A Multi-Path Approach to Protect DNS Against DDoS Attacks

Sahel Alouneh

German Jordanian University, Amman, Jordan
Al Ain University, Abu Dhabi, UAE
E-mail: sahel.alouneh@gju.edu.jo; sahel.alouneh@aau.ac.ae

Received 06 January 2023; Accepted 29 March 2023;
Publication 21 June 2023

Abstract

Domain Name System (DNS) is considered a vital service for the internet and networks operations, and practically this service is configured and accessible across networks' firewall. Therefore, attackers take advantage of this open configuration to attack a network's DNS server in order to use it as a reflector to achieve Denial of Service (DoS) attacks. Most of protection methods such as intrusion prevention and detection systems use blended tactics such as blocked-lists for suspicious sources, and thresholds for traffic volumes to detect and defend against DoS flooding attacks. However, these protection methods are not often successful. In this paper, we propose a new method to sense and protect DNS systems from DoS and Distributed DoS (DDoS) attacks. The main idea in our approach is to distribute the DNS request mapping into more than one DNS resolver such that an attack on one server should not affect the entire DNS services. Our approach uses the Multi-Protocol Label Switching (MPLS) along with multi-path routing to achieve this goal. Also, we use threshold secret sharing to code the distributed DNS requests. Our findings and results show that this approach performs better when compared with the traditional DNS structure.

Keywords: DNS, DoS, multipath routing, security, MPLS.

Journal of Cyber Security and Mobility, Vol. 12_4, 569–588.
doi: 10.13052/jcsm2245-1439.1246

1 Introduction

Domain name systems are key network identification systems used to discover network devices and resources that are accessible over the internet and computer networks. The DNS system transforms the network resource IP address into a domain name to make it easier to recognize and remember rather than an IP address, and it also supports easy and simplified access to Internet resources [1]. There are two meth-ods used by DNS systems to achieve domain/address resolution: (1) recursive query and (2) iterative query. The Iterative query denotes that Local Name Server (LNS) com-municates directly with an Authoritative Name Server (ANS), or with top-level name or root name servers [2]. However, the recursive query is initiated between a local name server and a higher domain name server to obtain the requested IP address.

Although the DNS systems is vital to the function of internet, it is still one of the under protected systems. DNS attacks can lead to devastating losses and thus making networks or the Internet inaccessible. Moreover, DNS servers can be misused and therefore be used as attacking agents to perform distributed denial of service attacks. Likewise, DNS spoofing can be used by malicious intruders to create websites that tru-ly match legitimate websites while indeed hosting traps. Nevertheless, defending DNS services' infrastructure is an evolving concern, and thus there is tremendously high awareness of this need to protect DNS services and systems.

The key contribution of this research work firstly demonstrates the status of cur-rent state-of-the-art practices and methods used to protect DNS systems. Then, we propose a new solution to protect the DNS systems against cyber-attacks using MPLS and multi-path routing techniques.

2 Data Centre Infrastructure Background

2.1 DNS Background Concepts

A DNS attack normally targets the availability and reliability of DNS services. The attack targets the DNS infrastructure and the goal is to render the DNS service in order to make it unreachable and to disrupt the DNS server's response. It is worth to note that DNS servers consist of two different modules, recursive and authoritative servers and thus attacks against each one of them varies. The structure of DNS servers is vulnerable to network-based attacks such as Internet Control Message Protocol/Transport Control Protocol (ICMP/TCP) flooding attacks. Likewise, attackers can cause the

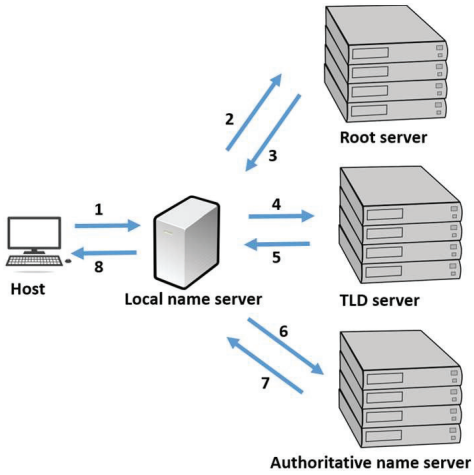

Figure 1 Domain Name System query steps.

DNS software or operating system create invalid entries, and ultimately causing the DNS server to crash or fail to respond to valid and legitimate requests.

To begin with, we need to highlight the DNS basic structure in order to be able to understand how DNS attacks happen. Figure 1 shows the DNS query steps. First, the host asks the LNS server for the wanted domain name address, and LNS server sends the response directly to the host if it has the address mapping. Otherwise, the LNS server delegates the host query to the root server. The root server responds to the request by directing the LNS server to contact the concerned Top Level Domain (TLD) server. After contacting the concerned TLD server, the LNS server should now have received the right contact for ANS server and thus should now be able to send the actual IP address to the host [3].

In the next section, we explore the related work in the literature concerning the DNS security.

2.2 Related Work

DNS systems are one of the most essential services of the Internet and simply act as the phonebook of the internet. The security and reliability of DNS systems are of great concern and of significant importance. There are several research works which tackled the security issues in DNS systems. We briefly in this section present the most recent ones, and we mainly focus on the recent

papers which are concerned with the DDoS/DoS DNS attacks. Chen et al. [4] have concluded that the volume of DDoS based attacks increase swiftly in recent years and when attacks happen, not only the authoritative servers were targeted and affected, but top TLDs were also targeted and suffered from these attacks. A protection strategy against DDoS attack not only should keep in mind the systems services manageable but also should consider to offer a proper Quality of Service (QoS), and therefore, the authors in reference [5] have introduced the use of Differentiated Services (DiffServs) and policies within network routers to decrease the effect of DDoS attacks.

