Information Dynamics and Applications
https://www.acadlore.com/journals/IDA

The Need to Improve DNS Security Architecture: An Adaptive Security Approach

Daniel O. Alao, Folasade Y. Ayankoya, Oluwabukola F. Ajayi, Onome B. Ohwo*

Department of Computer Science, Babcock University, 121103 Ilishan-Remo, Nigeria

* Correspondence: Onome B. Ohwo (ohwoo@babcock.edu.ng)

Received: 01-06-2023 **Revised:** 02-20-2023 **Accepted:** 03-10-2023

Citation: D. O. Alao, F. Y. Ayankoya, O. F. Ajayi, and O. B. Ohwo, "The need to improve DNS security architecture: An adaptive security approach," *Inf. Dyn. Appl.*, vol. 2, no. 1, pp. 19-30, 2023. https://doi.org/10.56578/ida020103.

Abstract: The Domain Name System (DNS) is an essential component of the internet infrastructure. Due to its importance, securing DNS becomes a necessity for current and future networks. Various DNS security architecture have been developed in order to provide security services; such as DNS over HTTPS (DoH), DNS over TLS (DoT), and DNS over QUIC (DoQ). Unfortunately, these security architectures, especially DoT, are limited and are open to a number of performance issues. In this paper, we evaluate the present state of DNS security architecture, and we would see clearly that existing DNS security architectures are insufficient to secure DNS data transiting over the network; considering the growing cybersecurity landscape. On this note, we propose the need and adoption of a security architecture named Adaptive Security Architecture. Adaptive Security Architecture is devised to guard against identified threats, and anticipate unidentified threats in a manner similar to the immune-response system of human. Basically, mimicking nature's biodiversity as the fundamental means of effective attack responses. Finally, we conclude by an analysis to prove the need to improve DNS security architecture.

Keywords: Domain Name System; Adaptive security; DNS security; Security architecture; Cybersecurity

1. Introduction

The DNS protocol provides a detailed specification of the data structures and data communication exchanges. The domain name of a host is composed from the individual group names, consisting of strings separated by dots. For example, www.example.com. The highest competent authority is the root domain (Top Level Domains (TLD)). This consist of two kind, Generic Top-Level Domain (gTLD) (such as edu, com, net, and mil) and Country Code Top-Level Domain (ccTLD) (such as, .se, .us, .ca) [1]. The DNS also stipulates the technical functionality of the database service at its core, traditionally stored in a structured zone file. With the most common types of records being Start of Authority (SOA), IP addresses, domain name aliases, mail exchangers, name servers, pointers for reverse DNS lookups, and other types of data for either automatic lookups, or for human queries and so on. But as an all-purpose database, it is used for storing real-time blackhole list for battling unwanted email (spam). Consequently, a DNS is a critical application for the reliable and trustworthy operation of the Internet, assuming a fundamental role in its operations. Any disturbance in these operations can have a dramatic effect on the service provided, as well as, the global Internet. Over the years, there has been various attempts at breaching the DNS security to carry out numerous attacks against it [2]. Considering, the DNS recursive resolver lacks the necessary security mechanism for data confidentiality, availability and integrity. Hackers and attackers can take advantage of these vulnerabilities to falsify the DNS records and redirect genuine users to malicious domains. To circumvent these problems, DNS Security is needed to secure the communication channel between the clients and the DNS recursive resolver, allowing the verification of origin of the requests and that responses are not falsified. Over the past decade, there has been an increase in attacks on the Internet and private networks [3].

The drive of DNS security is to include security mechanisms, to enable legitimate clients verify the data's origin and be assured the data is the same from source to destination. Also, digital signatures are added to server responses, thus, protecting genuine users from malicious clients who provide falsified data. It is noteworthy that, Internet Protocol Security (IPsec) and Transaction SIGnature (TSIG) can be used to secure a DNS infrastructure. In the

last mile communications, IPsec can be used to secure communication between client and resolver, while TSIG can be used to secure Zone Transfer to enhance the security. Although these security mechanisms are not complete security solution for DNS recursive resolver, they still provide better security improvement and reduces the likelihood of attacks being carried out [1]. Consequently, in recent times, new protocols to enhance the privacy and confidentiality of the recursive resolver using encryption have been proposed. New protocols such as DNS over HTTPS (DoH), DNS over TLS (DoT), and DNS over QUIC (DoQ) [2]. Generally, DNS encryption is realized via encrypting the content of queries and responses (between clients and recursive resolvers) using available cryptographic techniques in an upper layer protocol. Encrypting DNS queries and responses between clients and resolvers holds the capacity to preserve user privacy against eavesdropping and man-in-the-middle (MITM) attackers. This research focuses on DNS over TLS (DoT), which uses Transport Layer Security (TLS) under the Transmission Control Protocol (TCP) transport layer to encrypt DNS queries and responses. Before being applied to the DNS applications, TLS has proven its efficacy in encrypting popular network applications such as email (SMTP), hypertext (HTTP), and voice-over-IP (VoIP) [3]. However, DNS over TLS has some performance issues [4-7] that requires attention before wide-spread adoption. Thus, ideologies and features of a new security architectural approach need to be considered such as adaptive security.

Adaptive security techniques have a semblance to the concept of risk management, which seeks to contain risk and meet the required Service Level Agreement (SLA). Adaptive security attempts to circumvent the effect and degree of possible threats in a timely fashion. Although still evolving, the implementation of an adaptive security approach can be done using technologies available today [8]. Other than upholding SLAs, the goals of adaptive security are to preserve integrity, encourage trustworthiness, and offer assurance. In the end, a security model seeks to introduce confidence in data and processing resources, ensuring trustworthiness, reliability, availability, and operation within satisfactory parameters. The primary differentiator from existing advanced practices, is that an Adaptive Security Architecture is devised to guard against identified threats, and anticipate unidentified threats in a manner similar to the immune-response system of human. Basically, mimicking nature's biodiversity as the fundamental means of effective attack responses. Hence, with the performance issues inherent in DNS over TLS and the need to ensure complete security service, there is need to improve the DNS security by deploying an adaptive solution to enhance the performance of DNS over TLS with an increasing client base and satisfies diverse usage patterns.

