applied sciences

MDPI

Article

DoH Tunneling Detection System For Enterprise Network Using Deep Learning Technique

Tuan Anh Nguyen [1] and Minho Park [2,*]

[1] Department of Information Communication, Materials and Chemistry Convergence Technology, Soongsil University, Seoul 06978, Korea; anhtuan@soongsil.ac.kr
[2] School of Electronic Engineering, Soongsil University, Seoul 06978, Korea
* Correspondence: mhp@ssu.ac.kr

Abstract: In spite of protection mechanisms for Domain Name System (DNS), such as IP blacklist and DNS Firewall, DNS still has privacy issues in reality, since DNS is a plain-text protocol. Recently, to resolve this problem, an encrypted DNS, called DNS-over-HTTPS (DoH), has been developed, and is becoming more widespread. As the secured version of DNS, DoH guarantees privacy and security to prevent various attacks such as eavesdropping and manipulating DNS data by using the HTTPS protocol to encrypt the data between the DoH client and the DoH-based DNS resolver. DoH is one of the best security options for an enterprise network where more sensitive data protection is required. However, DoH may cause an unintended security breach, i.e., information leakage via malicious DoH tunneling. Since the DoH traffic is encrypted and indistinguishable from other HTTPS traffic, data hidden inside DoH packets can be easily leaked out of an enterprise network. Although some countermeasures to detect DoH tunneling attacks have been proposed, they still have limitations. Previous research used Supervised Machine Learning methods to detect DoH tunneling, which required a high volume of labeled data. In practice, collecting and labeling all of the data is an impossible task, especially in DoH, when all of the data are encrypted. Furthermore, Supervised Machine Learning methods rely heavily on human-engineered feature extraction, which makes classifying encrypted DoH traffic difficult. Furthermore, no previous research has mentioned a complete functional DoH detection applied to network infrastructure. Therefore, we propose a detection system for DoH tunneling attacks based on Transformer to detect a malicious DoH tunneling and build a fully functional DoH detection system that can be integrated with the security operation system of an enterprise network. The experiment results show a significant improvement compared with previous works.

Keywords: DNS-over-HTTPS; malicious DoH; semi-supervised learning

Citation: Nguyen, T.A.; Park, M. DoH Tunneling Detection System For Enterprise Network Using Deep Learning Technique. *Appl. Sci.* **2022**, *12*, 2416. https://doi.org/10.3390/app12052416

Academic Editors: Donato Impedovo and Giuseppe PIRLO

Received: 30 December 2021
Accepted: 22 February 2022
Published: 25 February 2022

1. Introduction

Domain Name System (DNS) is a hierarchical and decentralized naming system commonly used to map domain names to IP addresses. Since DNS is a public and plain-text protocol, it is vulnerable to various DNS attacks. In the past two decades, there have been various DNS attacks and countermeasures to protect DNS. Authentication was used for DNS spoofing attack [1,2]; a DNS amplification attack was prevented by response rate limiting or DNS load balancing [3]; and analyzing the DNS packets protected DNS from DNS tunneling [4]. In addition, various DNS protection technologies have been introduced, such as OpenDNS by Cisco [5], DNSFilter [6], and Infoblox BloxOne Threat Defense [7].

Despite these security solutions, DNS still has privacy issues in reality since DNS is a plain-text protocol. This issue should not be ignored, especially in an enterprise network where more sensitive data protection is required. In an enterprise network, many activities, such as web, email, and file exchange over the Internet or a local network, are heavily relative to the DNS [8]. In practice, an enterprise network can generate millions

of DNS queries and responses a day. The DNS traffic includes crucial data such as IP addresses, destination, source port number, and transaction ID, which can be exploited by attackers [9–11]. Furthermore, attackers can obtain sensitive information to create user profile and exploit it to launch attacks or even for the calculation of a user's geographical area [12].

To resolve this security issue, the Internet Engineering Task Force (IETF) developed DNS-over-HTTPS in 2018 [13]. In DoH, a DNS query is wrapped in the regular HTTPS traffic and forwarded to a DoH-capable DNS server called a DoH resolver via port 443. The query is resolved within a DoH request, and the response is also encrypted. DoH protects users' privacy while speeding up DNS query processing. Since DoH traffic is encrypted, even the internet service provider cannot examine the contents of the DNS packet. It can also protect DNS against phishing attack, eavesdropping, and DNS data modification by Man-in-the-Middle attack. DoH is now available as an extension service in various browsers, including Firefox and Chrome. The definition and implementation of DoH is explained clearly in RFC 8484 [13] and RFC 1035 [14].

Although DoH can prevent many security vulnerabilities, on the other hand, it may bring about a severe security problem. Because DoH APIs are public, attackers can easily use them to develop a more powerful DNS tunneling attack, known as DoH tunneling. DoH tunneling [15] is an upgrade of DNS via HTTPS, which is harder to detect. The traditional DNS [9] uses port 53 for communication in the DNS resolution process. Therefore, the existing security solutions such as OpenDNS by Cisco [5], DNSFilter [6], and Infoblox BloxOne Threat Defense [7] analyze the DNS packets on this port. In DoH, however, all the DNS queries and responses are exchanged on port 443 [13], which is the same port that other HTTPS applications use. Consequently, the existing DNS security techniques cannot capture the DoH packet separately. Moreover, since the DoH traffic is encrypted, the clues used to detect attacks such as packet header, packet number, packet length, packet direction, and arrival interval between packets are not available, which makes detection difficult [15]. Furthermore, DNS blocking solutions such as IP Blacklist cannot be executed because the information about IP address is encrypted in DoH.

The previous research on DoH tunneling detection has employed Supervised Machine Learning approaches such as Random Forest, Decision Tree, Support Vector Machine, and Gradient Boosting to distinguish DoH tunneling traffic from conventional HTTP traffic based on their statistical features [15–17]. Although they achieved high accuracy with their dataset, there remain some limitations. Firstly, these techniques require labeled data to achieve high precision. However, due to the massive number of websites on the Internet, labeling all of the traffic data is an impossible task. Therefore, Supervised Machine Learning algorithms are not a good choice [18]. Secondly, the performance of Machine Learning techniques relies heavily on human-engineered feature extraction, which makes the classification of encrypted DoH traffic difficult. On the other hand, in Deep Learning, feature extraction is automatically selected through the training process. This characteristic makes Deep Learning a highly suitable method for DoH traffic classification, especially when we have no prior knowledge about the weight of the feature in DoH statistical features [19].

To overcome the limitations of the existing proposals, we propose a DoH tunneling attack detection system based on Transformer, a Deep Learning model that adopts the mechanism of self-attention [20]. It includes an HTTPS traffic collection and a DoH attack detection module with a two-layer classifier to detect malicious DoH tunneling traffic. Both layers in the malicious DoH detection module use the Transformer [20] architecture as a classifier. The first layer distinguishes DoH traffic from normal HTTPS traffic and the second layer classifies malicious DoH traffic from DoH traffic. The Transformer is a well-known architecture in natural language-processing jobs that offers numerous advantages over previous time-series data classification approaches. The Transformer can handle sequence data in parallel, allowing it to process more data in the same time period. The self-attention module in Transformer helps it to autofocus on essential features in sequence.

