Survey of Methods to Detect DNS over HTTPS (DoH) Traffic

Praneeth Babu Marella[1,2]

[1]ISA 656: Network Security

[2]George Mason University, Fairfax, VA 22030, USA

Domain Name System (DNS) acts as a phone book of the internet and plays a vital role for users of the internet. It translates a human memorable domain name to an IP address, allowing the user to locate and access the resource hosted on the internet without having the user remember the IP address. Traditionally, DNS resolution is done over UDP in plaintext, which poses some risks related to privacy and security. Some of the specific risks that traditional DNS resolution is susceptible to are DNS Spoofing, eavesdropping, etc. DNS over HTTPS (DoH) was introduced as a standard in 2018 as a new way to perform DNS resolutions that addresses the privacy and security risks that the DNS over UDP (traditional) method has. DoH uses the HTTPS protocol as a method of communication for DNS related traffic, which allows for encryption of the DNS resolution and prevents data manipulation via various man-in-the-middle attacks and eavesdropping. While DoH improves privacy and security for its users, it also brings a lot of challenges for the security teams, especially in an enterprise environment, by making network traffic monitoring related to DNS harder and rendering most traditional security measures useless. DoH traffic now also blends in with regular HTTPS web browser traffic, making it harder to differentiate between DNS traffic or regular web traffic. So, new methods of detection need to be researched and utilized to effectively prevent or monitor DNS resolution traffic occurring over HTTPS. In this paper, we will be exploring several methods of detections that currently exist or being researched on. The detection methods we will be exploring primarily fall into one of the following categories: configuration managed, signature based, anomaly based, and machine learning based. Our main focus in this paper will be on the latter three categories.

1. Introduction

Domain Name Service (DNS) is an important part of the internet and plays an important role in network security as well. Domain Name System (DNS) acts as the phone book of the Internet. Whenever you type/click on a website name in your browser to visit, i.e. facebook.com, the browser needs to get the IP address associated with that domain since communication on the internet occurs via IP addresses. To find the IP address, a DNS resolution request is sent by your client (like a browser or a service application). Traditionally, these lookups are done via plaintext using the User Datagram Protocol (UDP) over port 53, this has user privacy and security concerns. DNS resolution activity via UDP can enable eavesdroppers to snoop and profile users because they can get access to the user's web browsing history via DNS resolution request tracking. Attackers can also sit in the middle and attempt to modify the plaintext DNS resolution requests in transit and redirect users to malicious websites by providing an IP address that is under their control instead of the IP related to the actual domain being resolved. Hence, there was a need for DNS resolutions to be more private and secure and various alternative methods

have been proposed throughout the years. One of the most recent and widely adopted method is DNS over HTTPS (DoH). DNS lookups in DoH occur over Transmission Control Protocol (TCP) over port 443 and uses an encrypted channel to send and receive DNS resolution traffic.

DoH was proposed as a standard via IETF in 2018 by Hoffman and McManus as an alternative method to traditional DNS resolution over UDP (Hoffman & McManus 2018). The use of encryption in DoH for transport makes it challenging for eavesdroppers to listen to the DNS traffic and profile users. It also makes it harder for attackers to perform man-in-the-middle attacks and manipulate DNS queries to direct users to malicious sites under the attacker's control. Some current popular public DoH resolvers are Cloudflare, AdGuard, Quad9, and Google. Since, DoH uses port 443 which is also used by regular HTTPS web traffic, so the DNS resolution traffic and the HTTPS web traffic is now jumbled together into one. This causes some challenges for the security team in organizations who rely on DNS traffic monitoring as a method to implement network security controls.

DNS blocklists are largely used by security teams as secutrity control to prevent users within an organization from accessing malicious sites or even stop malware from advancing by blocking DNS resolution for a C2 domain. This security functionality is enabled because DNS resolution occurs over UDP and various tools can inspect and check the domain name that is being requested to be resolved. But with DoH the packets are encrypted, which makes it difficult to continue using the same methods/security controls. So, new solutions to detect DoH need to be researched and implemented to aid in preventing or inspecting DoH traffic within an organization.