According to the statistics of Kaspersky Labs in reference [6], DDoS attacks have continued evolving and hitting different internet and networking infrastructures and services.

We list here the main attack surfaces and research works proposed to countermeasure against such attacks. Research works in references [7, 8] tackle the DNS amplification attacks which compromise today's Internet. On other hand, DNS cache poisoning is another attack surface where a DNS system resolver cache can be compromised by the insertion of illegal domain names or IP addresses to redirect a user request to the attackers malicious or controlled servers or services. On the other hand, the research works in references [9, 10] have attempted to address cache poisoning attacks on DNS systems. Botnet Attacks Using DNS is another attack surface type used by attackers to let them access devices using coordination between network devices to accomplish a wide range of malicious attacks, such as the DDoS attack. In literature, there are research proposals which studied botnets and solutions to protect against them such as the work in references [11, 12]. Nevertheless, such attacks are until today on the rise and endangering the DNS security and its functionality. Another prevailing DNS attack surface is caused by phishing attacks/DNS Manipulation and thus may cause a number of threats, such as phishing and malicious domains. Therefore, a large number of studies considered DNS phishing and manipulation attacks as in [13, 14 and 15].

The author of reference [16] concentrates handles the DNS systems security vulnerabilities by identifying DoS and DDoS attacks using an agent that is running on a hardware and an AI agent is a combination of both hardware and software solutions. The authors in reference [17] also use an AI solution by building an AI model to protect against amplification attacks directly from incoming traffic volumes. They process incoming data using a traffic throttling model with learning reinforcement strategy.

A summary on the related work is shown in Table 1.

Table 1 Summary of related work to DNS security in literature

Reference	Multipath	Encryption	Cache Poisoning	Amplification	Botnets/ DoS	Phishing
Wanga et al. [1]	*Partial*	*Partial*	×	×	×	×
Furfaro [5]	×	×	×	×	×	×
Zheng et al. [7]	×	×	×	/	×	×
Verma et al. [8]	×	×	×	/	×	×
Hao et al. [9]	×	×	/	×	×	×
Wu et al. [10]	×	×	/	×	×	×
Truong et al. [11]	×	×	×	×	/	×
Plohmann et al. [12]	×	×	×	×	/	×
Trevisan et al. [13]	×	×	×	×	×	/
Pearce et al. [14]	×	×	×	×	×	/
Kintis et al. [15]	×	×	×	×	×	/
Singh [16]	×	×	×	×	/	×
Zhnag et al. [17]	×	×	×	/	×	×
Niakanlahijia et al [21]	*Partial*	/	×	×	×	/

3 Proposed Work

The main goal of this research work is to countermeasure DDoS attacks on DNS systems. To achieve this goal, we use the multi-path routing approach to build the DNS infra-structure and its infrastructure networking requirements. Multipath routing approach offers major benefits such as fault tolerance and fast recovery, improved bandwidth utilization, and traffic engineering.

3.1 Supporting Multipath Routing Using Label Switching

To support multipath routing between LNS and ANS servers, conventional IP routing techniques such as TCP/IP will be a costly approach and choice [18]. We propose to use label switching technologies such as the Multi-Protocol Label Switching (MPLS) to support multipath routing capabilities. MPLS technology is nowadays supported by most network technologies in the market-place and Internet Service Providers (ISPs). MPLS technology has been used to support Traffic Engineering – (TE) solutions and high speed networking. MPLS technology provides network operators with the flexibility to re-route traffic around failing or congested links and to sense bottlenecks [19]. Here in this proposed work, we assume that MPLS is the backbone network infrastructure used between DNS servers.

The basic operation of MPLS networks is to identify and classify IP packets at the edge nodes with a fixed-length, short, and local identifiers called labels, and then forward the labeled packets to inner routers/switches

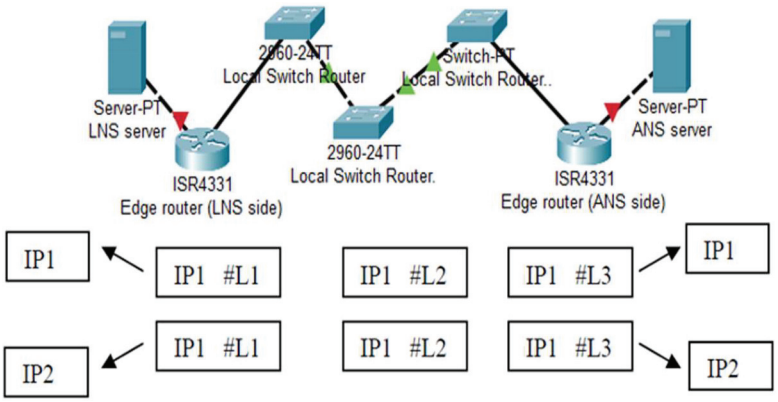

Figure 2 A simulation scenario of DNS-FEC in MPLS enabled networks.

that are modified to recognize and operate with labels instead of IP headers. MPLS nodes use labels information instead of IP layer information to forward the packets over the network.