These crucial characteristics make Transformer suitable for detecting DoH malicious traffic in HTTPS traffic.

The proposed system is a lightweight, intelligent, and efficient DoH tunneling detection solution for an enterprise network to detect malicious DoH traffic among HTTPS traffic successfully. Our proposed system provides a distributed collection system for the Security Operation Center (SOC) to capture HTTPS traffic at any machine in the network and analyze them to detect malicious DoH traffic in real time. The detection module with the Transformer model has many advantages compared with existing proposals. It requires a much smaller number of labeled data. Although it is trained with only 25% of the labeled data, it has better accuracy, up to 99%, than previous techniques with the same number of labeled data.

Our contributions in this study are as follows:

- We propose a two-layered detection approach based on Transformer architecture as an effective DoH tunneling detection model. The first layer separates DoH and HTTPS traffic. The second layer separates legitimate DoH traffic from malicious DoH traffic.
- We implement a fully functional DoH tunneling detection solution that can be integrated into an enterprise network's security operation system. The system is an end-to-end detection system that gathers HTTPS traffic traces, stores and analyzes them to detect malicious DoH traffic, and alerts the network manager.
- Finally, we conduct comprehensive experiments to evaluate the proposed scheme. The results show that our proposed method achieves significant improvement compared with other research.

The rest of the paper is organized as follows. Section 2 reviews some related work. In Section 3, background knowledge is described. Section 4 presents the model development and system architecture. Finally, the performance evaluation is shown in Section 5, and the paper is summarized with the conclusion in Section 6.

2. Related Works

2.1. DNS Attacks and Countermeasures Methods

Ref. [1] explained how DNS amplification attacks work and how difficult it is to detect them in the computer network. Since DNS amplification attacks are a kind of Distributed Denial of Service (DDoS) attack, DNS amplification attacks can be detected through other DDoS attack-detection techniques. Some solutions were proposed in [21] to mitigate and prevent DNS amplification attacks, such as source IP verification and response rate limiting.

R. Houser [2] proposed a DNS spoofing attack detection mechanism using a Support Vector Machine classifier to detect the attack in the LAN gateway. Their results show an accuracy of 89% in the early detection of a multi-day attack. Maksutov [22] detected DNS spoofing by analyzing the DNS packet, and DNSSEC [23] added an authentic fingerprint to prevent the DNS resolution process from spoofing.

Researchers in [24] applied a Machine Learning classifier using K-Nearest Neighbor and Support Vector Machine to classify tunneling traffic from non-tunneling traffic through statistical fingerprints of protocol messages. Their results show high accuracy, up to 99% in short detection time. According to [15], IP Blacklist is an effective tool to prevent DNS queries to malicious domains.

2.2. DoH Tunneling Detection Methods

Most previous researchers focused on the applicability and performance aspects of DoH [25–27]; only a few papers about DoH security have been published so far. Two of the most important objectives studied in DoH security are privacy and data exfiltration.

Refs. [28,29] studied the privacy of DoH protocol, and the results show that it is technically possible to defer private information about one's web browsing activity by capturing and analyzing their DoH traffic. Refs. [30,31] also studied DoH privacy, and the authors also agreed that despite the advantage in privacy, DoH could be a double-edged sword.

Since DoH is a very recently developed technology, there is little research detecting DoH tunnels. In [29], the authors surveyed the effectiveness of traffic analysis attacks on DoH traffic. They planned a new set of features and showed that the proposed DoH traffic analysis effectively identifies malicious websites. However, their work shows that training a model in a different environment to the one deployed causes a negative impact on the performance. Ref. [30] proposed a novel traffic analysis method that combines size and timing information to classify the encryption DNS traffic. A new encrypted DNS traffic detection was proposed in [32]. In this work, the captured traffic was analyzed based on temporal features and packet sizes to detect DoH traffic. SANS Institute published a white paper [33], clearly explaining DoH detection by applying the Real Intelligence Threat Analysis (RITA) framework.

The best-known example of malicious DoH attacks is the Godlua Backdoor [34]. In this attack, the attacker used DoH as a command-and-control (C2) communication channel between the attack command servers and malware or botnet in the victim network. Another cruel method of attackers is using DoH to trigger a redirected web page as a part of a spam campaign.

The author in [15–17] presented a two-layered binary classification approach, where the first layer distinguishes DoH traffic from non-DoH traffic and the second layer detects the malicious DoH. In their works, some traditional machine learning algorithms, such as Random Forest, Gradient Boosting, and K-Nearest Neighbors were applied for classification tasks. Eventually, these algorithms showed some limitations in term of accuracy and execution time, and required a large labeled dataset. Table 1 illustrates the previous research methods and their drawbacks.

Table 1. The drawbacks of previous DoH detection techniques.

Author	Method	Drawback
M. Montazeri Shatoori [15]	Supervised Machine Learning technique	Require a huge number of labeled data
Mitsuhashi R. [16]	Supervised Machine Learning technique	Require a huge number of labeled data
Sunil Kumar Singh [17]	Supervised Machine Learning technique	Only classifies the DoH traffic and traditional DNS traffic
D. Hjelm [33]	Application Fingerprinting	Not flexible and require to build an application fingerprinting database

3. Background Knowledge

3.1. Domain Name System (DNS)

The Domain Name System (DNS) [14] is the Internet's phone book. When a user queries the IP address of a domain name such as 'google.com' or 'facebook.com', DNS resolves the domain name and responds with the IP address of the website. The DNS resolution process is illustrated in Figure 1. The user sends a query to the local DNS resolver. If the local DNS resolver does not find the corresponding IP address in its cache memory, it forwards the DNS query iteratively or recursively through a root server, a Top-Level Domain (TLD) Server, and Authoritative Name Servers until the desired domain name resources are received. The DNS response is forwarded to the application through the Local DNS server. The application can access the web server after it obtains the DNS response.

From the local network to the public network, the data are transmitted in plain text over TCP or UDP in port 53 [14], which is extremely vulnerable to cyber attack. Attackers exploit this process to perform attacks such as DNS amplification [1], DNS hijacking [2], DNS cache poisoning [3], and DNS tunneling [4].

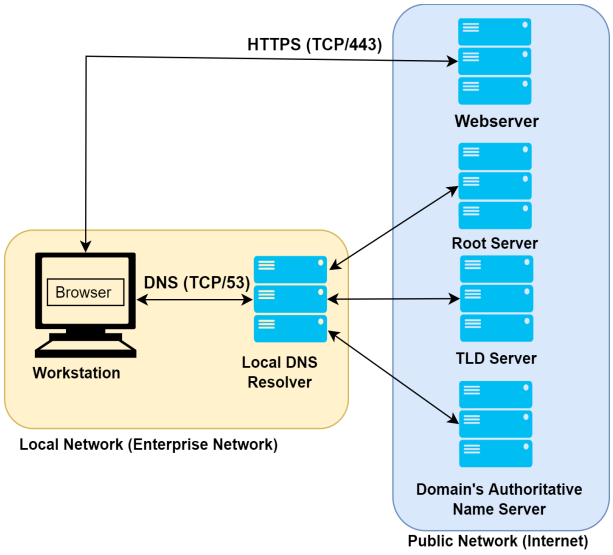

Figure 1. DNS resolution process.