There are already a few various methods proposed to detect DoH traffic. These approaches include using Application Logging to detect DNS resolutions by inspecting DNS files, using JA3/S detections implemented in Zeek, or using Machine Learning methods to differentiate DoH traffic from regular HTTPS web traffic. In this paper, we are going to review some of these methods mentioned to detect DoH traffic using configuration managed, signature based, anomaly based, and machine learning based detections. We are going to explore the effectiveness of each of the detection methods proposed. Finally, we are also going to briefly discuss future possibilities for detection in the area of DoH traffic monitoring.

The rest of this paper is structured as follows. Starting with section two, I discuss all the literature I reviewed in brief detail. In section three, I will discuss the difference between traditional DNS and DoH and the public threats that leverage DoH. In section four, I will cover the various detection methods proposed and evaluate them, as well as explore future opportunities in detection. In section five, I conclude my review of the detection methods and provide my references.

2. Related Works

We will be looking at some related works that discuss the security, privacy, and detection of DoH traffic within a network that were used as part of my literature review.

How DNS over HTTPS is Reshaping Privacy, Performance, and Policy in the Internet Ecosystem (Borgolte *et al.* 2019) explores and identifies the performance, privacy, regulatory policy implications that arise from the usage of DNS over HTTPS. They measure different load time difference when using DoH with different providers, evaluates arising issues in privacy and security which includes ISPs trying to block the widespread usage of DoH and countries trying to prevent DoH to preserve tracking capabilities. Finally, they also evaluate the concerns regarding the filtering and blocking capabilities that

are disrupted by the use of DoH. Their conclusion is that one must stay vigilant when implementing DoH and ultimately it isn't discriminatory or anti-competitive.

Analyzing the Costs (and Benefits) of DNS, DoT, and DoH for the Modern Web (Hounsel *et al.* 2019) aims to compare the performance of various DNS resolution methods and conclude which modern method performs the best. The methods evaluated in the paper include: DNS over TLS, DNS over HTTPS, and Do53 (aka DNS over UDP). To measure and compare performance differences between the methods, they go to specific webpages and measure the load times. They conclude that the DNS over TLS performed the best but, ultimately, it can be highly dependent on the network conditions at play.

An End-to-End, Large-Scale Measurement of DNS-over-Encryption: How Far Have We Come? (Lu *et al.* 2019) tries to show how DNS-over-Encryption, protocols like DNS over HTTPS or DNS over TLS, can scale well and has minimal impact on performance to push for wide adoption by service providers. This paper claims to be the first large scale performance dataset of different types of DNS-over-Encryption traffic. They successfully prove that DNS-over-Encryption does scale well and there is little overhead to using DNS-over-Encryption. But the authors do warn that the service providers must be very careful while implementing it as it could lead to invalid certificate issues, etc. if implemented incorrectly.

Privacy analysis of DNS resolver solutions (Van Heugten 2018) sets out to answer the question "How can modern techniques improve the privacy of DNS users?". To answer this question, the author evaluates various combinations of DNS resolver setups to try and maximize the privacy we can achieve for DNS resolutions. Some of the discussed methods in the paper include: Oblivious DNS, QNAME minimization, DNSCrypt, DNS over HTTPS, DNSSEC, etc. After experimentation with several combinations of DNS resolver setups, they conclude that you can achieve the best combination for privacy by having QNAME minimization and a forwarding resolver, you can achieve the best combination for privacy. In addition, they also propose that any one of the other combinations they evaluated in the paper are just as viable. In the paper, the part that we really drill down on is about the privacy considerations in DNS over HTTPS. The paper concludes that DoH provides great privacy since it used TLS to encrypt the traffic and is mixed in with regular HTTPS web traffic, which enhances privacy.

DNS Privacy not so private: the traffic analysis perspective (Siby *et al.* 2018) had the goal of attempting to determine whether it is possible or not to fingerprint and identify web pages from observed encrypted DNS traffic. By selecting the following features: size, timing, and ordering of the HTTPS traffic packets, the authors believe they will be able to distinguish DNS over HTTPS traffic based on the request and response packets. For their experimentation, they set up a Raspberry Pi as one of their DoH clients that sends single and multi-queries to DoH resolvers hosted by Google and Cloudflare. Then, they recorded the network traffic for the DNS requests and responses and used that data for their packet analysis. The conclusion of their analysis was that packet size by itself was enough to determine the difference in traffic.