1.1 Objective of the Study

This research seeks to elucidate on the need to improve DNS security architecture. The objectives are to: provide an understanding of the DNS security protocols; carry out a literature review of existing DNS security techniques and identify inherent limitations. Based on these limitations, propose a security architecture, discuss on findings and conclude.

1.2 Significance of the Study

DNS is one of the most outsourced services. Many organizations lack the in-house DNS expertise, so let their domain registrar or another third party manage the organization's and only run recursive DNS services internally or outsourced to the ISP providing connectivity to the organization. With little or no control of the DNS infrastructure residing within the organization it is easy to see how DNS security can become an afterthought in security plans. But DNS security needs to be at the forefront of every discussion about network security.

1.3 Contribution to Knowledge

1. The development of an adaptive security approach designed for DNS to enhance its performance and provide complete security service, which may be a new and innovative approach in the field.
2. The integration of adaptive security techniques to predict and prevent both identified and unidentified threats, which may be a unique and valuable contribution to the existing state-of-the-art practices.
3. The exploration of the potential of adaptive security techniques to improve the reliability, availability, and operation of DNS under diverse usage patterns, which may offer new insights and solutions to the challenges faced by DNS security.

2. Literature Review

2.1 The Domain Name System (DNS) Protocol

This is a globally distributed hierarchical caching database with delegated authority dispersed across thousands of servers over the Internet. For some specific DNS domains, some servers are considered authoritative, but might

require information from other DNS server components to interpret a given Uniform Resource Locator (URL) to an IP address. This suggests a local name server may lack the necessary information to resolve a request, but knows where to locate it. Both TCP and User Datagram Protocol (UDP) are utilized by the DNS protocol over port 53. For zone transfer among primary and secondary DNS server, TCP is utilized. While for query and responses between DNS server and client, UDP is utilized. The root DNS servers constitute the top/root of the DNS hierarchy, with 13 global root name servers containing the database for all the top-level domains (TLDs) and country-specific DNS TLD such as .com, .org, .us and .uk [9] (See Figure 1).

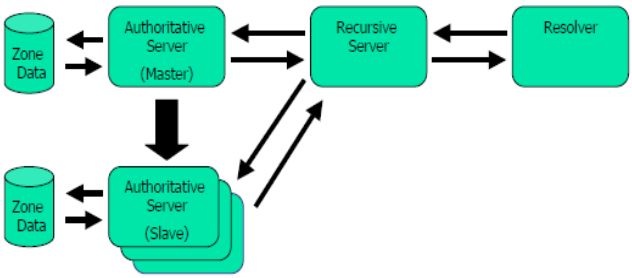

Figure 1. Domain Name System components **[**10**]**

2.1.1 Component of Domain Name System

(1) Root DNS Server: This sits at the top of the DNS hierarchy, that is the authoritative name server for the root zone [11]. There are 12 independent organizations who manages and maintains 13 root DNS servers named A to M: a.root-servers.net to m.root-servers.net. All A to M servers are copies of each other, containing the list of authoritative name servers for every TLD. For querying efficiency improvement, there are copies of several root server instances globally. Thus, the root name servers are the essential components in a DNS query resolution process [12].

(2) Authoritative DNS Server: The DNS services are operated by the Web hosting companies or DNS hosting companies; managing the authoritative DNS servers for a domain name which is queried through recursive resolvers. The authoritative DNS servers serve the final translation of the host name to the IP address, as it is the authoritative source. The authoritative name servers can be categorized into Master Authoritative Name Server, which keeps original copy of DNS records and database. While Slave Authoritative Name Server, keeps duplicate and contacts the master for refreshing its database [13].

(3) Recursive/Caching DNS Server: When a host name is entered into a browser, it checks its cache for the corresponding IP address of the host name. And if it does not have it, the stub resolver is contacted to check with the DNS server provided by Internet Service Provider (ISP) to find the corresponding IP address for the entered host name. This query is then redirected to the ISP's recursive DNS server, which checks its DNS record and cache data. And if it does not have the response, the DNS server is recursively queried until it gets the corresponding IP address for the given host name; and send the generated response back to the query originator. The recursive resolver will cache this response and will use it for replying the stub resolver for future queries [12].

(4) Resolver: Browsers and e-mail users require access to Internet resources using DNS resolver or stub resolver. It is denoted as a resolver for it has a resolving and a caching (name server) component. When a resource is requested for, the resolver formulates a name resolution query and sends it to a caching name server. Resolvers are largely configured with more than one resolving name servers, offering some level of effective fault tolerance. When a caching name server receives a query from the resolver, queries for sending them to authoritative name servers is formulated if unable to resolve it [14].

(5) Zone File: This comprises of information about various zone resources represented in a record called a resource record (RR). Therefore, several domains and their resources may be contained in a zone. Precisely, the RR contains the following major fields [14]: the domain or resource name (Owner name); time to Live (TTL)(seconds); Class; type of resource record (RRType); resource information depending upon the RRType (RData).

2.1.2 Domain Name System query types

There are 3 types of queries which can be raised to the DNS servers [12]:

(1) Iterative Query: If the name server is not authoritative for the requested data, other name servers are queried to find the response. Recursive queries are sent to the other name servers, thereby helping in finding the response and return it. Or iterative queries are sent and possibly referred to other name servers nearer to the domain name it's looking for. Present implementations do the latter, following the recommendations until a response is found.

(2) Recursive Query: A recursive query is sent by a resolver to a name server for information on a specific host name. The name server queried is then required to respond with the requested data or an error message stating that the requested data type do not exist or host name specified does not exist.