3.2. DNS-over-HTTPS

As one of the new methods to secure DNS communications, DNS-over-HTTPS [13] (or DoH) uses the widely-used HTTPS protocol between DNS clients and DoH resolvers. To use DoH, both the client and the resolver should be able to interpret this protocol. Figure 2 demonstrates how a browser executes DoH. The local DoH proxy wraps a DNS query in an HTTPS request and sends it to the DoH server. The DoH server, which acts as a DNS resolver, receives this request and responds after performing standard DNS resolving activity (such as checking the DNS cache and performing DNS resolution). The response would be back in the HTTPS protocol. Once the browser obtains the IP address, it can access the web server. Using the HTTPS protocol also retains the advantages of the HTTP protocol, such as caching, redirection, and compression. Since all the request and response actions relative to the browser are secure, no user's information is exposed.

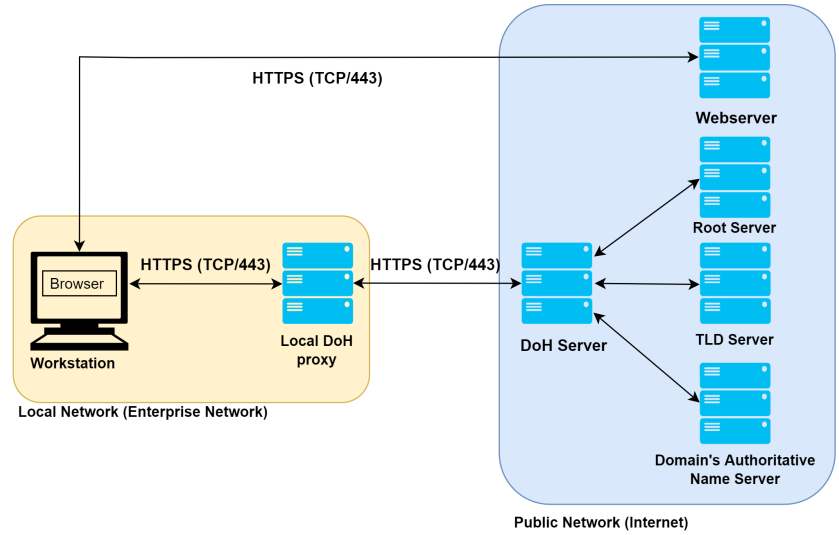

Figure 2. DNS-over-HTTPS resolution.

3.3. DNS-over-HTTPS Vulnerabilities in Enterprise Network

Since DoH is a fast, lightweight, and encrypted communication channel, it can be exploited as a data exfiltration channel. This attack is a much stronger version of DNS tunneling, called DoH tunneling. DNS tunneling can be easily detected by capturing and analyzing the DNS traffic content. However, the DNS tunneling detection methods cannot

be applied to DoH tunneling due to the encryption and the massive HTTPS traffic. In [13], HTTP/2 is a recommended version of HTTP for DoH. The malware can utilize this HTTP/2 connection to send several DoH requests and responses to its server without creating a separate connection (or packet) for each request and response. Through this method, malware can hide the frequency of their DNS resolution and divide the exfiltrating data into small packets to hide their activities. This attack is hazardous for enterprise networks due to the share of network infrastructure, which can quickly spread the malware or bot in the network and steal information from shared databases. DoH tunneling may be slow but very efficient because it is unsusceptible to the traditional threat detection method. This work focuses on detecting DoH tunneling in an enterprise network.

A typical DoH tunneling attack in an enterprise network is shown in Figure 3. Firstly, the attacker registers a domain and embeds it into the malware. The malware reads the exfiltrating data and divides it line by line. Each part of the exfiltrating data is added to the DNS query. The malware periodically sends a DNS query to the local DoH proxy to wrap the DNS query in HTTPS. The local DoH proxy executes a standard DoH process to locate the registered domain name of the attacker and direct the DNS query to that server. When the DNS query with the exfiltrating data arrives, the attacker can extract the needed data by analyzing the DNS query log. Then the attacker server sends back a DNS response to the new malware command-and-control encrypted in the resource record in DNS response format.

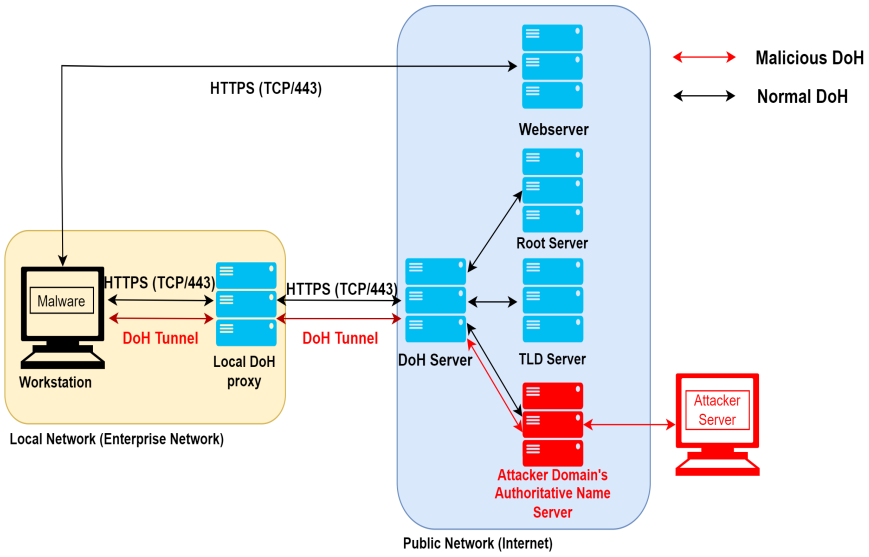

Figure 3. DoH tunneling attack scheme.

3.4. Transformer Architecture

Transformer [20] is a state-of-the-art technique in natural language processing. It was designed to deal with sequence-to-sequence problems, such as machine translation, text classification, and sentiment analysis.

The Transformer model proposes an architecture based on a self-attention mechanism and does not use Recurrent Neural Networks architecture. However, it provides a much better performance than other Sequence-to-sequence models in long sequence processing. Moreover, the transformer architecture processes the input sequences in parallel, which improves the training process time.

Some of important features of Transformer are:

- Encoder: The original encoder of the Transformer architecture has six identical layers, and each layer is constructed with a multi-head self-attention mechanism and a fully connected feed-forward network mechanism. Both of these mechanisms consist of a residual connection and a normalization layer. In our work, we only used four layers.