A New Needle and Haystack: Detecting DNS over HTTPS Usage (Hjelm 2019) aims to explore various methods to detect DNS over HTTPS traffic. The author starts with exploring various types of new threats DoH introduces to the landscape and discusses the significance of having detection methods in place to mitigate the risk the new threats pose. Then, explores several different detection methods like TLS Inspection, Zeek, Application Fingerprinting, RITA, etc. and describes how they would work and shows their implementation in a lab created for experimenting with DoH traffic. The author concludes by reiterating the fact that DoH is being used in the wild by threat

actors and implores the readers to explore existing methods described in the paper to detect DoH traffic in their environment.

DoH Insight: Detecting DNS over HTTPS by Machine Learning (Vekshin *et al.* 2020) explored the possibility of performing analysis using Machine Learning on encrypted traffic to determine if it is DoH traffic or not. In order to perform this analysis, they used five different machine learning classifiers and created a dataset with captured HTTPS and DoH traffic to use for training and testing those classifier models. That data was then processed and important features were extracted, the selected features were ranked by the importance they play in classifying certain traffic as DoH or HTTPS. For example, duration has an importance scoring of 0.239 while pktInPauses has only 0.015. They were able to identify 18 features that were selected as important during the feature selection phase. Using the dataset with the important features, they trained and tested using the following ML classifiers: 5-NN K-Nearest Neighbours, C4.5 Decision Tree, Random Forest, Naive Bayes, and Ada-boosted Decision Tree. They were able to get 99.6% accuracy using the 5-NN K-Nearest Neighbours ML classifier on the important feature dataset of the HTTPS and DoH traffic packets.

Detecting Malicious DNS over HTTPS Traffic Using Machine Learning (Singh & Roy 2020) used Machine Learning models to classify DoH traffic as malicious or non-malicious. They tested the model with various machine learning classifiers such as Naive Bayes, Logistic Regression, Random Forest, K-Nearest Neighbor, and Gradient Boosting to detect the malicious activity at DNS level in the DoH environment. The testing was done on a dataset with regular HTTPS traffic, benign DoH traffic, and malicious DoH traffic packet captures. They created regular DoH traffic by navigating to various popular websites and used public DoH resolver service like Google and Cloudflare, and then they used DNS tunneling tools like DNSCat2 and dns2tcp to create malicious DoH traffic and captured that. From those packets, they performed feature extraction using a statistical tool to extract about 28 statistical features that are useful in classifying the different types of traffic. Examples of some of these features are: Median Packet Length, Standard Deviation of Packet Time, Variance of Request/response time difference, Variance of Request/response time difference, etc. These statistical features were used to train and test the various machine learning classifiers. The conclusion of the paper was that Random Forest and Gradient Boosting classifiers were better at classifying certain traffic is malicious DoH or not. Furthermore, they also stated that ML-based algorithms are the best at detection DoH traffic and should be used to prevent DNS attacks on DoH traffic.

3. DNS over HTTPS and its Threats

In this section, I will briefly go over how the DNS process for traditional DNS and DNS over HTTPS works and highlight their difference. It is important that we understand what the differences are between traditional DNS and DNS over HTTPS to understand the rest of the paper. Then I will talk about some real-world threats that are leveraging DNS over HTTPS.

3.1. *Do53 (DNS over UDP, aka Traditional)*

Detailed in figure 1 above is the steps for the process for a regular DNS resolution using Do53 (DNS over UDP). Starting with step 1, the user types in a domain they want to visit into the Firefox browser and the request to resolve the domain to its IP address starts. It first sends the request to the client's DNS Resolver. Then in step 2 to 4, the DNS request is forwarded from the local DNS resolver to a public DNS resolver.