In MPLS network, an essential mechanism called Forwarding Equivalent Class (FEC) is used for packet identification and classification. FEC is a group of IP packets that are treated similarly by Local Switched Routers (LSRs). Therefore, these IP packets that are forwarded over the same path between a Local and an Authoritative name server (here this path is called LA path) can then be mapped with the same label by MPLS LSR routers although they may have different network layer header information. Figure 2 shows how MPLS networks work where the IP packets are denoted by IP 1 and IP2 and MPLS labels are denoted by L1, L2, and L3. The labels reduce essential information needed for packet forwarding and switching. This includes Quality of Service (QoS) information, and the routing information. The main outcome of this process in MPLS networks is that forwarding and switching decisions are based on labels rather than TCP/IP routing lookup tables. Therefore, MPLS networks improve TE capabilities and reduce significantly traditional IP networking overhead [19].

The use and application of MPLS technology to support multipath routing performs considerably better when compared with conventional TCP/IP networks. When analyzing the characteristics of each network, we summarize the comparison as shown in Table 2. It can be noticed that MPLS performs considerably better than TCP/IP networks. Only four bytes need to be added to each packet in order to support multipath routing, TE, and QoS in MPLS networks while a minimum of 20 bytes are needed in TCP/IP model. On the

Table 2 Multipath performance comparison between TCP/IP and MPLS networks

Core Network	TCP/IP	MPLS
Packet Header overhead	Min of 20 bytes	4bytes (fixed)
Traffic/packet distribution over multiple paths	Requires to involve routing lookup table	Requires label mapping
Time required for traffic/packet distribution over multiple paths	High	Low
Traffic engineering complexity	High	Low
Quality of Service complexity	High	Low

other hand, TCP/IP networks requires source routing to support explicit routing and thus this requires more overhead to be added to the IP packet header. Therefore, this makes MPLS networks more appropriate for the protocols that use small packets size such as the packets used by DNS systems.

In the next section, we present our multipath approach to support the security of DNS systems.

3.2 How Multipath Routing Can Improve the DNS Security

To start with, it is necessary to elaborate more on the multipath routing concepts and its applicability in DNS systems. It is worth mentioning that our DNS multipath concept can also be extended to cover other DNS servers such as TLD servers, and the root server. However, in this work we limit our discussion and therefore focus on protecting ANS servers as they are the DNS servers that are mostly targeted by attackers.

To perform DDoS attacks on DNS servers, this is usually made by overwhelming the ANS server with ICMP/TCP flood requests. Therefore, the LNS server fails to serve other legitimate requests and appear to be unavailable or not reachable. In our approach, the basic idea behind the use of multipath routing to secure DNS servers relies on changing the iterative model structure that is being used and implemented by traditional DNS servers. We need the LNS server to accept multipath ICMP/TCP requests arriving from multiple path routes from the LNS server. Now, the ANS server can only recognize a complete ICMP/TCP mapping request for an IP address if K requests have been received from at least K multiple paths. To illustrate more, let us refer to the case when only one single path is available between the LNS and ANS servers, if the attacker is able to exploit one or more than one node along this single path, then the attacker has the potential to perform a DoS attack. However, with the proposed multipath approach, the

probability of having compromised nodes in more than one path should be lower compared to the case of only one single path [20].

Here, the path LA term is used to refer to a path route between LNS and ANS servers. A path is compromised if there are one or more nodes compromised along this path. Now, the probability of a DNS compromised packet PDNS is defined as follows:

$$P_{DNS} = \prod_{i=1}^{n} P_{LA_i} \tag{1}$$

where P_{LA_i} refers to the probability that path *LA* is compromised, and n refers to the number of available paths between LNS and ANS servers. This probability of a compromised path P_{LA_i} can be found according to the following equation:

$$P_{LA_i} = 1 - \prod_{j=1}^{h} (1 - p_j) \tag{2}$$

where p_j refers to the probability that a node is compromised in *LA* path, and *h* refers to the number of nodes in this path. Figure 3 shows the effect of multipath routing in reducing the number of compromised nodes.

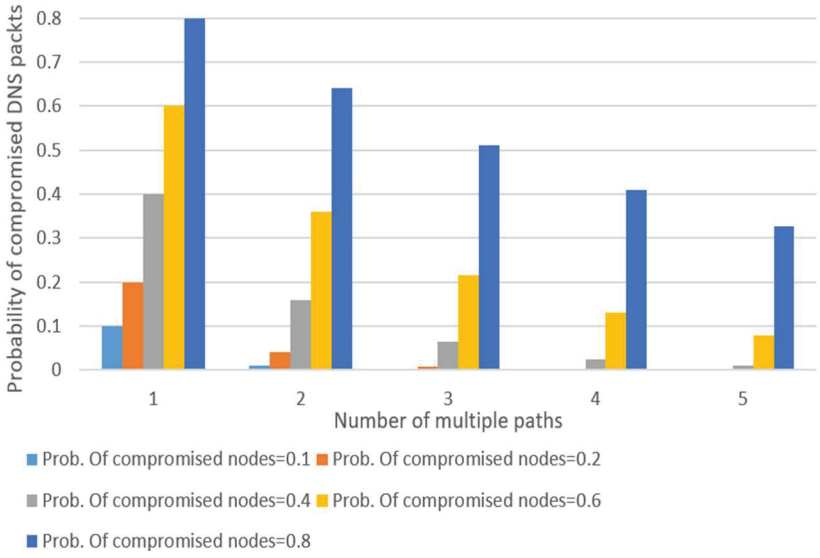

Figure 3 DNS compromised probability compared with different multipath values.