- Decoder: The original decoder also includes six identical layers, each consisting of an additional sublayer. The additional sublayer performs multi-head attention over the output of the encoder stack.
- Attention Mechanism: The attention component of the network maps the critical and relevant elements from the input sequence and assigns higher weights to these elements, which enhances the accuracy of the output prediction.
- Scaled Dot-Product Attention: Scaled dot-product attention is an attention mechanism that calculates the output as a weighted sum of the values, where the weight assigned to each value is determined by the dot-product of the query with all the keys:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

- Multi-Head Attention: The architecture of multi-head attention is shown in Figure 4. Instead of only computing the attention once, the multi-head attention mechanism operates through the scaled dot-product attention multiple times in parallel. The separated attention outputs are concatenated and linearly transformed into the expected dimensions. The multi-headed-attention matrix for input matrices (Q, K, V) is calculated as:

$$\begin{aligned} \text{MultiHead}(Q, K, V) &= \text{Concat}\,(\text{head}_1, \ldots, \text{head}_{\text{h}})W^O \\ \text{where head}_{\text{i}} &= \text{Attention}\left(QW_i^Q, KW_i^K, VW_i^V\right) \end{aligned} \tag{2}$$

where W_i^Q, W_i^K, W_i^V, and W^O are parameter matrices to be learned.

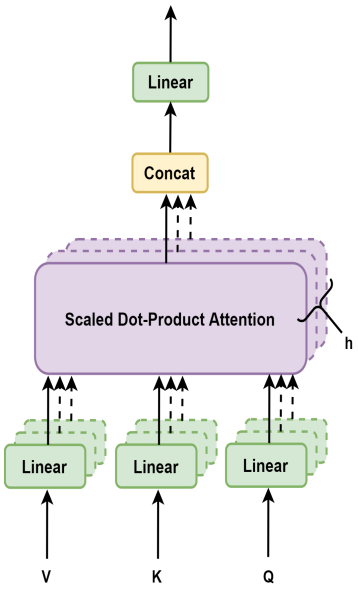

Figure 4. Multi-head attention architecture [20].

3.5. ELK Architecture

The ELK Stack [35] is an architecture including distributed search and analytics engines. The core of ELK Stack is four modules: Elasticsearch, Logstash, Kibana, and Beats. These modules are designed, managed, and supported by Elastic.

- Elasticsearch is a NoSQL database with RESTful APIs based on the Lucene search engine. It is a search and analytics engine that is highly adaptable and dispersed. It has powerful queries for deep analysis and centrally maintains all data for quick document searches. It also provides horizontal scalability, allowing fast deployment, excellent reliability, and control.

- Logstash is a data collection pipeline tool. It is the first part of the ELK Stack, and it collects data and feeds it to Elasticsearch. It gathers a variety of data kinds from numerous sources simultaneously and makes it immediately available for further use.
- Kibana is a data visualization tool. It is used to visualize Elasticsearch data and provide developers with fast access. The Kibana dashboard visualizes sophisticated queries with interactive charts, geographical information, timelines, and diagrams. Kibana allows network managers to create and save individual charts based on their explicit requirements.
- Beats are lightweight agents that are installed at every client's station to collect logs and send them to the ELK host station.

These modules provide general-purpose monitoring, troubleshooting, and securing of the enterprise network infrastructure. Besides the security task, the enterprise network can use the ELK Stack for business intelligence or web analytics.

The workflow begins with Beats collecting data from every client in the network. Then, Logstash aggregates and processes data and sends data to Elasticsearch for indexing and storing, and the Kibana provides a user dashboard that helps the Enterprise Network Security team to monitor all the network traffic and threats.

The ELK Stack provides users with a robust platform that collects and processes data from multiple data sources, stores data in one centralized data store that can be auto-scaled, and offers various tools to analyze the data. The ELK database is well constructed and contains all the information about network activities.

However, the enterprise network may generate a massive number of logs in a short time due to its complexity, which will quickly overload the simple ELK architecture. RabbitMQ adds additional components into ELK architecture to handle a complex enterprise network.

Figure 5 illustrates the architecture of ELK.

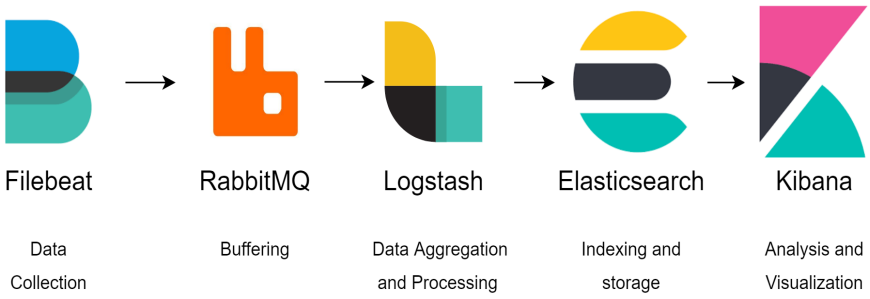

Figure 5. ELK architecture.

3.6. Security Operation Center (SOC) in Enterprise Network

Data breaches, malware infections, and cyberattacks commonly occur for large and small enterprises. Hundreds of new vulnerabilities threatening an enterprise's network environment are found every day. Defensive tools such as a firewall or an antivirus may not detect these threats. The longer unfixed threats result in more significant potential damage and expense to enterprises. Therefore, it becomes a daily priority for network security managers to detect and eradicate the threats before they cause any adverse effects.

Many IT departments address the issue by establishing a separate security operations center (SOC) internally or through a third-party security service provider. An SOC is a facility within an enterprise responsible for continuously monitoring and analyzing an organization's security and improving the organization's security while stopping and reacting to cybersecurity incidents.

SOC teams supervise and defend the organization's assets, including corporation property, confidential documents, and enterprise systems architecture. The SOC team is responsible for implementing the company's entire cybersecurity strategy and serving as a focal point for coordinated efforts to monitor, evaluate, and protect against cyberattacks.

Traditional security solutions such as passwords, Access Control Lists (ACLs), firewall rules, IDS rules, and others are becoming less practical. The enterprise network security team must always ensure that these tools work across all network devices to spot anomalous activity that may cause harm.

The primary responsibilities of an SOC include the following [36]:

- Prevention and detection: In cybersecurity, prevention is more effective than response. An SOC monitors the network to detect attacks rather than responding when they occur. As a result, the SOC team can see malicious activity and stop it before it causes any harm. When SOC analysts notice something unusual, they gather as much information as possible to conduct a more thorough investigation.
- Investigation: The SOC team analyzes anomaly activity at the investigation stage to establish the nature of the threat and the extent to which it has entered the infrastructure. The security analyst examines the network and activities of the company from the perspective of an attacker, looking for essential indicators and sections of weakness before they are exploited. By understanding how assaults escalate and successfully responding before they become uncontrollable, the analyst identifies and executes triage on various security problems. To achieve a successful defense, the SOC analyst combines the enterprise's network expertise with the most recent global threat intelligence, including specifics on attacker tools, techniques, and trends.
- Response: Following the investigation, the SOC team plans a reaction to address the issue. The SOC serves as the first responder as soon as an incident is confirmed, isolating endpoints, eliminating hazardous programs, stopping them from executing, removing them, and more.

The SOC attempts to restore systems and retrieve any lost or compromised data in the aftermath of an event. The process may include wiping and restarting endpoints, changing systems, or, in the case of ransomware attacks, deploying viable backups to evade the ransomware. This phase, if successful, will restore the network's status before the attack.

4. Model Development and System Architecture

In this section, we present the complete system, including the data collection process, the construction of two-layer classification models, and the system implementation into an enterprise network.