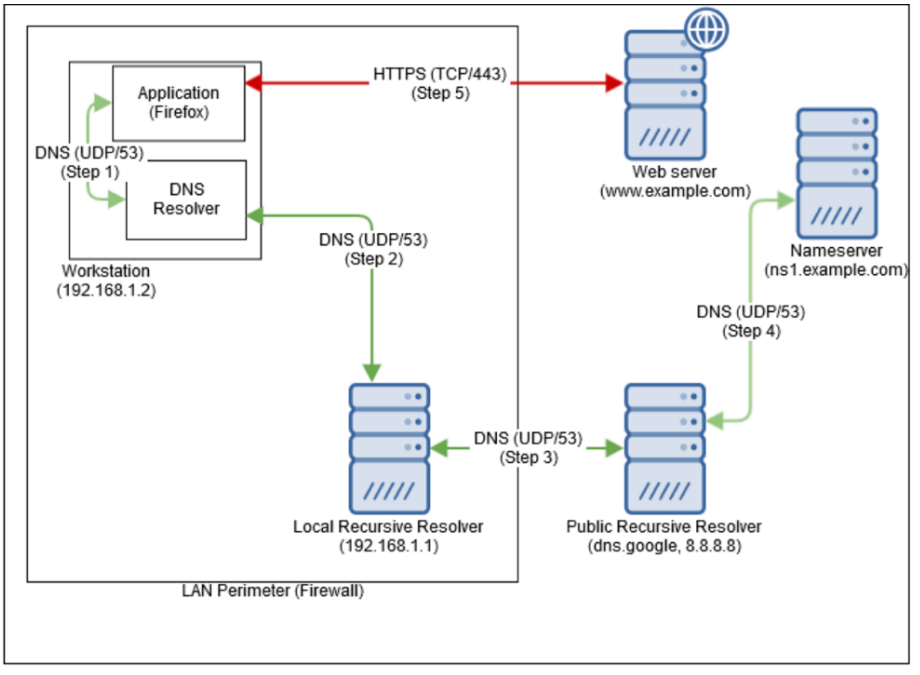

FIGURE 1. DNS Resolution. (Hjelm 2019)

This public DNS resolver will then look up the different levels of the domain and points to the nameserver that holds the IP information for that domain. All the pieces are then put together and the IP address pointing to the domain is sent back to the user, allowing them to visit the website via the Firefox browser, as seen in step 5. All of the DNS resolution traffic in steps 2 to 4 is in plaintext over UDP port 53 and can easily be monitored.

3.2. *DoH (DNS over HTTPS)*

Detailed in figure 2 above is the steps for the process for a DNS resolution using DoH (DNS over HTTPS). Starting with step 1, the user types in a domain they want to visit into the Firefox browser and the request to resolve the domain to its IP address starts. In the same, step 1, the DNS request is directly forwarded to a public DoH resolver via HTTPS over port 443. Then in step 2, the DoH resolver checks to see if it cached the IP address for the domain request. If not, then the DoH resolver will try to resolve the domain over UDP via port 53 at the nameserver. The IP address will then be forwarded back to the user via HTTPS and allows the user to navigate to the website either via HTTPS or HTTP in step 3. Since all the resolution steps from the user to the DoH resolver and the response back to the user are over HTTPS, the traffic is encrypted and blends in with regular HTTPS web traffic which makes it really difficult to detect.

3.3. *Real World Threats*

Having the ability to monitor network traffic in an organization is important and has many benefits to overall security controls and risk posture. Network traffic monitoring is important because it helps us monitor and manage network connection for users and prevent them from navigating to risky and/or malicious websites. The primary way a user is prevented from visiting these types of websites is by blocking the DNS resolutions