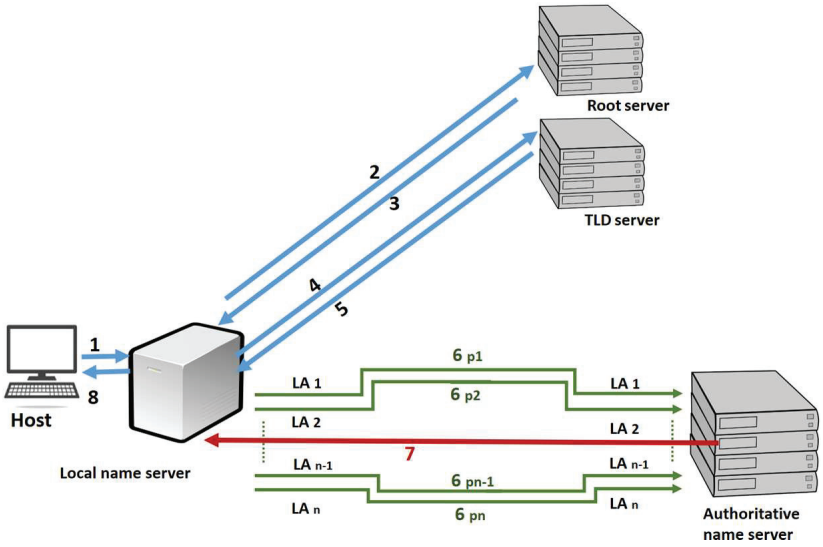

Figure 4(a) DNS query steps with multi-path routing.

In Figure 4(a), the abstract conceptual DNS multi-path design application is shown. The idea here is to require the ANS server to receive multipath *coded* DNS IP address mapping requests Pkt_{DNS} in order accept the establishment of TCP connections. The multipath connection between ANS and LNS servers consists of multi-paths $(LA_1, LA_2, \ldots, LA_{n-1}, LA_n)$ directed toward the ANS server labeled $(6_{p1}, 6_{p2}, \ldots, 6_{pn-1}, 6_{pn})$, and one request mapping directed from the ANS toward the LNS server labeled as 7. Here, the numbers 6 and 7 refers to the steps mentioned in Figure 1.

The multipath selection criteria between LNS and ANS servers should maintain the following conditions:

(i) ANS server assisted with multipath routing and more than two paths are found to be available to direct traffic through the target network.
(ii) Multipath selection at ANS server should consider selecting *disjoint LA* paths. In other words, two *LA* paths are considered to be disjoint if no common *node/link* can be found between both paths.

It is imperative to differentiate between link disjoint or node disjoint multipath connections. The LA paths between LNS and ANS servers are considered to be *node disjoint* if there are no common link(s) and node(s) between LA paths. On the other hand, the LA paths are said to be *link* disjoint if there are no common links between any of the LA paths, though there

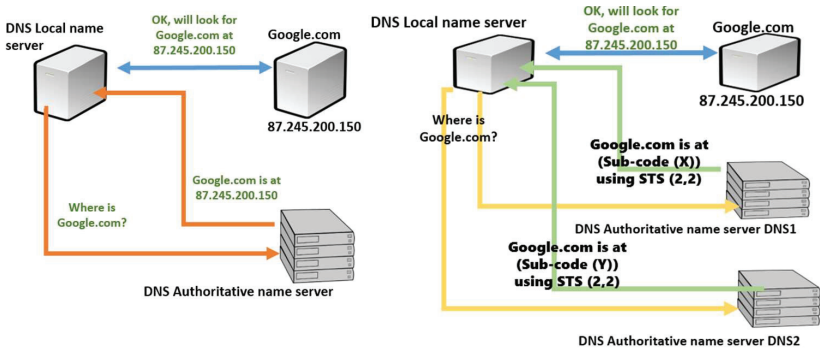

Figure 4(b) The detail of implementation scenario of the multipath approach using a (2,2) STS coding.

might be common node or nodes between LA paths. In this paper, we refer to both cases of node or link disjoint LA paths by the term "*maximally-disjoint" LA* paths. If the condition to have *disjoint* LA paths in point ii) cannot be satisfied, then multipath selection at ANS server should opt to consider selecting *maximally-disjoint LA* paths but this may be less secure choice because of the shared links/nodes between LA paths. In other words, two *LA* paths are considered *maximally disjoint* if common node/link can be found between paths. It is worth to note that *maximally disjoint* paths may result in an increased P_{DNS} probability values. *Algorithm 1* below illustrate the multipath selection process between LNS and DNS servers, mainly ANS or TLD servers. The detail of implementation scenario of this approach is shown in Figure 4(b).

It is worth to mention that multiple LA paths can appear to be *maximally disjoint* at the logical level while at the physical level they may not be. Thus, we assume LA paths are *maximally disjoint* at the physical level, i.e., each path corresponds to diverse Shared Risk Link Groups (SRLG). In MPLS TE, an SRLG is a set of links sharing a common source, which affects entire links in the set if the shared source fails. The links here share equal risk of failure and are consequently considered to fit in to the same SRLG. For instance, links sharing a shared fiber are said to belong to the same SRLG.