4.1. DoH Traffic Collection and Statistical Feature Extraction

To capture HTTPS traffic in this work, we simulate the works of several users in the enterprise network. Figure 6 shows the network topology used to capture the DoH traffic. All regular and malicious users send queries into multi-local DNS servers using the DoH protocol in our script. For ordinary users, we generated DoH traffic by accessing the 10,000 most popular websites, ranked by Alexa [37], with DoH extensions of web browsers. We also applied the four most popular DoH servers and open-source DoH tunneling tools to create the encrypted TCP traffic through the DoH protocol to simulate malware or bot activities. The data capture was saved in .pcap file format. The simulation parameters are as follows:

1. Browser: Google Chrome, Filefox.
2. DoH Server: Adguard, Cloudflare, Google, and Quad9.
3. DoH tunneling tools: dns2tcp [38], DNSCat2 [39], Iodine [40].
4. Traffic capture: tcpdump [41].
5. Data rate: Random from 100 B/s to 1000 B/s.

The statistical features of the PCAP files are extracted by CICFlowMeter [42]. CICFlowMeter generates two datasets. The first dataset is used for the first layer of the process to classify DoH traffic and non-DoH traffic, and the second dataset is used for the second layer to classify DoH traffic and malicious DoH traffic. Table 2 shows the list of statistical components extracted from CICFlowMeter.

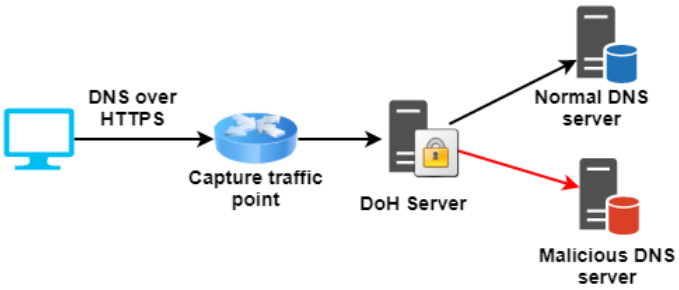

Figure 6. Raw DoH traffic capture.

Table 2. List of the 34 statistical features extracted from captured traffic using CICFlowmeter.

No.	Feature Name
1	Duration
2	Flow bytes sent
3	Flow sent rate
4	Flow bytes received
5	Flow received rate
6	Packet length variance
7	Packet length standard deviation
8	Packet length mean
9	Packet length median
10	Packet length Mode
11	Packet length skew from median
12	Packet length skew from mode
13	Packet length coefficient of variation
14	Packet time variance
15	Packet time standard deviation
16	Packet time mean
17	Packet time median
18	Packet time mode
19	Packet time skew from median
20	Packet time skew from mode
21	Packet time coefficient of variation
22	Response time variance
23	Response time standard deviation
24	Response time mean
25	Response time median
26	Response time mode
27	Response time skew from median
28	Response time skew from mode
29	Response time coefficient of variation

4.2. Training Process

Figure 7 displays the training process phases for DoH tunneling attack detection. This process begins with the data preprocessing phase, removing the data point with infinity and nulls and replacing it with the mean value of that feature in the dataset. This step guarantees that only meaningful data points can be fed into our model. Then, data are provided into two-layer classification, both using Transformer architecture. In the previous works, Refs. [15–17], the ratio of DoH traffic to regular HTTPS traffic is 8:2, and the ratio of normal DoH traffic to malicious traffic is 10:1. We kept the same percentage in our work to make a fair comparison. The first layer objective is to isolate DoH traffic from non-DoH traffic. The dataset for the first layer included 233,427 instances, which is only 20% of instances compared with the previous work, and the ratio of DoH traffic to regular HTTPS is 8:2. After the training process, the model was used to classify the rest of the dataset to check the accuracy of the model. The second layer categorizes instances as malicious DoH or normal DoH. From the DoH traffic classified from the first layer, we randomly chose

53,928 instances, which are only 20% of instances compared with the previous work that had a ratio between regular DoH traffic and malicious DoH of 10:1. In both layers, we used the same Transformer architecture with similar parameters. The models were trained on a local CPU (Intel Core i7 8th 3.20 GHz) and GPU NVIDIA GeForce GTX-3060 8 GB (32 GB memory). The model parameters were constructed as in Table 3.

Table 3. Model hyperparameters.

Hyperparameters	Values
Number of attention heads	4
Sequence length	128
Number of Transformer Blocks	4
Dropout	0.25
Batch size	64
Epochs	120
Optimizer	ADAM

