SANS | GIAC CERTIFICATIONS

WHITE PAPER

A New Needle and Haystack: Detecting DNS over HTTPS Usage

Drew Hjelm

A New Needle and Haystack: Detecting DNS over HTTPS Usage

GIAC (GCIA) Gold Certification

Author: Drew Hjelm
Advisor: David Hoelzer
Accepted: August 26, 2019

Abstract

Encrypted DNS technologies such as DNS over HTTPS (DoH) give users new means to protect privacy while using the Internet. Organizations will face new obstacles for monitoring network traffic on their networks as users attempt to use encrypted DNS. First, the paper presents several tests to perform to detect encrypted DNS using endpoint tools and network traffic monitoring. The goal of this research is to present several controls that organizations can implement to prevent the use of encrypted DNS on enterprise networks.

1. Introduction

Recently there has been an increase in encrypted traffic on the World Wide Web for reasons such as the fear of government and corporate spying, protection of business data from unwanted disclosure, and individuals' desires to prevent malicious actors from eavesdropping on online activity. Conservative estimates from Sandvine, a network solutions provider for carrier networks, show that Internet users encrypt more than half of Internet traffic (2018). Fortinet (2018), the security hardware and software provider, found that Internet users encrypted 72% of traffic in 2018 (up from 55% in 2017). Let's Encrypt (2019) showed that 78% percent of pages loaded in Firefox were over HTTPS compared to 29% of pages in April 2014. Whether you look at traffic over the wire or on the endpoint, the data show more users are accessing the Internet using encrypted methods.

One significant source of non-encrypted traffic is still the Domain Name Service (DNS) protocol. The volume of DNS traffic may not be substantial because requests contain relatively few bytes of data. However, viewing DNS traffic can give insights into what people are doing on the Internet: browsing to websites, consuming streaming media, and even visibility into malware. While the amount of DNS data is small in terms of bytes, the value of the DNS data can be quite great.

In recent years, the IETF approved two methods of encrypted DNS traffic to improve the privacy of domain name lookups using public recursive DNS resolvers. Encrypted DNS protocols such as DNS over HTTPS (DoH) and DNS over TLS (DoT) give users new means to protect privacy while using the Internet. The added privacy afforded to users can create challenges for organizations trying to protect their environments from malware or malicious insiders trying to steal data. This paper proposes methods for detecting encrypted DNS using host-based and network-based monitoring. The research proposes several controls that organizations can implement to prevent the use of encrypted DNS on enterprise networks.

Drew Hjelm, drew@vets.io

2. Protocol Explanations

Before describing challenges posed by encrypted DNS protocols, the traditional DNS protocol and its uses in traffic monitoring will be explored. The IETF (Mockapetris, 1987) describes domain names as mechanisms to make computer resources more usable for users. Computers resolve domain name resources by querying name servers to map names to resource records. Resource records typically hold IP addresses (A/AAAA records) or information about where to find an IP address (CNAME records). Domain name resolution is a recursive process. Figure 1 depicts the DNS resolution process. An application will query a local resolver (Step 1), which will then query recursive resolvers until a name server returns the desired domain name resource (Steps 2-4). The DNS record is returned to the application through the recursive lookup servers. Once the application has the results of the DNS query, the application may browse to the web server (step 5).