3.3 Deploying Multipath DNS Routing with Secret Threshold Sharing

In approach multipath approach, the original DNS request is sent in multipath routes as coded sub requests. One codec method that can be used to encode

Algorithm 1 LNS-ANS Server disjoint multipath selection

The purpose of this algorithm is to find the best maximally disjoint multi-path connection between LNS and ANS servers.

Input: Input: DNS server -> S: (LNS, TLD, ANS)

K: multipath routes, where Kd: disjoint paths, Km: maximally disjoint LA paths.

Assumption 1: K paths between LNS and TLD or ANS are assumed to be computed using reference [16].

Assumption 2: Paths selected should satisfy the path length constraint. Long paths are to be excluded.

Assumption 3: The group of the k LA paths is selected based on the minimum number of overlapping links compared to other groups, and the minimum cost. Cost = {cost1, cost2, ..., $\text{cost}_{\text{STS}-\text{level}}$}.

Step 1: Set C_{path} = The number of candidate LA or LT paths. (LT: Local authoritative server-TLD path)

Set $\text{N}_{\text{GC}} = \binom{C_{path}}{k}$, where N_{GC} is the total number of different groups g which consist of K paths.

Define a two-dimensional array H[r]c] to store the candidate paths.

Initialize:

$$\text{r} = \text{C}_{\text{path}}$$
$$\text{c} = 2 \times \text{Max}(\text{number of links in longest path})$$

Initialize:

$$\text{i}_1 = 0,\ \text{i}_2 = \text{i}_1 + 1, \ldots, \text{i}_\text{k} = \text{i}_{\text{k}-1} + 1$$

Step 2:

if $\text{i}_1 < r - (k-1)$

$\text{i}_2 = \text{i}_1 + 1$

else if $\text{i}_2 < r - (k-2)$

$\text{i}_\text{k} = \text{i}_{\text{k}-1} + 1$

else if $\text{i}_\text{k} < r$

for $p1 = 0; p1 < \frac{c}{2}, p1 = p1 + 1$

Compute $\text{Cost}_{1,2,\ldots,\text{k}}[p1] = \text{H}[\text{i}_\text{k}][p1+\text{c}/2]$

for $p = 0; p < \frac{c}{2}, p = p + 1$

Compute $\text{g}_{1,2,\ldots,\text{k}}[p] = \text{H}[\text{i}_\text{k}][p+\text{c}/2]$

For each group combination:

Disjoint ratio = $(1 - \frac{To}{Tl})$ where To is the total number of overlapping links, Tl is the total number of links in the connection.

Step 3: Compare two LSPs for overlapping links

Step 4: Repeat Steps 1–3 for $\binom{k}{2}$ to cover all combinations of groups

END

DNS requests along the multipath route between LNS and ANS servers (i.e., the LA path) is the Secret Threshold Sharing (STS) [21]. The basic idea of STS is to split data into Q fragments in such a way that data can simply be reconstructed from any J fragments, on the other hand, having $(J-1)$ fragments reveals no information about the DNS request data.

Therefore, the STS approach is used along with multipath routing to support confidentiality and authentication of DNS requests and networking. We call this approach Multipath DNS Security using secret threshold sharing (M-DNS). We combine the STS approach and use it along with multipath routing to support confidentiality and authentication of DNS networking. We call this approach Multipath DNS Security using secret threshold sharing (M-DNS).

3.3.1 Detection of poisoned DNS requests

To support detection of poisoned DNS requests, then we need to apply a (J, Q) STS scheme where $Q > J$. Therefore, redundant packet(s) are needed. The impact of redundant bandwidth overhead can be summarized as follows:

$$\textit{Redundant overhead} = \frac{Q-J}{J} \times (Pkt + 4)\ \textit{bytes} \tag{3}$$

where Pkt denotes the IP packet size, and the number 4 indicates the size of MPLS header.

The detection of modified DNS requests can be achieved by comparing values reconstructed at the ANS server from the different groups of DNS packet requests as shown in Figure 5. The original DNS IP packet is divided into three DNS sub requests or fragments $R1, R2, R3$ and allocated into $LA1, LA2$, and $LA3$ respectively using a (2, 3) STS scheme. Each sub DNS fragment represents a sub value of the original DNS request denoted as $R_{original}$, and it is encapsulated in MPLS packet. At ANS server side, the re-construction process requires the assembly of at least two fragments in order to reconstruct the original DNS request.

In Figure 5, the total number of groups are 3 ((R1,R2), (R1,R3), and (R2,R3)). In Figure 5-(a), all groups should reconstruct the same value, i.e., the original DNS request packet and thus this indicates that no attack or data change has occurred. Nevertheless, in Figure 5(b,c,d) if one or more LA paths have been attacked, then reconstructed values will not be the same for all groups and thus this indicates the possibility of attack(s) occurrence. Therefore, the ANS server should act accordingly and may consider this case as a poisoned DNS request and thus reject it.