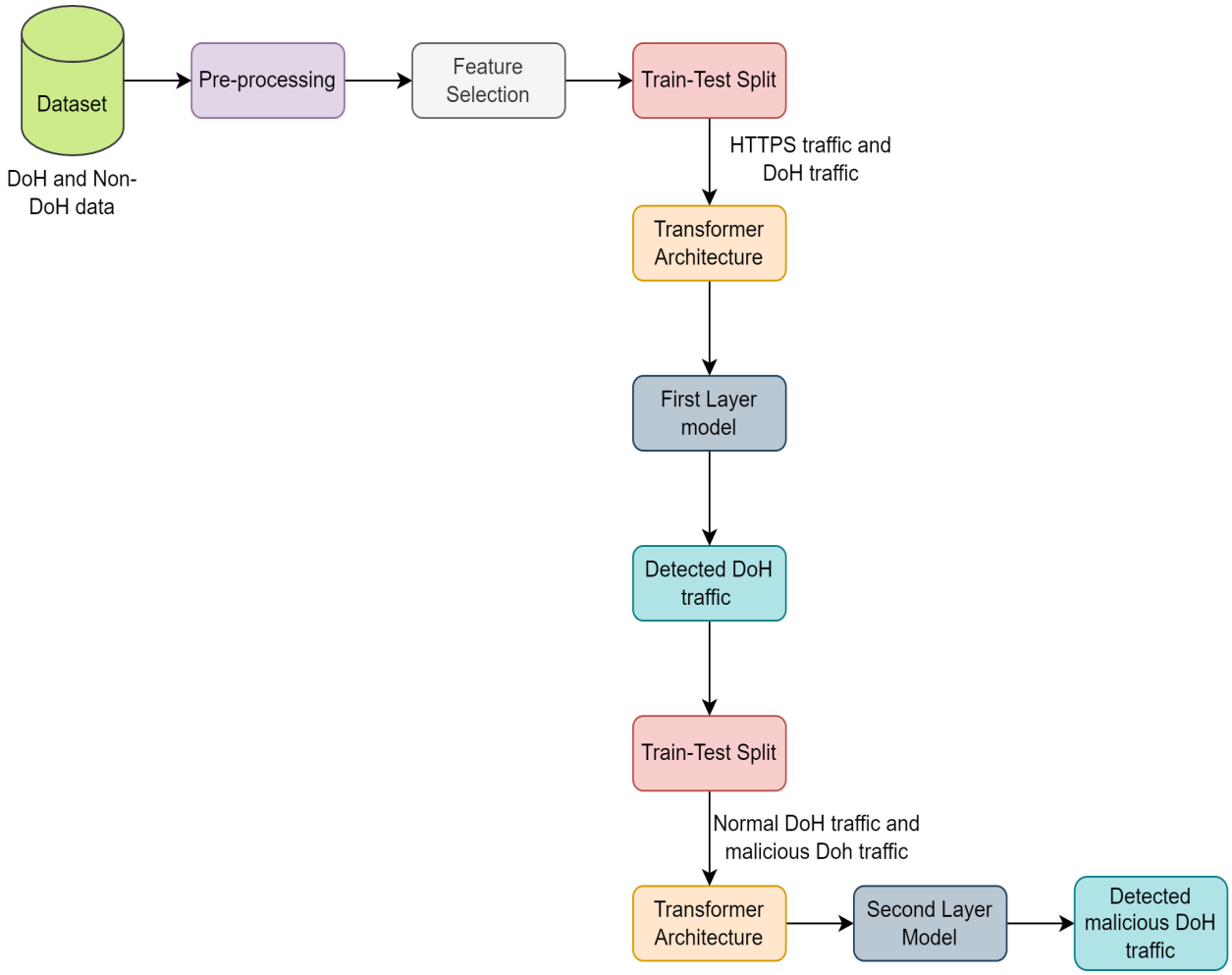

Figure 7. Training process phases for DoH tunneling attack detection.

4.3. System Implementation

Our proposed system architecture can work as a supporting tool for the SOC team in protecting the Enterprise DNS system. The ELK architecture offers extensive investigations and correlations from numerous sources on a single platform by concentrating logs from the complex cloud environment. The capacity for real-time inquiry and visualization cuts down on time to discover insights and enables continuous monitoring. The ELK also

supports many other platforms that help integrate other modules. Because of the limitation in our research scope, we only focus on the DoH tunneling attack. In practice, other deep learning models can be applied to our system to detect and prevent further attacks. Figure 8 shows the detail of our system.

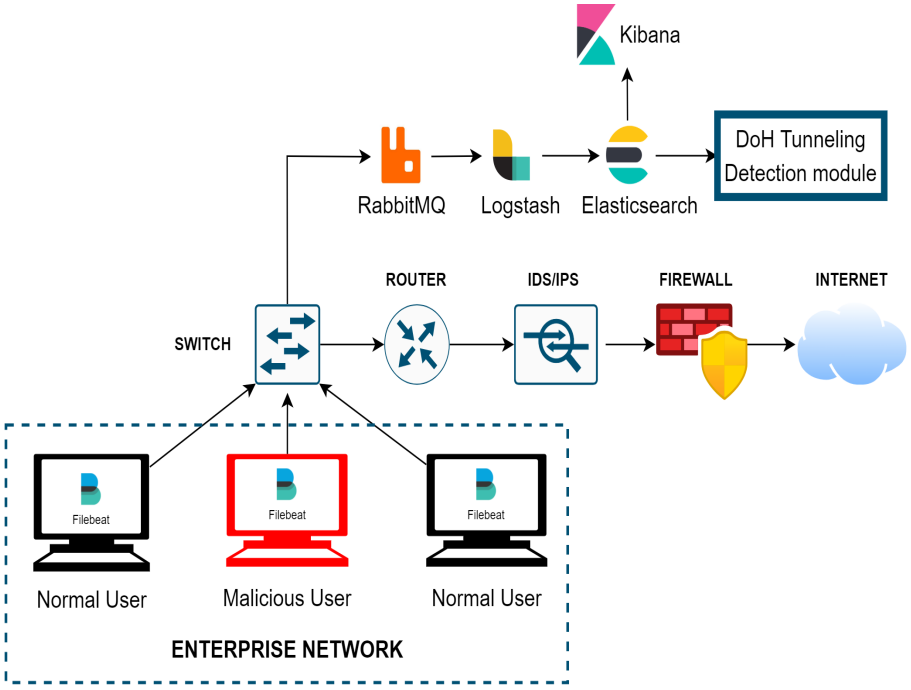

Figure 8. DoH tunneling detection system for SOC in an enterprise network.

5. Results and Evaluation

Figures 9–12 show the training and testing results of the Transformer models. We trained our model in 120 epochs and extracted the accuracy and sparse category cross-entropy loss. The best results for the first layer are 99% accuracy and 0.048 in loss after 63 epochs, and the second layer is 99.4% accuracy and 0.033 in loss after 118 epochs. If we increase the number of epochs, the best results are unchanged. The gaps between the training and validation lines are small, proving that our models are convergent and not overfitted.

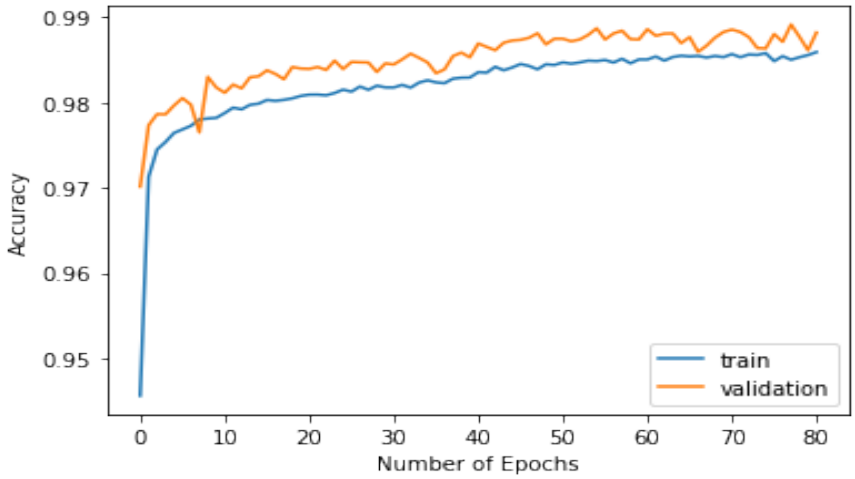

Figure 9. Model accuracy in training and validation phases in first layer.

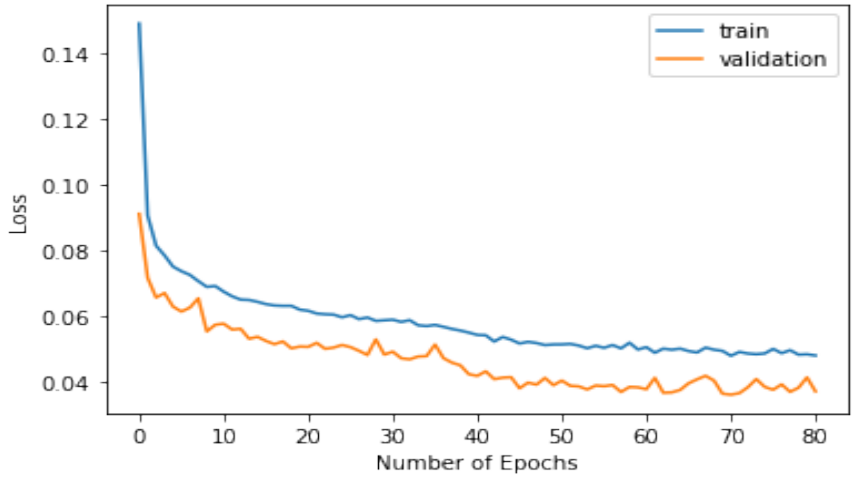

Figure 10. Model loss in training and validation phases in first layer.