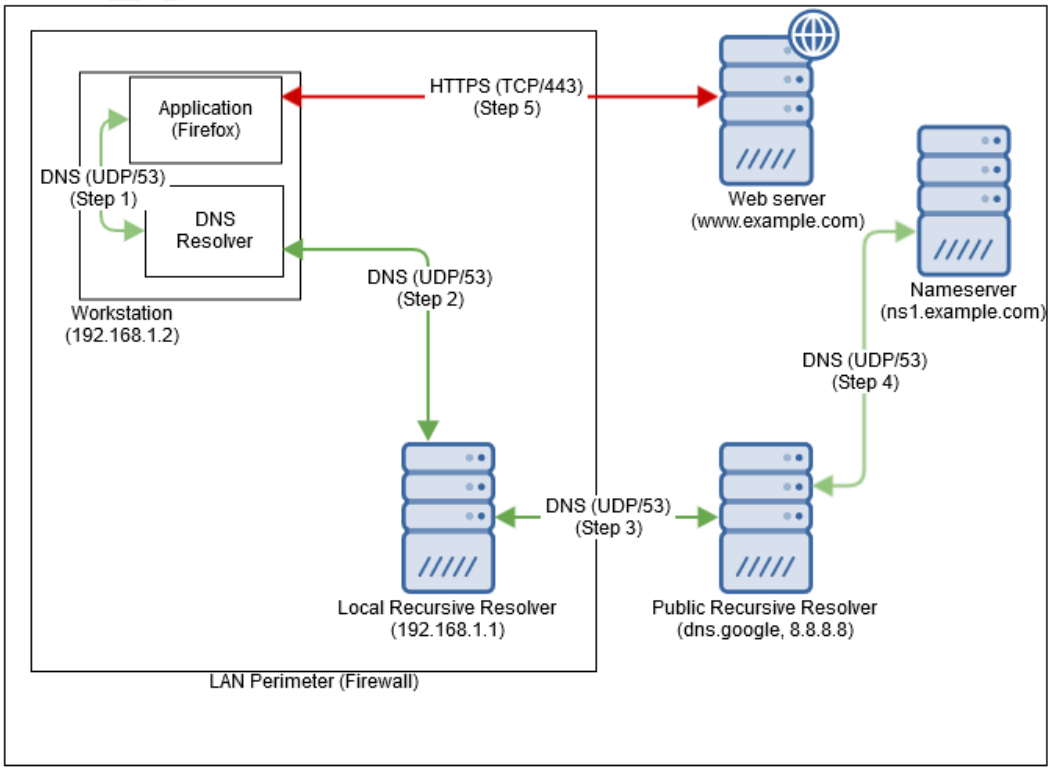

Figure 1: DNS Resolution

While a primary use for DNS is to resolve Internet resources to IP addresses, attackers will also use DNS for malicious purposes. Farnham (2013) describes the use of DNS as a tunneling mechanism to hide traffic using TXT records. Attackers may also create domains to make malware or phishing websites appear less malicious than they are. Attackers can even program domain names as kill switches for dangerous worm software, such as the WannaCry worm released in 2017 (Lee et al., 2017).

Organizations can easily monitor DNS traffic based on many characteristics of the protocol. DNS is a protocol that transmits data in plain text over either TCP or UDP port 53 (Mockapetris, 1987). The DNS resolution protocol appears in a defined format, per IETF RFC 1035. Since DNS is a plaintext protocol, organizations can get an idea of where even encrypted traffic is going by passively observing DNS queries on the wire. If the organization is capturing DNS queries, the organization should be able to determine which endpoints are connecting to servers mapped to malicious domains.

There are several well-known methods for watching traditional DNS queries noted in RFC 7626 (Bortzmeyer, 2015) and other sources. Inline traffic monitoring on a firewall or other network capture device can capture DNS queries. Organizations can enable logging on internal DNS resolvers. Endpoint detection and response (EDR) or other host-based logging programs (such as Microsoft Sysmon) can log DNS queries on the originating host (Kennedy, 2019; Draeger, 2019). Organizations typically deploy a mix of network-based and endpoint-based tools to log queries and alert on suspicious or malicious DNS traffic.

2.1. DNS over HTTPS

To address rising privacy concerns related to DNS, the IETF approved standards like DNS over HTTPS (DoH) and DNS over TLS (DoT) to allow for the transmission of DNS information using encryption. According to RFC 8484 (Hoffman & McManus, 2018), the use of encrypted traffic over the standard HTTPS port, TCP/443, "can deter unprivileged on-path devices from interfering with DNS operations and make DNS traffic analysis more difficult." Figure 2 demonstrates how an application can bypass

traditional DNS architecture using DoH. In contrast to Figure 1, the application can make a DNS request directly to a public DNS resolver (Step 1), and the public resolver will query a nameserver for a DNS record (Step 2) and return that information to the application. Once the application has the DNS response, it can then access the web server by IP address (Step 3). The DoH lookup process removes the need for the local workstation DNS resolver and the organization's local recursive resolver to query DNS records for applications using DoH. This means those system processes will no longer log DNS requests for DoH-enabled applications.