(3) Inverse Query: In normal DNS query, DNS user asks for the IP address for given host name but inverse or reverse queries works just exactly opposite to normal DNS queries. Here, the DNS user typically requests for resolution of IP address to host name and the DNS server have to fetch them through the Pointer (PTR) records in the DNS zone file.

2.1.3 Domain Name System query responses

The DNS servers can respond to queries in several ways, including [15]:

(1) Authoritative response: A DNS server that is authoritative for a domain name responds with the correct IP address of the requested domain name.

(2) Recursive response: If a DNS server receives a query for a domain name that it is not authoritative for, it can initiate a recursive query process to other DNS servers until it finds the authoritative server for the domain name.

(3) Referral response: If a DNS server is not authoritative for a domain name and does not have the answer in its cache, it can provide a referral response that includes the IP address of the authoritative server for the domain name.

(4) NXDOMAIN response: If the requested domain name does not exist, the DNS server can respond with an NXDOMAIN response, indicating that the domain name is not valid.

(5) SERVFAIL response: If a DNS server encounters an error while processing a query, it can respond with a SERVFAIL response, indicating that it was unable to process the query.

By understanding these responses, DNS users can diagnose and troubleshoot potential issues with their DNS queries.

2.1.4 Domain Name System: How it works

When the host name or Fully Qualified Domain Name (FQDN) is typed in, a sequence of events follows to resolve FQDN to its IP [16]: Search in hosts' file. For example (C:\Windows\System32\drivers\etc\hosts). If no response, a search in machine cache begins. If not found, a query is sent to the local DNS server, and a response is received back.

A lot goes on underneath unseen by the DNS users, as the DNS servers interconnect with each other to either recursively or iteratively resolve the user's DNS query. A DNS server response to queries in a number of ways: If the response is cached, the query is responded to from the cache. If the response is stored in a zone hosted by the DNS server, the query is responded to from its zone. A portion of the DNS tree stored on a DNS server is a zone. When a zone is hosted by a DNS server, it is authoritative for the names in that zone. If the DNS server cannot response to the query from its cache or zones, the response is queried from the parent servers.

2.1.5 Practical application cases of DNS

DNS has a wide range of practical applications, including:

(1) Website Hosting and Domain Names: One of the most common uses of DNS is to facilitate website hosting and domain names. Every website on the internet has a unique IP address, which is difficult for people to remember. Instead of using the IP address, website owners use a domain name, which is easier for people to remember. DNS translates the domain name into the corresponding IP address, allowing web browsers to connect to the website's server.

(2) Email Delivery: DNS also plays a critical role in email delivery. When an email is sent, DNS is used to determine the domain name of the recipient's mail server. Once the domain name is known, DNS is used again to translate the domain name into an IP address. This IP address is used to establish a connection to the recipient's mail server.

(3) Load Balancing: DNS can be used for load balancing across multiple servers. When a domain name is resolved, the DNS server can return multiple IP addresses, each corresponding to a different server. By rotating the order in which the IP addresses are returned, DNS can distribute traffic across multiple servers.

(4) Content Delivery Network: CDNs use DNS to distribute content across a network of servers around the world. When a user requests a piece of content, DNS is used to determine the closest server to the user. This server is used to deliver the content, reducing latency and improving performance.

(5) Security: DNS can also be used for security purposes. For example, DNS can be used to block access to malicious websites by returning an incorrect IP address. DNS can also be used for domain-based message authentication, reporting, and conformance (DMARC), which is used to prevent email spoofing.

2.1.6 Impact of DNS attacks

DNS attacks can have severe consequences for organizations in various industries. For example, in the finance

industry, DNS attacks can be used to redirect users to fake websites that mimic legitimate banking sites. This can enable attackers to steal login credentials, financial information, and other sensitive data.

In the healthcare industry, DNS attacks can disrupt access to critical systems and services, such as electronic health records and patient monitoring systems. This can compromise patient care and lead to serious medical errors.

In the government sector, DNS attacks can be used to compromise national security by redirecting users to malicious websites that install malware or steal sensitive information. This can enable attackers to gain access to classified information and disrupt critical government services.

One example of a DNS attack that had significant impact was the Dyn cyberattack [17] in 2016. This attack targeted DNS provider Dyn, which provides services to numerous high-profile clients, including Twitter, Spotify, and Reddit. The attack resulted in widespread service disruptions for these clients and other internet services, highlighting the critical role of DNS in internet infrastructure.

2.2 Domain Name System Security

Due to scalability issues, DNS protocol has been introduced without robust security features. Thus, people take advantage of this flaw to carry out malicious attacks on the DNS. To address the security issues, the following security concepts were developed:

(1) DNS over HTTPs (DoH): This is a protocol that enables DNS resolution over an encrypted HTTPS connection. Traditional DNS resolution is performed over a plain-text connection, which means that anyone with access to the connection can intercept and potentially manipulate the DNS queries and responses. This presents a security risk, as it allows attackers to redirect users to malicious websites or steal sensitive information. DNS over HTTPS addresses this security risk by encrypting DNS queries and responses using the HTTPS protocol.

(2) DNS over TLS (DoT): This is a security protocol that encrypts the communication between a user's device and a DNS resolver. Traditionally, DNS queries and responses are sent in plaintext, meaning that anyone who intercepts the communication can read the information being exchanged. This creates a potential security vulnerability, as attackers could potentially intercept and manipulate DNS queries to redirect users to malicious websites or intercept sensitive information. DoT aims to address this vulnerability by encrypting DNS queries and responses using Transport Layer Security (TLS), which is the same encryption protocol used to secure HTTPS connections.

(3) DNS over QUIC (DoQ): This is a protocol that allows for encrypted DNS queries and responses to be exchanged over a QUIC connection. QUIC is a transport protocol developed by Google that is designed to provide low-latency, high-performance connections over unreliable networks. The main benefits of using DNS over QUIC include increased privacy, improved performance, and reduced latency. By encrypting DNS queries and responses, DNS over QUIC helps prevent eavesdropping and tampering with DNS traffic, which can improve overall security.