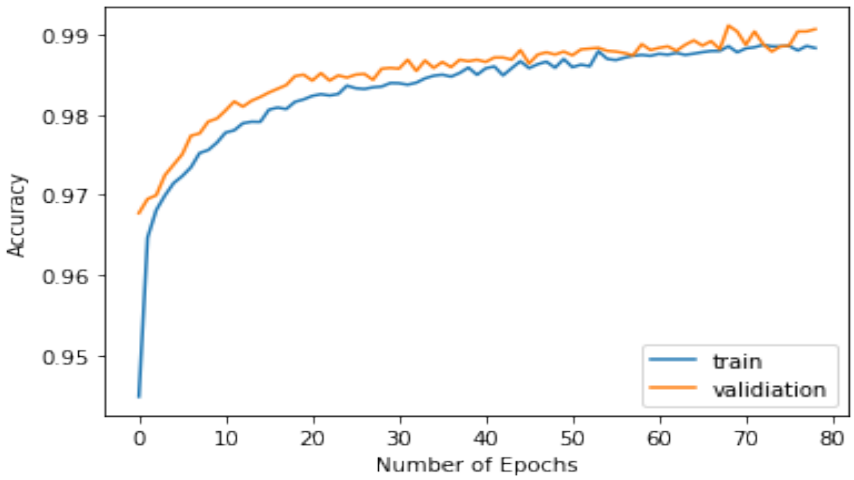

Figure 11. Model accuracy in training and validation phases in second layer.

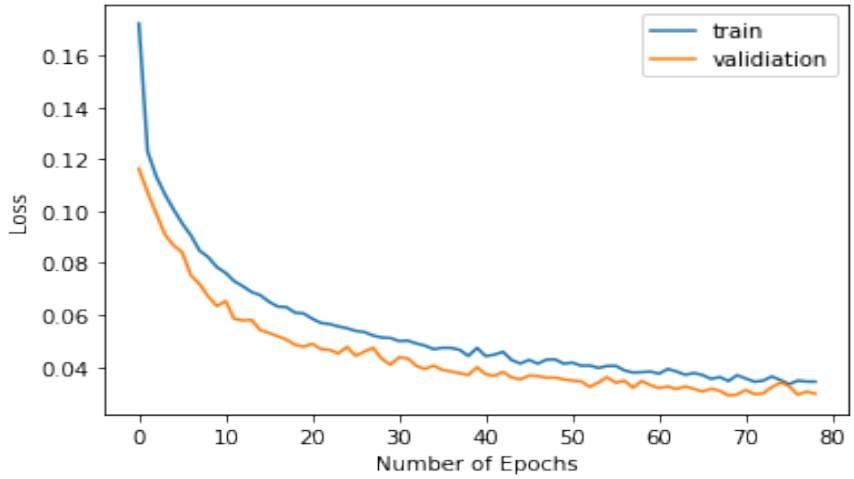

Figure 12. Model loss in training and validation phases in second layer.

To confirm the efficacy of the offered DoH tunneling detection system, we compared our model's evaluation parameters with other research in [15–17]. We implemented their best algorithms with our dataset. Table 4 shows the comparison between the performance metrics of our work and the previous works. The results show that we obtain better results

with our two-layer Transformer model than with the earlier algorithms if we use the same number of labeled data. These results prove that our model can work more effectively in practicality when labeling the DoH data is very hard. The performance metrics used were:

- *Precision*: defined as the number of true positives (*TP*) over the number of true positives plus the number of false positives (*FN*)

$$Precision = \frac{TP}{TP + FN} \tag{3}$$

- *Recall*: defined as the number of true positives *(TP)* over the number of true positives plus the number of false negatives.

$$Recall = \frac{TP}{TP + FP} \tag{4}$$

- *F*1-*scorce*: the adjustment average of the precision and recall

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{5}$$

Table 4. Model evaluation.

Model	Number of Instances in Training Data (Layer 1/ Layer 2)	Precision (Layer 1/ Layer 2)	Recall (Layer 1/ Layer 2)	F1-Score (Layer 1/ Layer 2)
RF [15]	233,427/53,928	0.92/0.93	0.92/0.93	0.92/0.93
C4.5 [15]	233,427/53,928	0.92/0.93	0.92/0.92	0.92/0.92
2D CNN [16]	233,427/53,928	0.91/0.91	0.91/0.91	0.91/0.91
XGBoot [17]	233,427/53,928	0.94/0.94	0.94/0.94	0.94/0.94
Transformer (our work)	233,427/53,928	0.99/0.994	0.99/0.994	0.99/0.994

6. Conclusions

With the emergence of cloud-based technologies, organization-specific security requirements for enterprise are changing. The traditional approach of addressing cybersecurity and physical security separately is no longer sufficient to protect an enterprise network. DNS is a crucial part of every network architecture. However, the traditional DNS architecture is vulnerable to cyberattacks. DoH is a privacy and security enhancement technology that uses HTTPS as an encryption technique for DNS queries. While DoH protects the data associated with the DNS query from DNS assaults, it also can be exploited by hackers to launch DoH tunneling attacks. Our research proposed a system that can be applied to enterprise networks as a security operation system. Our approach used the Transformer architecture as a classifier method to detect malicious DoH. Although the Transformer architecture is more complex than other Supervised Machine Learning models, it showed a significant improvement in the number of labeled data used. The results show that the Transformer model also achieved a very high accuracy of 0.994 but only needed a small number of labeled data, only 20% compared with other research. This advantage of the Transformer architecture makes it more suitable in malicious DoH tunneling detection because, in practice, labeling a large amount of encrypted network traffic is very complex work and requires many resources.