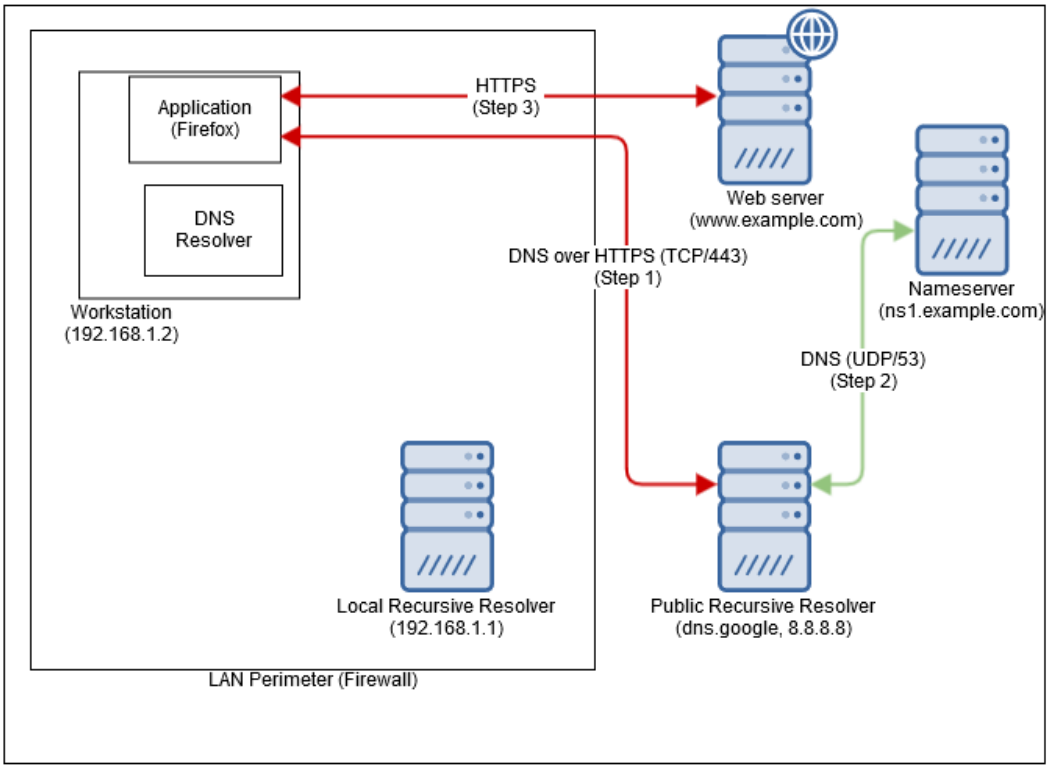

Figure 2: DNS over HTTPS Resolution

Public DNS recursive resolvers implement and support several implementations of DoH. RFC 8484 and RFC 1035 define a DNS wire format that Google and Cloudflare support. Meanwhile, Google (2019) created a DoH implementation that allows for query and response over JSON.

Organizations will need to change how they are monitoring network activity in the future because of DNS over HTTPS along with the overall increase in encrypted traffic. DNS traffic can now blend in with other HTTPS traffic originating from an application rather than allowing passive collection of DNS queries. Since applications can make DNS queries over HTTPS, threat analysis based on the plaintext content of DNS queries will require investment in TLS inspection infrastructure. Absent TLS inspection, organizations will need to use metadata or other information about network flows to identify web traffic or malware when they could previously use DNS as an indicator. For example, analysts can search for web traffic going to IP addresses without previously logged DNS lookups to determine whether it is using another form of domain name resolution. Organizations can create profiles of encrypted connections, called JA3 fingerprints, to identify known-good and malicious applications (Salesforce, 2019). The changes in traffic patterns will require organizations use tools and techniques able to monitor these newly relevant forms of metadata.