2.2.1 Security threat impact level

There are three threat impact levels defined by the FIPS Publication 199, based on security goals such as confidentiality, integrity and availability and are as follows [18]:

(1) Low impact: This has a limited impact on the network's operations and resources. Degrading network's capability with noticeably reduced effectiveness. Resulting in negligeable damage to part or entire network resources, negligeable monetary loss and/or harm to individuals.

(2) Moderate impact: This has serious impact on network's operations and resources. Causing substantial dilapidation of network's capability with considerably reduced effectiveness. Resulting in substantial damage to network's resources, substantial monetary loss, and/or substantial but not life-threatening injury to individuals.

(3) High impact: This has severe impact on network's operations and resources. Causing severe dilapidation of network's capability including the failure to carry out essential functions. Resulting in major damage to the network's resources, major monetary loss, and/or severe and life-threatening injury.

2.2.2 Domain Name System service availability

DNS is relied on for the basic functions of a network. Therefore, proper implementation and organization of a DNS structure is essential to providing satisfactory DNS performance for users. If DNS services are unavailable, even the most secure networks can be ineffective. Threats to DNS service availability includes [19]:

(1) DNS capacity inadequacy: This occurs when few servers are deployed in addition to servers with insufficient processing or memory.

(2) DNS services unavailability: This occurs when a network is unreachability due to poor placement of DNS servers.

(3) DNS server failure: This may occur when the server is unreachable and the load on other DNS servers increases due to power failure, human error, hardware failure or natural disaster.

(4) Segmentation of servers by role failure, that is authoritative and recursive, can cause an overload of the servers and open them to multiple malicious attacks.

2.2.3 Domain Name System hardware and operating system attacks

Attackers can exploit the weaknesses within the operating system of the servers and applications, so as to compromise the server. These attacks take the following forms [19]:

(1) Hardware attack: Physically accessing to DNS servers allows the servers to be physically compromise by an attacker. This could reduce DNS service availability and possibly compromise configuration information.

(2) Operating system attacks: Attempts are made to compromise the server by social engineering or overloading the code execution buffer. Also, a known weakness of the server's operating system can be exploited by an attacker.

(3) DNS service attacks: Attempts can be made to compromise a DNS server service by exploiting a known vulnerability by an attacker.

(4) Control channel attack: The control channel in a DNS server offers a suitable technique for remotely controlling the DNS server. Such that an attacker can hack the control channel and execute malicious attacks such as the DNS service stopping or denying.

2.2.4 Attacks on Domain Name System

These are attacks that exploit the DNS infrastructure to create attacks on the DNS and other resources on the internet [12, 20]:

(1) Man-in-the-Middle Attack: DNS query responses are authenticated by the DNS server's IP address. So, an attacker can intercept and spoof the DNS server's IP address and make a response look genuine – as if it had originated from the intended DNS server. This is done by spoofing the source IP of the DNS servers and can become a bridge between the real DNS server and the client.

(2) Cache Poisoning Attack: This happens when an attacker successfully injects malicious data into the recursive servers of the DNS operated by multiple Internet Service Providers (ISPs). These DNS server types are closest to users from the perspective of a network topology, thus localizing the damage to specific users connected to those servers.

(3) Downloads: Distributed Denial of Service (DDoS) and Denial of Service (DoS) Attacks: DoS perpetrators typically attack high-profile web servers hosting websites or services such as banks, payment gateways, and also root name servers. This entails overloading the target server with external requests, so it cannot or slowly respond to legitimate traffic, rendering it essentially unavailable. These attacks are deployed by either forcefully resetting the targeted devices, or no longer providing intended service by consuming resources, or hindering communication between users.

(4) DNS Tunneling Attack: This enables attackers steal internal data in a network, by simply infecting a DNS user with a malware, which opens a tunnel through the DNS recursive server situated within the user's network. This circumvents the network's firewall and preserves the confidentiality of any security breach.

(5) DNS Hijacking Attacks: This attack involves corrupting the resolution of a DNS queries by malware that nullifications a device's TCP/IP configuration to redirect to a rogue DNS server controlled of an attacker. Or through altering the trusted DNS server behaviour to not comply with internet standards. These alterations may allow malicious attacks, or for selfish purposes. Such selfish purposes of ISPs involve redirecting web traffic to their own web servers where ads can be administered and statistics recorded. And by DNS service providers, to provide a form of censorship to designated domains [21].

(6) DNS Amplification Attack: This is one of the largest attacks, in terms of size measured in Gigabits per second (Gbps), enough to bring down even a large web host. This attack magnifies the amount of bandwidth they can target at a potential victim. For example, an attacker who controls a botnet capable of transmitting a 100Mbps of traffic. Although sufficient to bring down some websites, it is quite trivial compare to DDoS.

(7) Modified Data Attack: The authoritative DNS server is hosted on a number of machines. This server can be configured either by Command Line Interface (CLI) or Graphical User Interface (GUI). In a modified data attack, the attacker attacks the DNS server by obtaining root access to these machines using buffer overflow vulnerabilities.

(8) Spoofing Master Attack: In this attack, an attacker gets copies of critical zone files by masquerading as a slave DNS server to the master DNS server. The zone files can disclose a lot about the internal network topology, which can be used to attack specific nodes or a subnet.

(9) DNS ID Spoofing Attack: This is a type of MITM attack wherein DNS query sent by client contains unique identification number to identify queries and responses and tie them together. In a DNS configuration, the unique identification numbers are generated by pseudo random number generators (PRNG), and if few of the consecutive IDs can be predicted then the whole PRNG function can be predicted [21].

(10) NXDOMAIN Attack: This attack mainly targets the authoritative DNS server or recursive DNS server. The attacker sends the flood of queries to these DNS servers for resolving of non-existing domain name. DNS server cannot find the answer of the query and reply back with NXDOMAIN results, filling the recursive DNS server cache with NXDOMAIN results, which slows down the response time for the legitimate user. If high volume of queries is generated and the cache fills up very quickly, then the legitimate user feels high delay for their

responses.