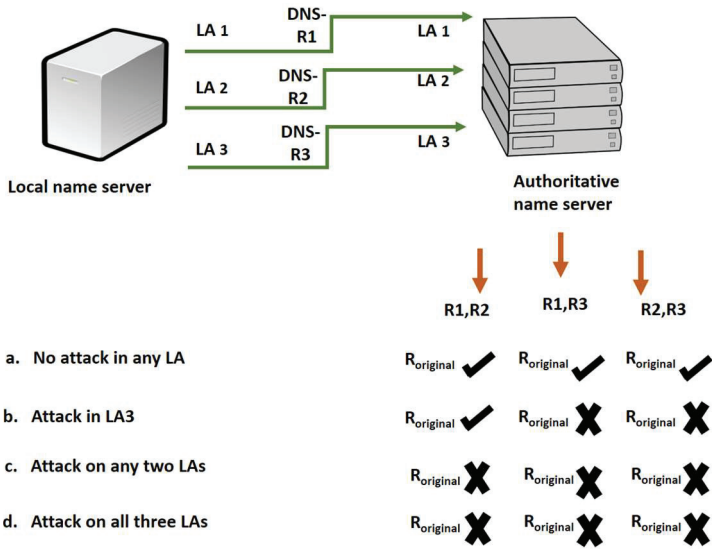

Figure 5 M-DNS Case scenario using a (2,3) STS scheme, (a) No attack (b) one attack in LA3 (c, d) attack on any two or three LAs.

The total number of group comprising different combinations of DNS sub fragments is given by:

$$G_{(J,Q)STS} = \binom{Q}{J} \tag{4}$$

It is worth to note that Equation (4) can be simplified as:

$$G_{(J,J+1)STS} = \binom{J+1}{J} = J+1 \tag{5}$$

when only one redundant *LA* path is used, i.e., $Q = J + 1$.

3.3.2 Identification of DNS poisoned requests

To support the identification of poisoned DNS requests, it is then required to satisfy this condition where: Q > J + 1. Figure 6 demonstrates this using a (2, 4) STS scheme. Based on the example in Figure 6, we can observe the following:

The groups $(R1, R2)$, $(R1, R4)$ and $(R2, R4)$ reconstructed the same value of the DNS request packet. We realize that the reconstructed DNS request packet is truthful and thus represent the original DNS request *only and only if* the same value has been reconstructed by more than one

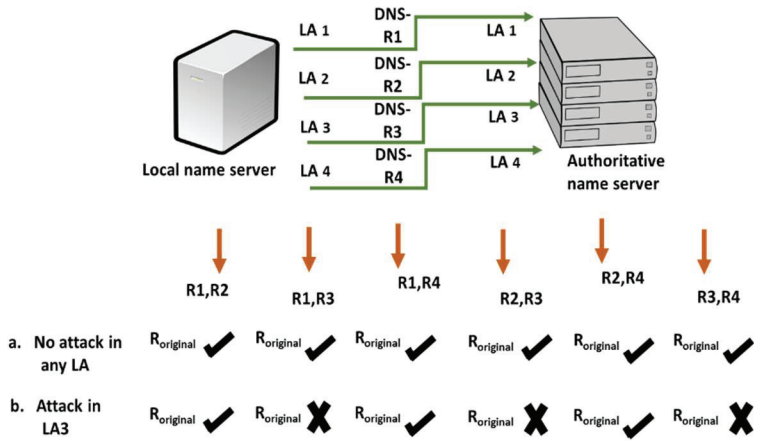

Figure 6 Identification of poisoned DNS requests using a (2,4) STS scheme.

group combination. The sub DNS request $R3$ is not part of any of the truthful values. The remaining groups $(R1, R3)$, $(R2, R3)$ and *(R3, R4)* produce different reconstructed values. Here, we can find out that $R3$ is the common part between the different reconstructed values. Consequently, it can be concluded that the DNS sub request packet of R3 arriving from LA3 is the *poisoned* DNS sub request packet.

The total number of groups that can produce the original DNS request G_{ORG} can be formulated according to the following equation:

$$\mathrm{G}{-}_{\mathrm{ORG}} = \binom{Q - dla}{J} = \frac{(Q - dla)!}{J!(Q - dla - J)!} \tag{6}$$

Where *dla* represents the number of defected or attacked LA paths, and *dla* $\leq$ (Q–J).

Equation (6) can be extended to calculate $P_{\mathrm{G-ORG}}$ which represents the probability of truthful reconstructed original DNS request assuming that *dla* LA paths have been attacked:

$$P_{\mathrm{G-ORG}} = \frac{\binom{Q-dla}{J}}{\binom{Q}{J}} = \frac{(Q - dla)!(Q - J)!}{Q!(Q - dla - J)!} \tag{7}$$

Equation (7) can be verified when applied to Figure 6. For Q = 4, J = 2, $dla = 1$, then we obtain $P_{\mathrm{G-ORG}} = 1/2$, i.e., 3 groups out of 6 are able to reconstruct the same truthful DNS request.

The identification method can provide protection against IP spoofing and therefore reduce the DoS attacks on DNS servers, i.e., the ANS servers. The protection against DoS attacks can be achieved by having the DNS request packet split, encoded and allocated to multiple LA paths, where each split part is considered as a new IP packet that consists of a new IP header. Consequently, the IP header of the original DNS request packet is also encoded. Therefore, the attacker cannot obtain useful information about the original DNS IP header and thus cannot obtain traces of DNS servers such as ANS servers. We propose to use the Multi-Protocol Label Switching (MPLS) technology as the underlying infrastructure to better utilize the network resources. Consequently, ANS servers can detect and identify corrupted DNS sub requests as we have explained in Figures 5 and 6.