2.2. DNS over TLS

In contrast to DoH, systems resolve DNS over TLS (DoT) on a protocol and port that are separate from other transmission mechanisms, making its control by network operators much simpler. While DoH transmits DNS queries over TCP port 443, DoT must use TCP port 853 as its method for performing name resolution by RFC 7858 (Hu et al., 2016). A network operator could easily block port 853 traffic to prevent the use of DoT on a network. Figure 3 shows that DoT can function similar to traditional DNS (Figure 1) with the exception of running over TCP port 853 for a portion of the resolution process.

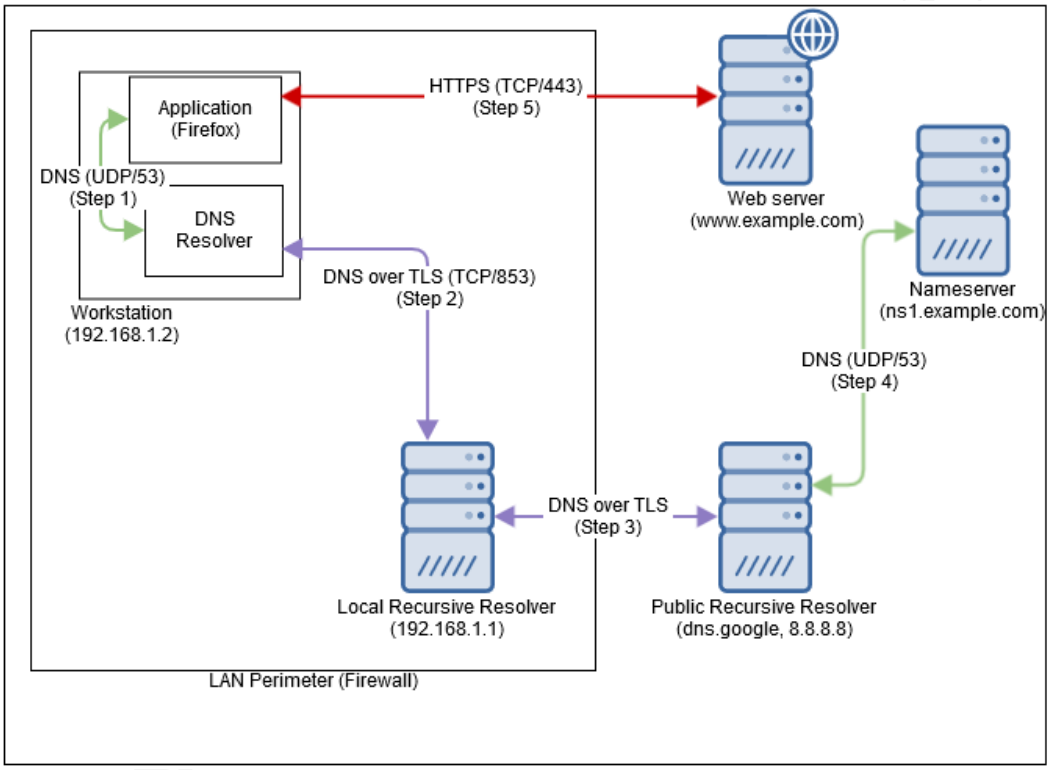

Figure 3: DNS over TLS Resolution

The separation of network traffic and DNS traffic contained within the DoT specification does have support among some experts in the information security community. Paul Vixie (2018), an architect of several DNS protocol extensions and the author of the BIND DNS application, argued that "DoH is an over the top bypass of enterprise and other private networks. But DNS is part of the control plane, and network operators must be able to monitor and filter it. Use DoT, never DoH." For organizations considering the implementation of encrypted DNS, using DoT may better fit into the organization's architectural strategy than DoH.