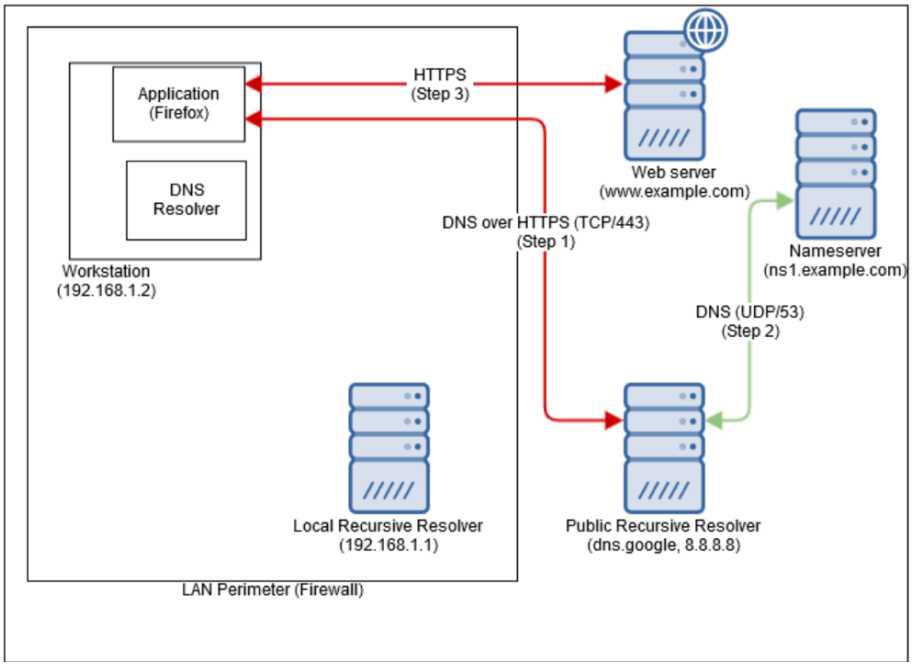

FIGURE 2. DNS over HTTPS. (Hjelm 2019)

for those websites by either pointing them to a sinkhole or just dropping the request. With Do53 (DNS over UDP), it is fairly straightforward to implement some sort of blocklist because we can see the request in plaintext and easily compared to our managed blocklist. But with DoH, we are unable to use the same sort of solution to block DNS requests because the traffic is encrypted and we cannot see what the requested domain. As previously mentioned, this has both security and privacy strengths, but those strengths are also being leveraged by threat actors.

For example, a new variant of PsiXBot malware that focuses on sexploitation and cryptocurrency mining attacks has been seen using Google's public DNS over HTTPS as a new way to resolve it command and control servers (Proofpoint 2019). Specifically, PsiXBot gets it IPs for it CnC servers as a JSON blob using Google's JSON API for DNS over HTTPS format (Gatlan 2019*b*). DoH allows PsiXBot to hide all of its DNS query traffic behind encryption and circumvent any organization security controls against known network IOCs for the malware. Hence, any organization that doesn't have configuration enable to disable DoH traffic from all managed applications and endpoints will be susceptible to this malware, likely go hidden as it moves onto its later stages in the malware lifecycle. Another example of a malware using DoH for getting its CnC server information for later stages in the malware lifecycle is Godlua. Godlua is a Lua-based backdoor malware that primarily targets and infects Linux servers, but Windows machines are also susceptible. It retrieved URLs needed in the later stages of the malware execution via DNS over HTTPS text records of a domain (Gatlan 2019*a*). As previously mentioned, since DoH was used, this malware can also succeed in evading network defenses to successfully execute and not give up it's CnC infrastructure details via DNS packets.

Let's take a look at a couple more real world threats that took advantage of DoH to the next level. In July 2020, researchers from Kaspersky discovered an Iranian threat

group, dubbed OilRig, evolving their tactics to use DoH as a channel for communication and exfiltration. The threat group was observed leveraging an open source tool called DNSExfiltrator2 that uses DoH to move data internally in a compromised network and externally to exfiltrate data to threat actor controlled infrastructure (Quointelligence 2021). Additionally, in late 2020, another threat actor group called ReconHellCat were observed using Cloudflare workers as a C2 interface and performed name resolution via DoH. Cloudflare Workers is a serverless application platform running on Cloudflare's global cloud network that allows users to create an application that can performs function and can be interacted with via API without necessarily having to stand up any dedicated infrastructure (Cloudflare 2021).

The malware examples briefly mentioned above are just a few examples, but they show how malware is evolving their tactics by leveraging DoH for malicious purposes and evading existing network defenses. As wide adoption and popularity of DoH grows, so will the risk that another threat group will add this to their toolbox and attempt to infiltrate your organization and go unnoticed. Hence, new methods of detection for DoH traffic must be researched and implemented to expand network security coverage to DoH traffic.

4. Methods to Detect DNS over HTTPS Traffic

In this section, we will explore some methods proposed in various papers on how to detect DNS over HTTPS traffic. Starting with configuration management, we will move on to explore content/TLS inspection, JA3/S fingerprints, application logging, network traffic analysis (aka network detection and response), RITA, and finally machine learning based detections.

4.1. *Configuration Management*

Methods discussed here are not necessarily detections but focus on managing/maintaining control over the usage of DNS over HTTPS via configuration management.