2.3 Review of Related Literature

It is unknown how much information may be gleaned via traffic analysis on DoT communications, despite the fact that DoT is meant to stop on-path adversaries from observing and manipulating the victims' DNS requests and responses. A DoT fingerprinting technique was proposed by Houser et al. [22] to examine DoT traffic and identify whether a user has visited websites that are of interest to adversaries. When DNS messages are not padded, the suggested approach can detect DoT traffic for websites with a false negative rate of less than 17% and a false positive rate of less than 0.5%. Furthermore, we demonstrate that even when DoT messages are padded, information leakage is still feasible.

For five months at the start of 2021, this study tracked the adoption of DoH (DNS over HTTPS), DoT (DNS over TLS), and DoQ (DNS over QUIC) by three separate enterprises with worldwide reach. García et al. [5] analyzed the overall numbers, requests made per user, and traffic seasonality in order to determine the potential adoption trends. It was concluded that, despite increasing in 2020, there was statistically substantial evidence that the average volume of Internet traffic for DoH, DoT, and DoQ remained constant throughout the first five months of 2021. However, we discovered that the number of DoH servers that are available for use has increased by a factor of 4. These findings indicate that although the volume of encrypted DNS is not now increasing, there may soon be an increase in connections to unknown DoH servers for both good and bad intentions.

Although DNS over TLS (DoT) was established as an addition to the DNS protocol in 2016, little research has been done on how it performs. Research by Doan et al. [7] used 3.2k RIPE Atlas probes installed in home networks to quantify DoT from the edge and compare its adoption, dependability, and response times to DNS via UDP/53 (Do53). It was found that open resolvers are becoming more supportive of DoT. DoT is still only supported by regional resolvers. The reliability of DoT decreased while failure rates rose. Response times, according to DoT, are getting longer.

Using Transport Layer Security (TLS) to secure DNS communication has become popular recently. But at least two significant problems continue: (1) How can DNS-over-TLS endpoints be authenticated by clients in a scalable and extendable way? and (2) How can clients be confident that endpoints will act as expected? A revolutionary Private DNS-over-TLS (PDoT) architecture was proposed by Nakatsuka et al. [23]. A DNS Recursive Resolver (RecRes) that works in a Trusted Execution Environment (TEC) is part of PDoT. The study offered an open-source PDoT proof-of-concept implementation and empirically showed that its throughput and latency matched those of the well-known Unbound DNS-over-TLS resolver. The functionality that is available to code that runs within them is constrained, which presented the following major difficulties throughout the design process of PDoT. It also has a little quantity of memory. And applications must move to the non-TEE side if they need functionality that is not provided by the TEE.

Security has been largely handled by Transport Layer Security (TLS). However, the initial handshakes of vanilla TLS send information about the sort of service being accessed in plain-text, possibly disclosing user behavior and jeopardizing privacy. The "Encrypted ClientHello," or ECH, is a TLS 1.3 extension that Khandkar et al. [24] suggested to address the privacy concerns in TLS 1.3 by masking all of the information that may potentially disclose the service type. This study showed that the Encrypted Client Hellos (ECH) TLS 1.3 enhancement does not deliver the desired privacy. This is partly due to the fact that many services continue to use TLS 1.2, whilst ECH only supports TLS 1.3. The limited switchover to TLS 1.3 + ECH can fall short of protecting against malicious attacks that throttling/blocking particular internet services, as well as failing to fulfill the stated goals of privacy and anonymity.

The impact of Do53, DoT, and DoH on query response times and page load times was measured, in the study by Hounsel et al. [8], from five different worldwide perspectives. This study discovered that although while DoH and DoT response times are often higher than Do53, both protocols can outperform Do53 in terms of how quickly pages load. However, significant packet loss and latency are introduced when network conditions deteriorate.

Böttger et al. [9] examined the DNS-over-HTTPS environment in this study, paying particular attention to the cost of the added security. And to demonstrate the gains DoH offers over its predecessor, DoT, they examined various secure DNS protocols. It was then determined that head-of-line-blocking affects DoT and DoH/1. This difference in behavior may (at least partially) explain why DoH/2.0 gained traction more quickly than DoT.

According to research by Jonglez [10], DNS-over-TCP performance with few clients is comparable to DNS-over-UDP with only a 30% lag. Performance of DNS-over-TCP decreases as the number of clients rises and stabilizes at a 75% slowness. The performance profile for DoT is comparable to TCP, although there is a 30% to 45% speed impact. However, performance suffers noticeably as the number of clients rises for both TCP and TLS. This was thought to be a result of the kernel's need to manage a large number of TCP connections concurrently.

2.4 Limitations of Reviewed/Related Literature

It From the literatures reviewed, some of the proposed security model employed in Domain Name System

(DNS), for achieving secure communication are able to meet their security requirements but some still suffers from the following:

- The reliability of DoT decreased with increased failure rates.
- Increased response times, according to DoT, are getting longer.
- It does not support real-time, automated monitor.
- Head-of-line blocking.
- Cryptographic algorithm processing overhead.
- Increases the bandwidth consumption on both the client and the server network.
- Privacy issues.

3. Methodology

The Results section may be divided into subsections. It should describe the results concisely and precisely, provide their interpretation, and draw possible conclusions from the results.

The management of attacks against a network is the core practice of network security. These attacks can the external (outside the network) and internal (inside the network). The development of security strategy requires the careful considerations and understanding of your security needs. The National Institute of Standards and Technologies (NIST) Cybersecurity Framework [25], will be applied as a conceptual framework within which the security strategies would be presented.

While this framework might be a de facto standard globally, it is not a "one size fits all" framework. Rather, it provides [19]:

1. a taxonomy and methodology for characterizing the current and desired strategy,
2. ways to enable improvement of current strategy toward desired strategy,
3. communication among stakeholders about cybersecurity threats,
4. guidance to execute risk evaluations and
5. to manage risk despite vulnerabilities, threats, and risk tolerance.