In contrast to current DoH application proofs-of-concept that can natively query the DNS records, typically DoT resolution requires another program. Local DNS resolver applications that support DoT include BIND and Unbound, as well as newer versions of Android operating systems.

2.3. Public Threats from Encrypted DNS

Organizations need to start evaluating the risk associated with the DoH protocol because attackers have already begun using DoH to look up command-and-control (C2) servers. The best-known example of DoH as a C2 mechanism came in April 2019 with the Godlua backdoor (360 Netlab, 2019). A newer variant of the Godlua backdoor runs on Linux and Windows and uses a DoH request to grab a part of its C2 information.

Another way an attacker could use DoH in an attack is to trigger a redirected webpage as part of a spam campaign. Researchers at MyOnlineSecurity (2019) found a sample where an email attachment had a Base64 encoded string that would query Google DoH for a TXT record. The TXT record would have a JavaScript redirect to a spam webpage whose address often changed.

Numerous DoH C2 proofs-of-concept are publicly available, meaning that the threat of malicious actors using DoH is likely to increase soon. Organizations such as Sensepost (2019) and Spider Labs (2018) along with individual contributors like Magisterquis (2019) have all made proof-of-concept DoH C2 or botnets available. With several examples of attacks using DoH in addition to the ability of multiple researchers to build tools to leverage DoH for malicious purposes, attackers have several methods they can use to conduct future attacks.

As of this writing, there was less risk posed by DoT as a malicious vector than DoH. First, information security news outlets have not widely reported the use of DoT-based malware using TCP port 853. Second, network administrators or other personnel can easily block DoT because it uses a single well-known port. Malicious activity using DoT may be a future risk, but the current threat is not high. The broader adoption of DoH by both legitimate application developers and malware authors means that research into DoH is a higher priority than DoT. For this reason, the following study will focus on indicators of DoH in network traffic captures and endpoint logs.

3. Testing

3.1. Lab Configuration

This study consisted of several tests run against a lab environment with basic segmentation and instrumentation to test the effectiveness of DoH at bypassing controls. The lab included a Windows domain (Windows 2012 R2 Domain Controller, 1 Windows 10 PC, 1 Windows 7 PC), a pfSense firewall, and a Security Onion virtual appliance. The Security Onion ingested traffic from the domain network segment. An attacker workstation running Kali Linux also existed in the lab, which connected to an EC2 instance running as a nameserver for a DoH C2 server (GoDoH). The public recursive DNS resolver for the lab is Quad 9 (9.9.9.9, from https://quad9.net/), which is a free recursive resolver with added malware protection. DHCP leases for the workstations directed them to use the domain controller as their DNS resolver, which sent requests to the firewall and then to Quad 9. Malware also used Quad9, but the unfiltered version (9.9.9.10). Figure 4 shows the various components of the lab network.