- **Disable DoH in Manage Endpoints**: Settings for browsers/applications and OSs can be configured centrally and will likely have settings to disable DoH capabilities. Ensuring that centrally managed policies include the configuration to disable DoH and is applied to all the managed endpoints in the environment can alleviate the problem of users using DoH (Hutchison 2021). However, this still leaves unmanaged endpoints to have their own configuration and also cannot enforce policies for malware that has native capabilities to connect outbound via DoH.
- **Block Known DoH Providers**: This method involves blocking all egress traffic to the known public DoH resolvers. Available public lists of DoH resolvers must be monitored regularly, and internal access control lists and firewall rules must be updated to block the outbound connections to those DoH resolvers. This way, the users within the organization are likely not able to use DoH since most configurations depend on publicly available DoH resolvers. However, this method still leave a gap for threat actors to have their own self-host DoH resolvers that they can connect to and avoid the blocklist.
- **Allowlisting**: Another approach to maintaining control over the usage of DoH is to implement an allowlist of applications or public DoH resolvers that are known to the organization and are approved to use for DoH. This method will require a lot of effort to maintain and can be hindrance of UX for the users within the organization (Hutchison 2021).

While these recommendations are fine for mainly regulating normal user usage of DoH, they aren't well suited for blocking malicious use or completely blocking users from using DoH since a lot of it is dependent on making sure the configuration is in place and is up-to-date on a regular basis. Attempting to block all DoH usage should be a last resort control, and more detection based solutions must first be tested and implemented.

4.2. *Content/TLS Inspection*

TLS inspection involves using a middle box that is placed within an organization's network that ingests all HTTPS traffic and decrypts the encrypted traffic to be used for analysis. This decrypted traffic is then forwarded to an internal network security analysis tool/product for analysis and take any actions necessary based on the organization's security policies/controls. Once the traffic analysis is complete, the traffic is then encrypted with a new key and certificate before being routed to its intended destination. For detection DoH traffic, this method can be very advantageous as we can just look for things like 'application/dns-json' or 'application/dns-message' in the contents to determine that this traffic is DoH. From there, we can take the appropriate actions on the traffic packets, like blocking the traffic is it attempting to resolve a known malicious domain.

While this method seems like the best, there are some considerations and disadvantages to take into account before implementing a TLS inspection middle box in an organization. Starting with the fact that a separate TLS inspection appliance must first be purchased and configured in-line within the network to ingest all HTTPS traffic. If this configuration is not carefully done, it can actually lead to more security problems than it solves. A study was conducted, and it showed that 62 percent of the traffic that is redirected through a middle box had their security weakened (Thakkar 2018). Also, with the introduction of TLS version 1.3, new features such as encrypted certificate, no more static cipher suites for RSA and Diffie-Hellman, ability to disable the middle box, and more have been enabled which can make it harder for the middle box to decrypt and inspect HTTPS traffic(Hjelm 2019). Decrypting traffic through a middle box and encrypting it again with different certificates can also cause applications to break because some of them maybe dependent on the originating certificate to function properly.

4.3. *JA3/S Fingerprints*

Developed by a team at Saleforce, JA3 is a method for profiling SSL/TLS clients through a method of fingerprinting. A JA3/S fingerprint is the hash of the following fields gathered from the TLS Client/Server Hello packet: TLSVersion, Ciphers, Extensions, EllipticCurves, and EllipticCurvePointFormats. A list of known good and bad JA3/S fingerprints are publicly available and be ingested as intelligence to be used as part of network security analysis. JA3 can be integrated with Zeek to detect known bad HTTPS traffic by calculating the JA3 fingerprints and comparing them with the list of known bad fingerprints.

This method essentially works like a signature based detection that needs to be constantly updated to effectively work. But even then, it is not effective against novel malware because their server might not have been fingerprinted yet, and we wouldn't have it as part of the known bad fingerprint list to compare to and block the traffic. Proper configuration and ingestion is also needed to use the publicly available lists as part of your traffic analysis process.

4.4. *Application Logging*

The method involves configuring all know applications that natively perform DNS queries to save those queries in a file. The file is then collected and imported into a log exploration tool and analysis is performed. Generally, any DNS requests that doesn't directly relate to the application is considered suspicious and need to be further investigated to determine the classified (malicious or benign) of the domain.

The problem with this approach is that it is more reactionary than a proactive detection method. It also requires that the application has the capability to log DNS queries and is configured to store DNS queries by default or through a centrally managed configuration policy. If all of those are properly setup, there still needs to be a process in place that collects all those DNS files and ingest them into a log exploration tool in order to analyze the DNS entries.