Other pre-existing security standards consulted including COBIT 5 [26], ISA 62443 [27], ISO/IEC 27000 [28], NIST SP 800-53 Rev4 [29]. Referencing specific sections of these standards, the framework fundamentally represents the common principles amongst these aforementioned security standards to outline a strategy for cybersecurity.

Security threats are counter-productive to the functionality, performance, availability, and integrity of Information Technology systems. The goal is to decrease possible security threats to the level wherein Service Level Agreements (SLAs) can still be satisfied; as well as the risk management concept. By timely prevention of an attack, adaptive security attempts to decrease the effect and degree of possible threats.

This research adopts a comprehensive adaptive system architecture proposed by Lamprecht [30]. A comprehensive design of the Adaptation Unit, is needed to apply the principles and deployment of an Adaptive Security Service. The security service in the design is various cryptographic algorithm. The following design is presented:

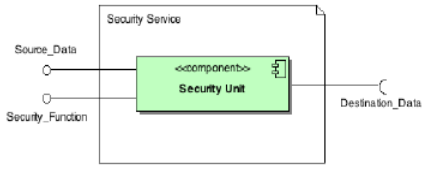

Figure 2. Security service [30]

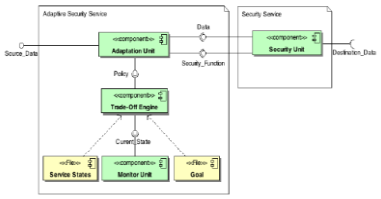

Figure 3. Adaptive Security Service [30]

Figure 2 represents a typical security service. Plaintext is inputted as source data into the Security Unit, the relevant security function is applied to the data and ciphertext is outputted as destination data. Cryptographic algorithm is employed in the data encryption process.

Figure 3 represents the Adaptive Security Service components that makes up a typical security service with adaptive features.

(1) The Adaptation Unit is a key component who is accountable for imposing the security adaptation. It captures the source data as it transfers to the Security Unit, creating an adaptive security control point. It aligns the given Policy rule to the source data telling the Security Unit which security function to apply to which datasets.

(2) The Monitor Unit permits the Adaptive Security Service to monitor and give an account of the cross-cutting concerns in the service environment. Through runtime monitoring, the Trade-Off Engine is able to make up-to-date decisions based on the present state of the system.

(3) The Trade-Off Engine component operates as the decision point on behalf of the Analyze component. The Service States file encompasses of pre-compiled data which denotes the inter-relationship between security and the environmental cross-cutting concerns.

(4) The Plan component regulates whether or not, as well as, how security should be adapted based on the adaptation goal. Based to the Service States, the Trade-Off Engine creates a Policy which gratifies the adaptation goal.

3.1 Adaptation Unit Design

This is involved in the adaptation process in two key areas. First of all, it captures data sent to the Security Service, thus leveraging control over which security function to apply to the data in advance. Secondly, a given set of policy rules aids in directing the Security Service on which security functions are appropriate for which dataset (See Figure 4 and Figure 5).

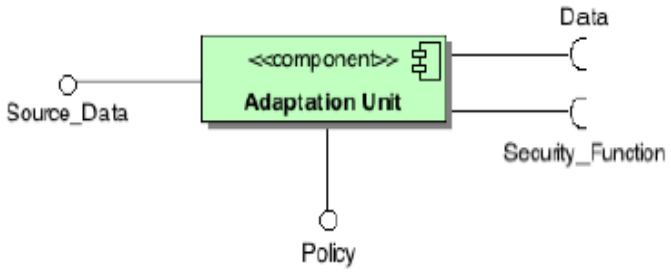

Figure 4. Adaptation Unit [30]

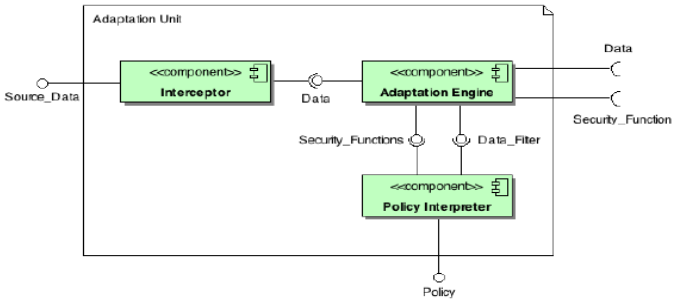

Figure 5. The Generic Adaptation Unit [30]

(1) The Interceptor redirects data meant for the Security Unit to the Adaptation Engine. Data may be evaluated and reported in a format stated by the Adaptation Engine interface.

(2) The Policy Interpreter evaluates the arriving policy updates, creating a properly organized security function and data filter pairs as the Adaptation Engine input. Such evaluation may include authentication, policy syntax checking and aligning policy rules to internal security algorithm and data filter representations.

(3) The Adaptation Engine applies the conditions stated in the Data Filter to the arriving data from the Interceptor. Once the data filtered, it is transmitted to the Security Unit stating the suitable security function applicable to a particular data subset.

4. Discussion

Based on these limitations and the changing cybersecurity landscape, we propose an Adaptive Security

Architecture as a better alternative to existing techniques identified from literature. As Darwin tells us, organisms must adapt or die. To extend the parallel between biological ecosystems and IT ecosystems, successful IT infrastructures must adapt or they will eventually fail due to predator attacks, viral infections, or the inability to adjust to environmental changes. Also, part of the human immune system includes immune response mediators (for example, T-cells). The role that such mediators play in the IT infrastructure is almost identical to how they are used in the biological sense — they are guardian agents or sentinels that are deployed throughout the infrastructure that act as sensors to identify potential threats. In conjunction with threat response and feedback mechanisms, these sentinels moderate the immune response as they do in a biological system.