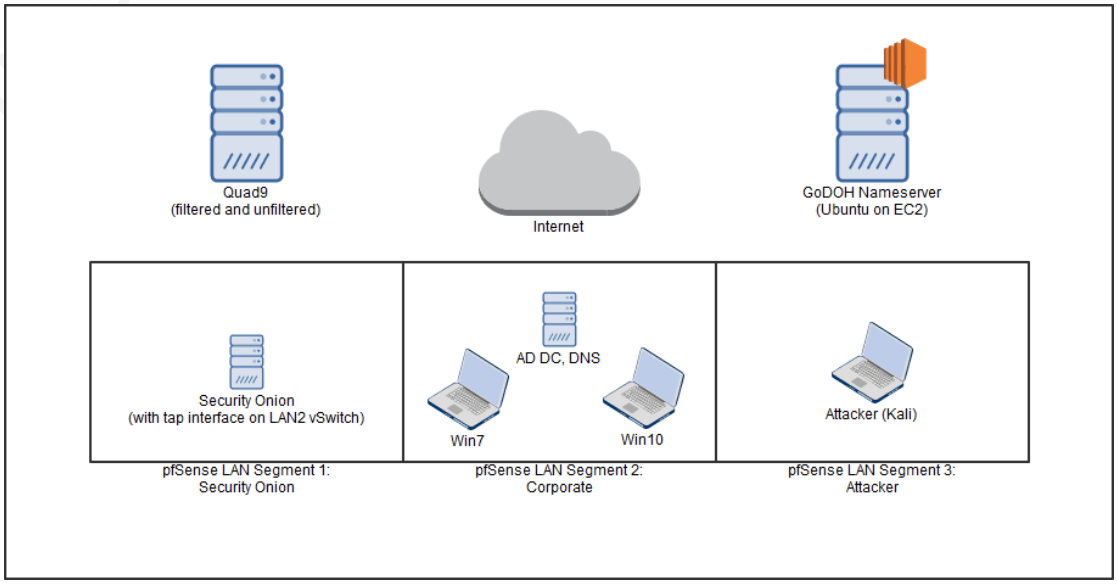

Figure 4: Lab Setup

In addition to configuring Security Onion as an IDS, the lab systems had other instrumentation and tools. Workstations ran Sysmon, Mozilla Firefox 67, Visual Studio Code, and Notepad++. The latest version of Sysmon includes DNS logging (Russinovich,

2019). The researcher used SwiftOnSecurity (2019) Sysmon rules as a base configuration, with added rules added for DNS Query log ingestion. The researcher upgraded the Sysmon rule schema to Version 4.21 and added the following lines:

```
<DnsQuery onmatch="include">
    <Image condition="image">firefox.exe</Image>
</DnsQuery>
```

Sysmon stores all its logs, including DNS logs, in the Windows Event Log. Sysmon logs are under Applications and Services\Windows\Sysmon\Operational. The new DNS logs in Sysmon are Event ID 22. Figure 5 shows a sample Sysmon DNS log event.