3.3.3 Multipath approach impact on DNS reliability

We describe that an *LA* fails (i.e., defected) if at least one node along the *LA* path fails. The DNS request failure probability is defined as the probability of DNS request traffic sent by LNS server fails to reach the ANS server. In each path, it is either that the DNS request traffic succeeds or fails to perform the IP/domain name mapping. In order to be able to measure the failure probability of DNS requests, we need to compute the failure probability as follows:

$$P_{LNS-A}(n) = \prod_{LA_j}^{n} [1 - (1 - P_{LA_j})^i] \tag{8}$$

Where *i* represents the number of nodes on Lai. The Equation (8) holds when LA paths are node disjoint. Now, using the fact that for any *LA*j path the probability of failure is in the range of $0 \leq$ P (*LA*j) ≤ 1, then $P_{LNS-A}(n)$ keeps declining monotonically as *n* rises as shown in Figure 7.

3.3.4 Buffer requirements for variable length of LA paths

Usually, the *LA* paths between the LNS and ANS servers are not equal in length (i.e., due to variable number of nodes and links along each path). The DNS requests delay could be different because one *LA* path may be longer or slower than the other path. Consequently, for the ANS server be able to recognize and successfully process the DNS request, it should carefully calculate its needed buffer size (i.e., memory size) to store arriving sub DNS requests from multiple *LA* paths. Hence, the calculation of needed buffer is measured by the slowest *LA* path. To validate this point, consider an original DNS request DNS_f with *n* DNS sub request flows $DNS_{f1}, DNS_{f2} \ldots, DNS_{fn}$.

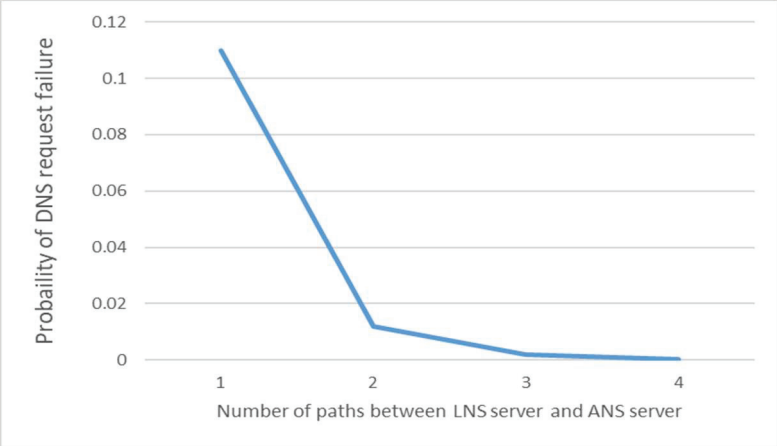

Figure 7 The impact of multi LA paths on the probability of DNS request failure.

The delay between the LNS server and the ANS server is calculated as:

$$Delay^{DNS_{fi}} = \sum_{i \in LAi} d_i \tag{9}$$

and the slowest LA path is the one with maximum delay value.

$$Delay^{DNS_{fslowest}} = Max\{Delay^{DNS_{fi}} \; for \; n \; LA \; paths\} \tag{10}$$

the size of buffer for each sub DNS flow is:

$$B^{DNS \; fLA} = (Delay^{DNS_{fslowest}} - Delay^{DNS_{fi}}) \cdot R_{DNSfi} \tag{11}$$

and thus the total buffer size required at the ANS server is:

$$B = \sum_{i \; each \; LAi} B^{DNS \; fLAi} \tag{12}$$

To illustrate this finding, the total buffer size needed in the example shown in Table 3 is equal to 10000 bits. This value is the summation of the two partial buffer values from the fastest paths (i.e., $LA1$ and $LA2$), and assuming the bit rate for all network links is equal to 2 Mbps. The results obtained can be compared with DNSSec [22, 23, and 24] and this is part of the future work of this research work.

Table 3 An example of buffer allocation needed at the ANS server

	$Delay^{DNS_{fi}}$	$Delay^{DNS_{fslowest}}$	Partial Buffer of LA (Bits)	Total Buffer Size (Bits)
DNS f1	32		(34-32). 2M = 4000	
DNS f2	31		(34-31). 2M = 6000	10000
DNS f3	34	34	0	

4 Conclusion and Future Work

In this paper, we examined the use of multipath routing to provide DNS security. The impact of multipath use has shown significant improvement when compared to the use of only single path between LNS and ANS servers for DNS request mapping. An algorithm to select the maximally disjoint LA paths have been introduced. Also, the buffering requirements for ANS servers have been formulated and discussed with examples. Our approach can be used to protect the DNS systems against cyberattacks such as DoS and DNS poisoning attacks.

However, the proposed work has limitations such as finding enough disjoint or maximally disjoint paths between DNS servers in a network. The overhead can be an issue if there is a large variation in delay between the multiple paths.

As a future work, we aim to continue this re-search work by comparing this method with other DNS security methods such as the DNSSec. Also, TCP flag sensing using artificial intelligence methods will also be investigated. Finally, we intend to use network simulation tools such as NS3 to validate our work.

References

[1] Y. Wanga, A. Zhoua, S. Liaoa, R. Zhengb, R. Huc, L. Zhang, "A comprehensive survey on DNS tunnel detection", Computer Networks, Vol. 197, 9 October 2021.

[2] Li Li; Liu Jiayong; Jia-Peng; Zheng-Rongfeng; "PSPAB:Privacy preserving average procurement bidding system with double spending checking" PloS One, (10) Vol. 15, 2020.

