per second. This means that the operating system switches between different threads of the same process less frequently. When context/switching occurs, the state of the running process is saved in a portion of memory called Process Control Block (PCB). Then the system starts loading the PCB for another process. During the storing and loading of the process's PCBs, the computer is doing nothing. This indicates that context switching generates overhead for the system [17]. This is a demonstration that the proposed defense mechanism can operate with less context/switching per second which is an indication of less overhead on the system.

4.3. Link Utilization

Link usage or link utilization is the amount of successful message transmission through a communication. The analysis of the outcomes of simulations shows different results about link usage of name resolving transactions.

4.3.1. Normal DNS transactions

The purpose of proposing DDM is to allow legitimate users to get responses and discard all attack traffics when the system undergoes a massive flooding attack. due to the authentication procedure by DNS Checkpoint, each DNS transaction (i.e. request and response) consumes a slight portion of the bandwidth which is not more than 21 bytes. The devised experiment showed that the size of DNS requests is 70 bytes, while the size of the response is 501 bytes. DDM adds another 21 bytes for the transaction which is the authentication request and response packets. as shown in figure 6.

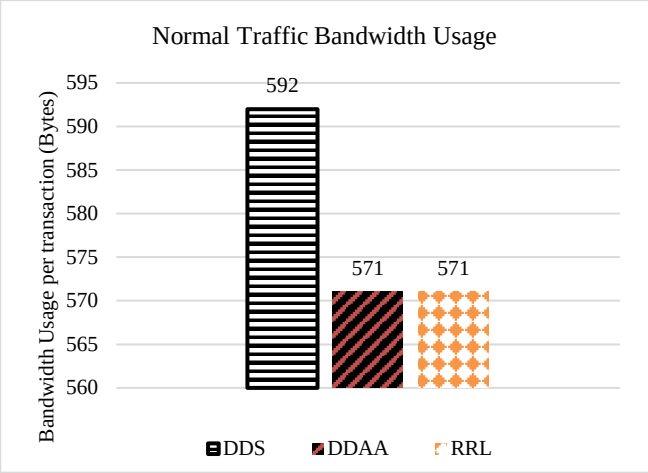

Figure 6: Bandwidth usage per DNS transactions (Normal traffic)

However, both DDAA and RRL are single-point defense mechanisms and they do not have any cooperation between different entities of the network. DDAA is a destination-based defense mechanism that is deployed at the destination of attack. Therefore, it doesn't require any message exchanging with any other entities to protect the network on which it is deployed in it. On the other hand, RRL is deployed at the reflector's side (ANS) and its purpose is to limit the amount of traffic towards the victim. Therefore, they send back only the DNS response which makes their transaction consumes around 21 bytes less bandwidth compared to DDM.

4.3.2. Spoofed DNS transactions

When the victim's machine is subjected to the heavy load of attack traffic due to DNS amplification, the downlink bandwidth towards the victim suffers serious starvation. Single-

point defense mechanisms are not capable of protecting the network link, especially when the attack traffic is highly dynamic. Figure 7 shows the bandwidth consumed for each spoofed DNS transaction.
As shown in figure 7, with DDM, the spoofed addresses receive no responses at all, which leads to an immediate effect on the amount of traffic that passes through the network channel. Therefore, while during the attack each transaction significantly reduces the link bandwidth, DDM countermeasures such effect. The DDM generates only two tiny packets for the authentication purpose. Unlike legitimate traffic, when DDM encounters spoofed packets, it ends up generating only the small authentication message per traffic, which prevents unauthenticated requests from being redirected toward the targeted victim at all.

However, other centralized defense mechanisms allow the victim to receive DNS responses without any mechanism to protect the downlink from those bogus packets. The bandwidth consumption per transaction is equal to the size of spoofed DNS response which is generated by the DNS reflector which is 501 bytes for both DDAA and RRL. Since DDAA is lunched at the destination of attack, therefore the upstream networks on the attack path suffer starvation due to the bandwidth usage. As an intermediate-based defense mechanism, RRL does not reduce the size of the traffic, it just limits the number of responses that are targeting the victim.

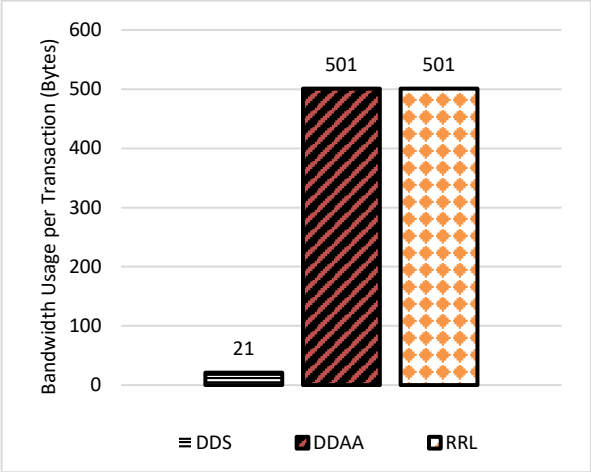

Figure 7: Bandwidth usage per transaction (Attack traffic)

4.3.3. Traffic Volume
DDM adds two new packets with tiny size to DNS name resolution transactions, which slightly increases the link usage per transaction. However, it significantly reduces the massive impact of spoofed responses along with the downlink and the entire network channels. The fact is that preventing spoofed requests from getting responses has a direct effect on total bandwidth. Figure 5 demonstrates the impact of eliminating the amplification factor on the bandwidth as per each replication.