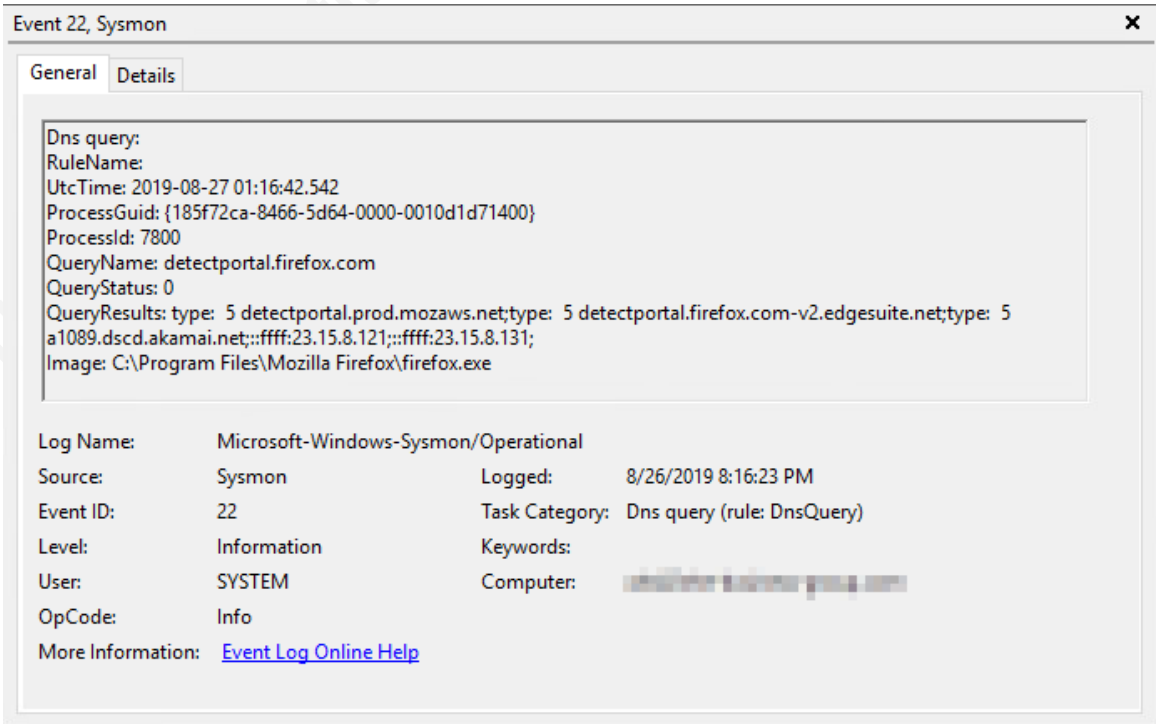

Figure 5: Sysmon DNS Log Sample

3.2. Web Browsing with DoH

The first baseline test to determine the effectiveness of DoH to circumvent controls consisted of web browsing for five minutes with Mozilla Firefox through various sites. The web browsing consisted of clicking through a variety of links on popular news

and entertainment sites. The researcher chose Firefox for this test because it already had a documented DoH resolver (Mozilla [3], 2019) while Google had not yet added DoH to Chrome (Chromium, 2019). On the Windows 10 workstation, the researcher flushed the DNS resolver cache before each test. Clearing the cache ensured that the workstation would not be using cached records and it also served as good delineation points in the Sysmon log. For the test with DoH disabled, the researcher used Firefox normally without configuration changes. For the DoH test, the researcher configured Firefox with the following changes made in about:config (Mozilla [3], 2019):

```
network.trr.mode = 3
network.trr.bootstrapAddress = 1.1.1.1 (Cloudflare DNS)
network.trr.uri = https://mozilla.cloudflare-dns.com/dns-query
```

After browsing with these configurations for five minutes, the researcher reviewed data from Sysmon logs in Windows Event Viewer and the Security Onion Kibana instance that aggregated Zeek logs. Table 1 shows the results of the Baseline test.

Test	Sysmon DNS Events	Zeek DNS Queries	Zeek HTTP Logs	Zeek SSL Logs
DoH Disabled	5142	5560	325	2154 (449 TLS 1.3)
DoH Enabled	0	848	530	2747 (499 TLS 1.3)

Table 1: DNS over HTTPS Baseline Test

The most obvious difference in the test was the absence of DNS logs on the endpoint during the DoH test, but the decrease in DNS activity in the test was also noticeable. In this case, Firefox completely bypassed DNS resolution on the OS. The OS could not log DNS queries when DoH was enabled. Since the OS did not perform DNS resolution over UDP port 53 for resolving domain names in Firefox, the Security Onion did not capture DNS queries in its packet captures and the Zeek logs did not contain as

many DNS queries. The majority of the remaining DNS queries in the Zeek logs came from other Windows 10 processes such as Windows Update and Microsoft Telemetry. There was an increase in Zeek SSL logs, as Firefox was performing DNS lookups over HTTPS (Zeek SSL logs).

The fact that DoH completely removed the ability of a well-known endpoint detection tool from seeing DNS queries from a process introduces a monitoring gap for organizations. An attacker who builds DoH malware would completely circumvent some types of endpoint detection controls. They can also remove the ability of defenders to match on static indicators of compromise such as domain names. Defenders could rely on network traffic capture and TLS inspection to decrypt and log DNS requests. TLS inspection may interfere with other operations and may become unwieldy as TLS 1.3 adoption becomes more widespread (See *Section 4.1: TLS Inspection*).

3.3. DNS Command-and-Control over DoH

This research also includes sample traffic from a DoH Command-and-Control tool to simulate what an attacker might use. In this case, the simulated attacker used GoDoH by Leon Jacobs at Sensepost (2018) in the second baseline test.

GoDoH is a simple DoH proof-of-concept (POC) tool written in Go with an agent and server application. The tool is cross-platform and supports Windows, Linux, and Mac OSX, as well as several public DoH providers by default, including Google, Cloudflare, and Quad9. GoDoH (like other DNS C2 servers) uses DNS TXT records to pass information between agents and the server.