[3] IDC: Elevating Network Security with DNS, News Report Analysis, Journal of Network Security, Volume 2021, Issue 9, 2021, Page 4, ISSN 1353-4858.

[4] Liguo Chen; Yuedong Zhang; Qi Zhao, Guanggang Geng;ZhiWei Yan, "Detection of DNS DDoS Attacks with Random Forest Algorithm on Spark", Procedia Comp Sci, 2018, Volume 134, pp. 310–315.
[5] Angelo Furfaro, Pasquale Pace, Andrea Parise, Facing DDoS bandwidth flooding attacks, Simulation Modelling Practice and Theory, Vol. 98, 2020.
[6] Kupreev O., Badovskaya E.; Gutnikov A., "DDoS Attacks in Q1, Q2, Q3, Q4 2021", Tech. rep, Kaspersky (2021).
[7] Zheng J; Li Q; Gu G., Cao J.; Yau D, Wu J; "Realtime ddos defense using cots sdn switches via adaptive correlation analysis", IEEE Transaction. Information. Forensics Security, (7) 13 (2018), pp. 1838–1853.
[8] Verma S., Hamieh A, Huh J, Holm H; Rajagopalan S; Korczynski M, Fefferman N, "Stopping amplified DNS ddos attacks through distributed query rate sharing", Proceedings of the 11th International Conference on Availability, Reliability and Security ARES, 2016, pp. 69–78.
[9] Hao S, and Wang H, "Exploring domain name based features on the effectiveness of DNS caching", Computer Communication Review, (1) 47, 2017, pp. 36–42.
[10] Wu H, Dang X, Zhang L, Wang L, Kalman, "filter based DNS cache poisoning attack detection", Proceedings of the IEEE International Conference on Automation Science and Engineering CASE2015, 2015, pp. 1594–1600.
[11] Truong D., Cheng G. "Detecting domain-flux botnet based on DNS traffic features in managed network", Secur. Commun. Netw., (14) 9 (2016), pp. 2338–2347.
[12] Plohmann D, Yakdan K, Klatt M, Bader J, Gerhards-Padilla E, "A comprehensive measurement study of domain generating malware", Proceedings of the 25th USENIX Security Symposium, Austin, TX, 2016, pp. 263–278.
[13] Trevisan M, Drago I, Mellia M, Munafo M, "Automatic detection of DNS manipulations", Proceedings of the IEEE International Conference on Big Data, 2017, pp. 4010–4015.
[14] Pearce P, Jones B, Ensafi F, Feamster N, Weaver N, Paxson V, "Global measurement of DNS manipulation", Proceedings of the 26th USENIX Security Symposium, Vancouver, Canada, 2017, pp. 307–323.
[15] Kintis P, Miramirkhani N, Lever C, Chen Y, Gomez R, Pitropakis R, Nikiforakis N, Antonakakis M, "Hiding in plain sight: A longitudinal study of combosquatting abuse", Proceedings of the 24th ACM Conference on Comp and Comm Security (CCS), Dallas, USA 2017, pp. 569–586.

[16] Singh J., "Mitigating DoS and DDoS based Attacks: An Artificial Intelligence Approach", International Journal of Innovative Science and Research Technology, Vol. 5, Issue 5, 2020.
[17] Zhang Y., Cheng Y., "An Amplification DDoS Attack Defence Mechanism using Reinforcement Learning", IEEE SmartWorld, Leicester, UK, 2019, pp. 634–639.
[18] Lee S, and Shong C, "A K -Best Paths Algorithm for Highly Reliable Communication Networks", IEICE Trans. Commun., Vol. E82-B, No. 4., April 1999, pp. 586–590.
[19] Ridwan M., Radzi N., Wan W., Abdullah F., Jamaludin M., Zakaria M., "Recent trends in MPLS networks: Technologies, applications and challenges", 2020, IET Communications, 14 (2), pp. 177–185.
[20] Zhang Y, Fang Z, and Xu Z, "An optimal design of multiprotocol label switching networks achieving reliability requirements", Reliability Engineering and System Safety journal, 182, 2019, pp. 133–141.
[21] Shamir A., "How to Share a Secret", Communications of ACM, Vol. 22, Issue. 11, 1979.
[22] Z. Wang, H. Hu, G. Cheng, "Design and Implementation of an SDN-Enabled DNS Security Framework", Networks & Security, china comm., pp. 223–245, 2019.
[23] A. Niakanlahijia, S. Orlowskib, A. Vahidc, J. HaadiJafarianc, "Toward practical defense against traffic analysis attacks on encrypted DNS traffic", Computers & Security, Volume 124, 103001, January 2023.
[24] J. Quab, X. Maab. W. Liu, "Who is DNS serving for? A human-software perspective of modeling DNS services", Knowledge-Based Systems, Volume 263, 110279, 5 March 2023.

Biography

Sahel Alouneh is a full professor of electrical and computer engineering. Currently, he is the program director of the Cybersecurity program in the college of engineering, Al Ain university, Abu Dhabi campus, UAE. He is

currently on Sabbatical leave from the German Jordanian University. Prof. Alouneh obtained his B.Sc. in electrical and computer engineering from Jordan University of Science and Technology (JUST), Jordan in 2000. His M.Sc. and Ph.D were obtained from Concordia University, Canada in 2004 and 2008 respectively. His research interests include computer and communication networks, big data security, cloud computing, software security, MPLS security and recovery, Wireless networks security, Software testing, computer design and architecture.