Author Contributions: Conceptualization, T.A.N.; project administration, M.P.; supervision, M.P.; writing—original draft, T.A.N. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2020R1F1A1076795).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Houser, R.; Hao, S.; Li, Z.; Liu, D.; Cotton, C.; Wang, H. A Comprehensive Measurement-based Investigation of DNS Hijacking. In Proceedings of the 2021 40th International Symposium on Reliable Distributed Systems (SRDS), Chicago, IL, USA, 20–23 September 2021; pp. 210–221. [CrossRef]
2. Stewart, J. Dns Cache Poisoning—The Next Generation. 2003. Available online: https://www.ida.liu.se/~TDDD17/literature/dnscache.pdf (accessed on 23 February 2022).
3. Vaughn, R. DNS Amplification Attacks (Preliminary Release). 2006. Available online: http://index-of.es/Tutorials/AstalaVista/dns-amplification-attacks.pdf (accessed on 23 February 2022).
4. Wang, Y.; Zhou, A.; Liao, S.; Zheng, R.; Hu, R.; Zhang, L. A comprehensive survey on DNS tunnel detection. *Comput. Networks* **2021**, *197*, 108322. [CrossRef]
5. OpenDNS. Available online: https://www.opendns.com/ (accessed on 23 December 2021).
6. DNSFilter. Available online: https://www.dnsfilter.com/ (accessed on 23 December 2021).
7. Infoblox BloxOne Threat Defense. Available online: https://www.infoblox.com/products/bloxone-threat-defense/ (accessed on 23 December 2021).
8. Bieler, D.; Kindness, A. The Enterprise Network Enables Business Innovation. Available online: https://www.hughes.com/ (accessed on 9 February 2021).
9. Domain Name System—Request For Comments 1034. Available online: https://tools.ietf.org/html/rfc1034 (accessed on 9 February 2022).
10. Carli, F. Security Issues with DNS. Available online: https://www.sans.org/white-papers/1069/ (accessed on 9 February 2022).
11. Kim, T.H.; Reeves, D. A survey of domain name system vulnerabilities and attacks. *J. Surveill. Secur. Saf.* **2020**, *1*, 34–60. [CrossRef]
12. Guha, S.; Francis, P. Identity trail: Covert surveillance using DNS. In Proceedings of the Internationl Workshop on Privacy Enhancing Technologies (PET), Ottawa, BC, Canada, 20–22 June 2007.
13. Hoffman, P.; McManus, P. RFC 8484—DNS Queries over HTTPS. October 2018. Available online: https://datatracker.ietf.org/doc/html/rfc8484 (accessed on 23 May 2021).
14. Mockapetris, P. Domain Names—Implementation and Specification. Available online: https://datatracker.ietf.org/doc/html/rfc1035 (accessed on 23 February 2022).
15. Montazeri Shatoori, M.; Davidson, L.; Kaur, G.; Lashkari, A.H. Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic. In Proceedings of the 2020 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Calgary, AB, Canada, 17–22 August 2020; pp. 63–70. [CrossRef]
16. Mitsuhashi, R.; Satoh, A.; Jin, Y.; Iida, K.; Shinagawa, T.; Takai, Y. Identifying Malicious DNS Tunnel Tools from DoH Traffic Using Hierarchical Machine Learning Classification. In *Information Security. ISC 2021*; Lecture Notes in Computer Science; Liu, J.K., Katsikas, S., Meng, W., Susilo, W., Intan, R., Eds.; Springer: Cham, Switzerland, 2021; Volume 13118. [CrossRef]
17. Singh, S.K.; Roy, P.K. Detecting Malicious DNS over HTTPS Traffic Using Machine Learning. In Proceedings of the 2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT), Sakheer, Bahrain, 20–21 December 2020; pp. 1–6. [CrossRef]
18. De Vries, L.J.W. Detection of DoH Tunnelling: Comparing Supervised with Unsupervised Learning. 2021. Available online: http://essay.utwente.nl/88335/ (accessed on 23 December 2021).
19. Rezaei, S.; Liu, X. Deep Learning for Encrypted Traffic Classification: An Overview. *IEEE Commun. Mag.* **2019**, *57*, 76–81. [CrossRef]
20. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
21. CISA Alert (TA13-088A) DNS Amplication Attacks. Available online: https://www.cisa.gov/uscert/ncas/alerts/TA13-088A (accessed on 23 February 2022).
22. Maksutov, A.A.; Cherepanov, I.A.; Alekseev, M.S. Detection and prevention of DNS spoofing attacks. In Proceedings of the 2017 Siberian Symposium on Data Science and Engineering (SSDSE), Novosibirsk, Russia, 12–13 April 2017; pp. 84–87. [CrossRef]
23. Arends, R.; Telematica Instituut. "DNS Security Introduction and Requirements" RFC4033. Available online: https://datatracker.ietf.org/doc/html/rfc4033 (accessed on 9 February 2022).
24. Aiello, M.; Mongelli, M.; Papaleo, G. DNS tunneling detection through statistical fingerprints of protocol messages and machine learning. *Int. J. Commun. Syst.* **2015**, *28*, 1987–2002. [CrossRef]
25. Bottger, T.; Cuadrado, F.; Antichi, G.; Fernandes, E.L.; Tyson, G.; Castro, I.; Uhlig, S. An Empirical Study of the Cost of DNS-over-HTTPS. In Proceedings of the Internet Measurement Conference, Amsterdam, The Netherlands, 21–23 October 2019; pp. 15–21.

26. Hounsel, A.; Borgolte, K.; Schmitt, P.; Holland, J.; Feamster, N. Analyzing the costs (and benefits) of DNS, DoT, and DoH for the modern web. In Proceedings of the Applied Networking Research Workshop, Montreal, QC, Canada, 22 July 2019; pp. 20–22.
27. Wijenbergh, J.; Moonsamy, V.; van Rijsdijk-Deij, R.; Kuijsters, D.D. Performance comparison of DNS over HTTPS to Unencrypted DNS. Bachelor's Thesis, Radboud University, Nijmegen, The Netherlands, 2019.
28. Siby, S.; Juarez, M.; Vallina-Rodriguez, N.; Troncoso, C. DNS Privacy not So Private: The Traffic Analysis Perspective. 2018. Available online: https://cpb-us-e1.wpmucdn.com/sites.usc.edu/dist/5/475/files/2020/01/hotpets18.pdf (accessed on 23 December 2021).
29. Siby, S.; Juarez, M.; Diaz, C.; Vallina-Rodriguez, N.; Troncoso, C. Encrypted DNS → Privacy? A Traffic Analysis Perspective. *arXiv* **2019**, arXiv:1906.09682.
30. Bushart, J.; Rossow, C. Padding Ain't Enough: Assessing the Privacy Guarantees of Encrypted DNS. *arXiv* **2019**, arXiv:1907.01317.
31. Lu, C.; Liu, B.; Li, Z.; Hao, S.; Duan, H.; Zhang, M.; Leng, C.; Liu, Y.; Zhang, Z.; Wu, J. An end-to-end, large-scale measurement of DNS-over-Encryption: How far have we come? In Proceedings of the Internet Measurement Conference, Amsterdam, The Netherlands, 21–23 October 2019; pp. 22–35.
32. Nijeboer, F. Detection of Https Encrypted Dns Traffic. Bachelor's Thesis, University of Twente, Enschede, The Netherlands, 2020.
33. Hjelm, D. A New Needle and Haystack: Detecting DNS over HTTPS Usage. Available online: https://www.sans.org/white-papers/39160/ (accessed on 23 December 2021).
34. Godlua Backdoor. Available online: https://blog.netlab.360.com/an-analysis-of-godlua-backdoor-en/ (accessed on 23 December 2021).
35. What Is the ELK Stack? Available online: https://www.elastic.co/what-is/elk-stack (accessed on 23 December 2021).
36. Checkpoint SOC. Available online: https://www.checkpoint.com/cyber-hub/threat-prevention/what-is-soc/(accessed on 23 December 2021).
37. Alexa Traffic Rank. Available online: https://www.alexa.com/topsites (accessed on 23 December 2021).
38. dns2tcp. Available online: https://github.com/alex-sector/dns2tcp (accessed on 23 December 2021).
39. DNSCat2. Available online: https://github.com/iagox86/dnscat2 (accessed on 23 December 2021).
40. Iodine. Available online: https://github.com/yarrick/iodine (accessed on 23 December 2021).
41. tcpdump. Available online: https://github.com/the-tcpdump-group/tcpdump (accessed on 23 December 2021).
42. CICFlowMeter. Available online: https://github.com/ahlashkari/CICFlowMeter (accessed on 23 December 2021).