4.5. *Network Traffic Analysis (aka Network Detection and Response)*

"Network traffic analysis can be applied to raw traffic to model normal network behavior and perform non-signature-based techniques to detect suspicious and/or anomalous network activity"(Hutchison 2021). Modeling can be done specifically on just HTTPS traffic to understand the underlying semantics that can be used to tag DoH traffic as an anomaly.

While this method is independent of external source to flag DoH traffic, "The enterprise needs to operate and maintain infrastructure for collection, processing, and analysis; analytic development and testing is often time- and expertise-intensive. This approach requires further research for DoH-specific use cases" (Hutchison 2021). Good modeling also requires a large bake in time and expertise to appropriately tune the anomalies, or it will result in skewed balance of TPs vs FPs.

4.6. *RITA*

RITA, stands for Real Intelligence Threat Analysis, is an open-source tool developed by Active Countermeasures that ingests Zeek logs or PCAPs converted to Zeek logs and performs statistical analysis on the logs, even if they are encrypted, to detect beaconing, DNS tunneling, and traffic to blacklists. RITA doesn't natively support the detection of DoH traffic, however, flags traffic to DoH resolvers as beaconing activity.

One problem with RITA's statistical approach is that it is heavily dependent on the duration time of a connection to detect unusual traffic. Duration time of a connection can be really hard to baseline, especially in the case of DoH traffic to a resolver. It also doesn't support the detection of DoH natively, and have to rely on its beaconing detection pattern to get detections and then manually validate to see if it is DoH traffic or not.

4.7. *Machine Learning*

Machine Learning based detections involve training and tuning one or more ML classifier algorithms with a large dataset that contains HTTPS and DoH traffic captures. The first step to having an ML model that can detect DoH traffic is, gathering a large enough dataset that has a mix of HTTPS and DoH traffic packets. Then that dataset is processed to select important features (in this case, a field in a packet) that play an important role in differentiating the types of traffic. We can see a sample of feature selection in Figure 3 below. Once the feature selection part is completed, the selected features (packet fields) are fed into various machine learning algorithms like Random Forest, Decision Tree, KNN, etc. to classify a certain packet as DoH or regular HTTPS. Then the best performing ML models are selected and tuned until they reach a point of

Category	Feature(s)	Overlap DoH / non-DoH *	Overlap benign / malicious DoH *, **	Useful?
Flow Statistics	Duration	0	0	Yes, browsers keep connection with DoH server open whilst a web page is fetched in a few seconds
	Number of packets in/out	0	-	Yes, DoH flows consist out of more packets than non-DoH flows
	Ration packets in/out	1	-	Yes, DoH always has a ratio close to one due to the DoH protocol
Flow Bytes	Number of bytes sent/received	2	0	Useful for detecting malicious DoH since these flows contain more bytes
	Ratio bytes sent/received	1	-	Yes, DoH always has a ratio close to one
	Rate sending/receiving	2	2	No, too much overlap
Packet Length	Variance packet length in/out	1	1	Yes, benign DoH has a low variance and malicious DoH has a large variance
	Mean packet length in/out	1	1	Yes, benign DoH has a low mean packet length and malicious DoH an average mean value
	Median packet length in/out	0	1	Yes, clear seperation from small to large: non-DoH, malicious and benign-DoH
	Mode packet length	2	0	No, the values for malicious DoH do not overlap due to the capturing method
Packet Time	Mean, median, mode, etc.	1	1	Rather use the correlated Duration feature.
Inter-Packet delay	Min, average, max delay	0	-	Rather use the correlated Duration feature.
	Mean, median, mode, etc.	2	2	No, too much overap since it relates to RTT which is traffic independent
Other	Bursts	0	-	Yes, DoH flows consists of more bursts than non-DoH
	Autocorrelation	2	-	No, too much overlap
	Symetry	2	-	No, too much overlap

FIGURE 3. Sample Feature Selection of Fields in a DNS Traffic Packet.

high accuracy (95%+) in classifying (aka detecting) DoH packets from a mixed dataset of HTTPS and DoH packets.