As shown in the figure, DDM significantly minimizes the bandwidth exhaustion along the victim's path while the system is under an attack. This reduction happens due to the usage of DNS Disinfector. Even though DDM querying size is increased by 21 bytes for one packet in each attack flow, it is ability to reduce the traffic volume to a minimum value. This is because

DDM grants DNS response to legitimate DNS requests only, and the spoofed ones are discarded. Moreover, DDM performs a detection mechanism on traffic flow, not individual packets. Therefore, the total amount of traffic that passes through the system significantly decreased.

5. CONCLUSION

Reflection/Amplification attack is one of the worst types of attacks deployed by cyber attackers on the Internet, which utilizes DNS for its deployment. The main reason that makes DNS a target for these types of attacks is that DNS uses UDP as a transport protocol. UDP is an unreliable protocol with source authentication issues. Therefore, an attacker can simply forward bulk request traffic toward DNS servers and fabricating the source IP address of their victim to be reflected to their target. Nevertheless, DNS response queries are larger than the corresponding request query. It causes the reflected traffic toward the target to be greater than the request traffic.

We have proposed DDM, which is a distributed mechanism to detect and prevent DNS reflection/amplification attacks with lower overhead on network computational resources compared to centralized base defense mechanisms. DDM introduces the authentication mechanism for DNS queries which puts a security layer on DNS service. Also, DDM implements a classification filtering strategy that can be triggered only when spoofed traffic is detected. Our analysis showed that implementing DDM can provide a defense mechanism with lower computational overhead by reducing the context switching between different processes while it is working. Furthermore, it protects upstream networks from exhaustion during attacks with a slight network overhead on the attack path.

6. RECOMMENDATION AND FUTURE WORK

We implemented this work in a controlled experimental environment. Also, we neglected any lost DNS packets during our experiments. Therefore, we recommend that this work should be implemented in a realistic testbed to measure the actual potential of DDM in mitigating DNS reflection/amplification attacks in a real-world attacking scenario. In the future, we intend to implement DDM for recursive servers along with authoritative name servers. Also, implementing DDM in such a way that it can be operated only during reflection/amplification attacks occurrence.

REFERENCE

[1] R. van Rijswijk-Deij, A. Sperotto, and A. Pras, “DNSSEC and its potential for DDoS attacks,” in Proceedings of the 2014 Conference on Internet Measurement Conference - IMC ’14, 2014, pp. 449–460, doi: 10.1145/2663716.2663731.

[2] J.-Y. Bisiaux, “DNS threats and mitigation strategies,” Netw. Secur., vol. 2014, no. 7, pp. 5–9, Jul. 2014, doi: 10.1016/S1353-4858(14)70068-6.

[3] M. Anagnostopoulos, G. Kambourakis, P. Kopanos, G. Louloudakis, and S. Gritzalis, “DNS amplification attack revisited,” Comput. Secur., vol. 39, pp. 475–485, Nov. 2013, doi: 10.1016/j.cose.2013.10.001.

[4] Y. Koç, A. Jamakovic, and B. Gijsen, “A global reference model of the domain name system,” Int. J. Crit. Infrastruct. Prot., vol. 5, no. 3–4, pp. 108–117, Dec. 2012, doi: 10.1016/j.ijcip.2012.08.001.

[5] S. Abbasi, “Investigation of open resolvers in DNS reflection DDoS attacks,” Université Laval, 2014.

[6] C. Marrison, “DNS as an attack vector – and how businesses can keep it secure,” Netw. Secur., vol. 2014, no. 6, pp. 17–20, Jun. 2014, doi: 10.1016/S1353-4858(14)70061-3.

[7] X. Ye and Y. Ye, “A practical mechanism to counteract DNS amplification DDoS attacks,” J. Comput. Inf. Syst., vol. 9, no. 1, pp. 265–272, 2013.

[8] S. T. Zargar, J. Joshi, and D. Tipper, “A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks,” IEEE Commun. Surv. Tutorials, vol. 15, no. 4, pp. 2046–2069, 2013, doi: 10.1109/SURV.2013.031413.00127.

[9] B. Liu et al., “SF-DRDoS: The store-and-flood distributed reflective denial of service attack,” Comput. Commun., vol. 69, pp. 107–115, Sep. 2015, doi: 10.1016/j.comcom.2015.06.008.

[10] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking named

content," in Proceedings of the 5th international conference on Emerging networking experiments and technologies - CoNEXT '09, 2009, p. 1, doi: 10.1145/1658939.1658941.
[11] C. Rossow, "Amplification Hell: Revisiting Network Protocols for DDoS Abuse," in Proceedings 2014 Network and Distributed System Security Symposium, 2014, doi: 10.14722/ndss.2014.23233.
[12] P. Vixie and V. Schryver, "Response Policy Zones," Internet Engenieering Task Force, p. 10, 2017.
[13] M. Kührer, T. Hupperich, C. Rossow, and T. Holz, "Exit from hell? Reducing the impact of amplification DDoS attacks," Proc. 23rd USENIX Secur. Symp., pp. 111–125, 2014.
[14] S. Di Paola and D. Lombardo, "Protecting against DNS Reflection Attacks with Bloom Filters," 2011, pp. 1–16.
[15] X. Jing, J. Zhao, Q. Zheng, Z. Yan, and W. Pedrycz, "A reversible sketch-based method for detecting and mitigating amplification attacks," J. Netw. Comput. Appl., vol. 142, pp. 15–24, Sep. 2019, doi: 10.1016/j.jnca.2019.06.007.
[16] K. Ozdincer and H. A. Mantar, "SDN-based Detection and Mitigation System for DNS Amplification Attacks," in 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Oct. 2019, pp. 1–7, doi: 10.1109/ISMSIT.2019.8932809.
[17] A. Silberschatz, P. B. Galvin, and G. Gagne, Operating System Concepts Essentials. John Wiley & Sons, Inc., 2013.