The application of some adaptive ideologies is a "defense-in-depth" security approach that incorporates numerous diverse strategies. This is achieved by utilizing techniques such as clustering, hardware redundancy, or several types of firewall applications. In so doing, when or if a particular threat causes one component to fails, there is a likelihood other component will not fail; therefore, ensuring the survivability and availability of the system. Similarly, flexibility and elasticity can be maintained through virtualization techniques. The utilization of virtualization technologies can categorize diverse system services into secure execution containers to isolate service instances. Such that an affected service instance has no impact on the execution of another service instance; thus, ensuring continuity of services. Consequently, the affected container can be isolated to localize the attacks impact by the response mechanisms. The principal distinguishing factor of an adaptive security procedure includes protection not only against known attacks, but also anticipation of unknown attacks, similar to the human immune response system.

When outlining the possible security approaches, it is noteworthy that these approaches must align the structure of the general security architecture. Thus, ensuring all components and design entities accomplish the general security policy. The following steps should be considered when developing an Adaptive Security Architecture:

1. Define attacks and their characteristics that are required to circumvent or terminate.
2. Identify acceptable and trusted behaviours, components, and actions that must not be mistaken for an attack.
3. Define triggers to invoke an auto-resistant system response for attack monitoring.
4. Implement critical functions redundancy, such that if or when compromised, the entire system will not fail.
5. Define an attentive attack response mechanism that do not result in eliminating the host.
6. Define a robust system recovery process proficient in adaptively reconfiguring and rebooting.
7. Define a feedback process that allow the attack response mechanisms to confirm attacks so as to only respond to genuine and accurate attacks.

The Adaptive Security Architecture proposed is designed to address the impact of different levels of security threats mentioned in section 2.2.1 accordingly. The architecture aims to protect against both identified and unidentified threats by continuously monitoring the network and adapting the security measures in response to the level of threat detected. The architecture also includes different security layers and techniques that can be applied based on the severity of the threat, such as intrusion detection and prevention systems, firewalls, and access control mechanisms. Therefore, the architecture can help mitigate the impact of security threats at different levels and ensure the security of the network and its resources.

The Adaptive Security Architecture proposed by the authors can address the different types of attacks mentioned in sections 2.2.3 and 2.2.4. The architecture is designed to provide defense in depth, which includes multiple layers of security controls, such as firewalls, intrusion detection systems, and security monitoring. These security controls can help prevent, detect, and respond to attacks on the DNS infrastructure. For example, firewalls can be used to restrict access to DNS servers, intrusion detection systems can be used to detect attacks on the DNS infrastructure, and security monitoring can be used to monitor the DNS infrastructure for suspicious activity. Additionally, the architecture includes incident response planning and testing, which can help organizations respond quickly and effectively to attacks on the DNS infrastructure.

The Adaptive Security Service components presented in Figure 3 can correspond to several of the problems mentioned in section 2.4:

1. The Monitor Unit can address the need for real-time, automated monitoring, which is a limitation of some existing security models in DNS.

2. The Trade-Off Engine component can help mitigate the overhead caused by cryptographic algorithm processing, which is a concern for some security models.

3. The Adaptation Unit, with its ability to capture data and direct security functions based on policy rules, could potentially address issues with head-of-line blocking and increase the efficiency of data transmission.

It's important to note, however, that the Adaptive Security Service is presented as a comprehensive system architecture rather than a specific solution to the limitations of existing security models. While it may offer potential solutions to some of the problems mentioned in the previous section, it would require careful implementation and testing to determine its effectiveness in practice.

5. Conclusions

This research has been able to establish that DNS security provides an additional layer of protection between a

user and the internet by circumventing known and unknown malicious attacks. Also, a precedent was set by reviewing literature to show the trends in DNS security technique. Subsequently, consider the changing cybersecurity landscape and limitations identified from literature review, this research was able to support the need to improve the present state of DNS security architecture. On this note, principles and characteristics of a new security architectural approach need to be considered such as adaptive security. Adaptive Security Architecture employs the elementary concept of "self" and "non-self", which allows the ability to comprehend and identify what a typical behavior is, what it's not, and determine if it's a possible attack. Hence the essential method of adaptive security includes attack detection, analysis and response. This should be automatic to allow fast implementation of security patches and updates. By using secure DNS servers both at home and at work, users can avoid unnecessary risks and the potential for malicious attack. Further studies are needed to explore specific solutions for the contemporary DNS problems discussed in this paper. Specifically, future research can focus on developing efficient and secure DNS protocols that address the limitations identified in this study, such as the reliability and response time of DoT, real-time automated monitoring, and cryptographic algorithm processing overhead. Such solutions would greatly enhance the security and performance of DNS, which is critical for ensuring the secure communication of online services.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] L. Dostálek and A. Kabelová, "DNS in action: A detailed and practical guide to DNS implementation, configuration, and administration," Packt Publishing Ltd, Birmingham: 2006.

[2] B. B. Gupta, "Computer and cyber security: Principles, algorithm, applications, and perspectives," CRC Press, Taylor & Francis, 2018.

[3] I. Khan, W. Farrelly, and K. Curran, "A demonstration of practical DNS attacks and their mitigation using DNSSEC," *Int. J. Wirel. Networks and Broadb. Technol.*, vol. 9, no. 1, pp. 56-78, 2020. https://doi.org/10.4018/ijwnbt.2020010104.

[4] S. Rostampour, "Deploying DNS Security Extensions," Master Thesis, Chalmers University of Technology, Sweden, 2012. https://publications.lib.chalmers.se/records/fulltext/173693/173693.pdf.

[5] S. García, K. Hynek, D. Vekshin, T. Čejka, and A. Wasicek, "Large scale measurement on the adoption of encrypted DNS," 2021. https://doi.org/10.48550/arXiv.2107.04436.

[6] M. Lyu, H. H. Gharakheili, and V. Sivaraman, "A survey on DNS encryption: Current development, malware misuse, and inference techniques," *ACM Comput. Surveys*, vol. 55, no. 8, pp. 1-28, 2022. https://doi.org/10.1145/3547331.