Some issues with having an ML learning model that detects DoH traffic is it requires maintenance of a large pre-classified network dataset to train the ML model initially. The ML learning model also needs to be self-hosted (aka can get expensive) unless a network security service you're using is already providing this. Finally, it also requires expertise in ML model training and time to tune it to a state where it is classifying the types of traffic with a high accuracy. While these issues are there for this type of detection, I believe this is still the best approach, as it doesn't depend on any signatures or anomalies to detect DoH traffic. Rather, it understands the characteristics of a packet that represents HTTPS traffic and characteristics of a packet that represents DoH traffic.

4.8. *Future Works*

There are several possibilities for future works to improve the way DoH traffic is detected. I believe many of these revolve around machine learning, specifically the deep learning branch. I haven't seen any solutions that leverage deep learning to classify packets as HTTPS or DoH. So, research on detecting DoH traffic using machine learning can be avenue for future works. Then comparing it to see how effective and efficient it is compared to non-deep learning based ML models could insightful. Another avenue for research is, what do we do now that we know this packet is a DoH request? How can we use this information about the packet to analyze it for suspicious behavior or determine it is part of malicious activity.

5. Conclusion

In this paper, we talked about what DNS over HTTPS was and how it is evolving the network security and threat landscape. We understood the differences between DoH and traditional DNS over UDP and what the benefits of each of them were. Then we talked about real world threats that are leveraging DoH to advance their attack tactics and why the need for proper detection of DoH traffic is needed. Finally, we went through several proposed solutions for detecting DoH traffic and understood how each of them work and what some issues with each of the proposed solutions were. Hopefully this paper helped understand the importance of detecting DoH traffic as it is being widely adopted and what the current detection solutions and how you could use them interim while more advanced detection methods are being researched.

REFERENCES

Borgolte, Kevin, Chattopadhyay, Tithi, Feamster, Nick, Kshirsagar, Mihir, Holland, Jordan, Hounsel, Austin & Schmitt, Paul 2019 How dns over https is reshaping privacy, performance, and policy in the internet ecosystem. *Performance, and Policy in the Internet Ecosystem (July 27, 2019)* .

Cloudflare 2021 Cloudflare workers documentation.

Gatlan, Sergiu 2019*a* New godlua malware evades traffic monitoring via dns over https.

Gatlan, Sergiu 2019*b* Psixbot modular malware gets new sextortion, google doh upgrades.

Hjelm, Drew 2019 A new needle and haystack: Detecting dns over https usage. *SANS Institute, Information Security Reading Room, August* .

Hoffman, P. & McManus, P. 2018 Dns queries over https (doh).

Hounsel, Austin, Borgolte, Kevin, Schmitt, Paul, Holland, Jordan & Feamster, Nick 2019 Analyzing the costs (and benefits) of dns, dot, and doh for the modern web. In *Proceedings of the Applied Networking Research Workshop*, pp. 20–22.

Hutchison, Sean 2021 Dns over https: 3 strategies for enterprise security monitoring.

Lu, Chaoyi, Liu, Baojun, Li, Zhou, Hao, Shuang, Duan, Haixin, Zhang, Mingming, Leng, Chunying, Liu, Ying, Zhang, Zaifeng & Wu, Jianping 2019 An end-to-end, large-scale measurement of dns-over-encryption: How far have we come? In *Proceedings of the Internet Measurement Conference*, pp. 22–35.

Proofpoint 2019 Psixbot now using google dns over https and possible new sexploitation module.

Quointelligence 2021 New godlua malware evades traffic monitoring via dns over https.

Siby, Sandra, Juarez, Marc, Vallina-Rodriguez, Narseo, Troncoso, Carmela & others 2018 Dns privacy not so private: the traffic analysis perspective. In *The 11th Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs 2018)*.

Singh, Sunil Kumar & Roy, Pradeep Kumar 2020 Detecting malicious dns over https traffic using machine learning. In *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)*, pp. 1–6. IEEE.

Thakkar, Jay 2018 What is ssl inspection? how does it work?

Van Heugten, JHC 2018 Privacy analysis of dns resolver solutions. *Master's thesis, Master of System and Network Engineering, University of Amsterdam, The Netherlands* .

Vekshin, Dmitrii, Hynek, Karel & Cejka, Tomas 2020 Doh insight: Detecting dns over https by machine learning. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, pp. 1–8.