[7] T. V. Doan, I. Tsareva, and V. Bajpai, "Measuring DNS over TLS from the edge: Adoption, reliability, and response times," In Passive and Active Measurement: 22nd International Conference, vol. 12671, 2021, Switzerland: Springer, Cham. https://doi.org/10.1007/978-3-030-72582-2_12.

[8] A. Hounsel, K. Borgolte, P. Schmitt, J. Holland, and N. Feamster, "Comparing the effects of DNS, DoT, and DoH on web performance," In Proceedings of The Web Conference 2020, 2019. https://ui.adsabs.harvard.edu/link_gateway/2019arXiv190708089H/doi:10.48550/arXiv.1907.08089.

[9] T. Böttger, F. Cuadrado, G. Antichi, E. L. Fernandes, G. Tyson, I. Castro, and S. Uhlig, "An empirical study of the cost of DNS-over-HTTPS," In Proceedings of the Internet Measurement Conference, Amsterdam, Netherlands, 2019. https://doi.org/10.1145/3355369.3355575.

[10] B. Jonglez, "End-to-end mechanisms to improve latency in communication networks," *Networking Internet Architecture*, pp. 1-137, 2021.

[11] W. Joel, "Designing an Adaptive Security Architecture," *Sun BluePrints™ Online*, pp. 1-19, 2008. https://fliphtml5.com/flqg/rlsk/basic.

[12] B. Rajendran and P. Shetty, "Domain Name System (DNS) security: Attacks identification and protection methods," In Proceedings of the International Conference on Security and Management, Las Vegas, USA, 2018.

[13] S. Ariyapperuma, C. J. Mitchell, "Security vulnerabilities in DNS and DNSSEC," In the Second International Conference on Availability, Reliability and Security, Vienna, Austria, 2007. https://doi.org/10.1109/ARES.2007.139.

[14] M. Müller, G. C. Moura, R. D. O. Schmidt, and J. Heidemann, "Recursives in the wild: Engineering

authoritative DNS servers (extended)," *SIDN Labs*, vol. 21, 2017. https://ant.isi.edu/~johnh/PAPERS/Mueller17a.pdf.
[15] R. Chandramouli and S. Rose, "Secure Domain Name System (DNS) deployment guide," *NIST Spec. Public.,* vol. 800, 2006. https://doi.org/10.6028/NIST.SP.800-81-2.
[16] K. Young, "Cyber case study: The Mirai DDoS attack on dyn," *Coverlink Insur.*, 2022. https://coverlink.com/case-study/mirai-ddos-attack-on-dyn.
[17] F. I. P. S. Pub, "Standards for security categorization of federal information and information systems," 2004. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=86faf7acfa528acb3435285dbd09bb3f1b7bbe38.
[18] M. Dooley and T. Rooney, *DNS Sec. Management*, USA: John Wiley & Sons., 2017.
[19] A. A. Z. Hudaib and E. A. Z. Hudaib, "DNS advanced attacks and analysis," *Int. J. Comput. Science and Sec.*, vol. 8, no. 2, 2014. https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=ed8a6c7609e44640365b66e3ded35ea847591fec.
[20] V. Paxson, "An analysis of using reflectors for distributed denial-of-service attacks," *ACM SIGCOMM Comput. Commun. Review*, vol. 31, no. 3, pp. 38-47, 2001. https://doi.org/10.1145/505659.505664.
[21] M. H. Jalalzi, W. B. Shahid, and M. M. W. Iqbal, "DNS security challenges and best practices to deploy secure DNS with digital signatures," In 2015 12th International Bhurban Conference on Applied Sciences and Technology, Islamabad, Pakistan, 2015. https://doi.org/10.1109/IBCAST.2015.7058517.
[22] R. Houser, Z. Li, C. Cotton, and H. Wang, "An investigation on information leakage of DNS over TLS," In Proceedings of the 15th International Conference on Emerging Networking Experiments and Technologies, Orlando, FL, USA., 2019. https://doi.org/10.1145/3359989.3365429.
[23] Y. Nakatsuka, A. Paverd, and G. Tsudik, "PDoT: private DNS-over-TLS with TEE support," *Digital Threats: Res. and Practice*, vol. 2, no. 1, pp. 1-22, 2021. https://doi.org/10.1145/3431171.
[24] V. S. Khandkar, M. K. Hanawal, and S. G. Kulkarni, "Challenges in adapting ECH in TLS for privacy enhancement over the Internet," 2022. https://doi.org/10.48550/arXiv.2207.01841.
[25] C. I. Cybersecurity, "Framework for improving critical infrastructure cybersecurity," 2018. https://www.baltimorecityschools.org/sites/default/files/inline-files/NIST.CSWP_.04162018.pdf.
[26] M. Garsoux, "ISACA COBIT 5 ISACA's new framework for IT Governance, Risk, Security and Auditing An overview". https://www.academia.edu/42165806/COBIT_5_ISACA_COBIT_5_ISACAs_new_framework_for_IT_Governance_Risk_Security_and_Auditing_An_overview_M_Garsoux_COBIT_5_Licensed_Training_Provider_COBIT_5_ISACA.
[27] ANSI/ISA, "Security for industrial automation and control systems: System security requirements and security levels," 2013. https://securityboulevard.com/2020/09/everything-you-need-to-know-about-nist-cybersecurity-frameworks-informative-references/.
[28] S. N. V. Schweizerische, "Information technology - security techniques - information security management systems - requirements," 2013. https://eldritchdata.neocities.org/PDF/CS/SecManagmentSystemsReq.pdf.
[29] J. T. Force and T. Initiative, "Security and privacy controls for federal information systems and organizations," *NIST Spec. Public.*, vol. 800, no. 53, pp. 8-13, 2014. http://dx.doi.org/10.6028/NIST.SP.800-53r4.
[30] C. J. Lamprecht, "Adaptive security," Doctoral Dissertation, Newcastle University, UK, 2012. http://hdl.handle.net/10443/1435.