Configuring a GoDoH server is similar to configuring another DNS nameserver. In this case, the researcher registered the domain playingwithgodoh.xyz on Namecheap. Meanwhile, the researcher configured a t2.micro EC2 instance on Amazon Web Services using an Ubuntu 18.04 AMI. The researcher used the following configuration to run a DNS C2 name server on AWS:

- Create a Security Group to allow UDP inbound from the Internet

- On Ubuntu 18.04, the systemd-resolved stub listener needs to be disabled to allow the GoDoH DNS resolver to bind to port 53. Modify /etc/systemd/resolved.conf at the line `DNSStubListener=Yes` to `DNSStubListener=no`
- Download and build GoDoH from https://github.com/sensepost/goDoH
- Configure the server to allow GoDoH to bind to port 53 as a low-privileged user.

```
sudo setcap 'cap_net_bind_service=+ep' ~/godoh-linux64
```

- Run the GoDoH server (Figure 6), specifying the C2 domain for the test. Once the server is started and an agent connects, run sample C2 activity.

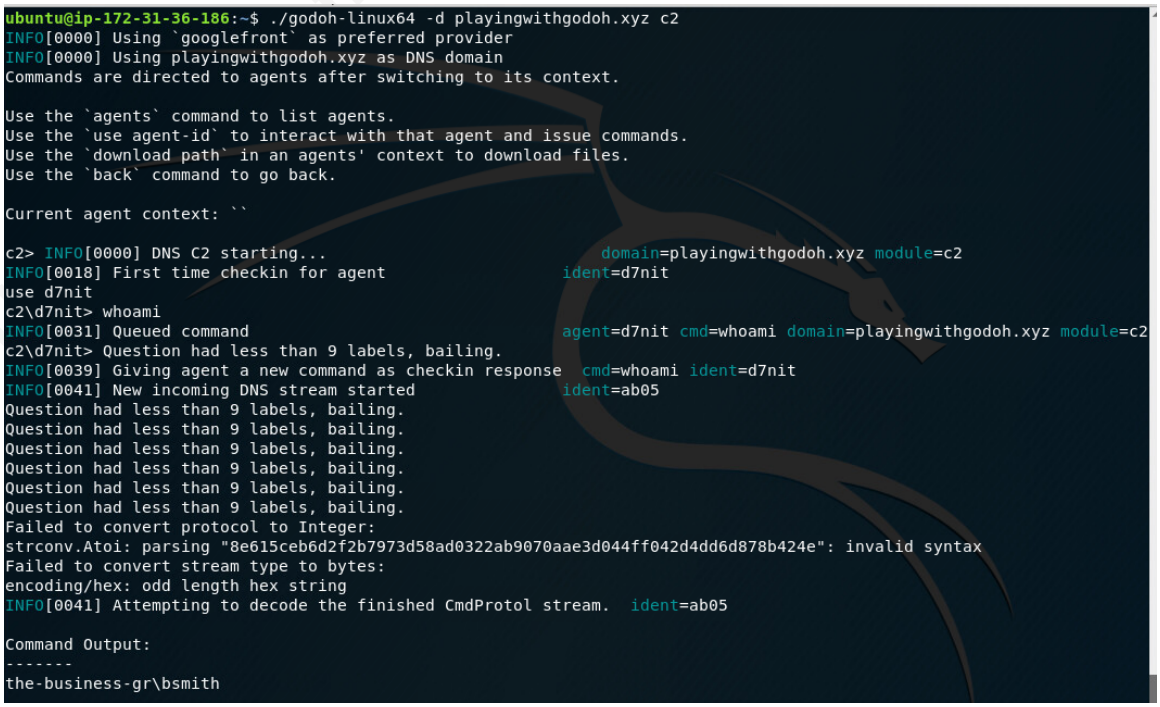

Figure 6: GoDoH Server

- Run the GoDoH agent from a victim computer (Figure 7), then view C2 commands from the server:

```
PS C:\Users\bsmith\Downloads> .\godoh-windows64.exe -d playingwithgodoh.xyz -p quad9 agent
INFO[0000] Using `quad9` as preferred provider
INFO[0000] Using playingwithgodoh.xyz as DNS domain
INFO[0000] Starting polling...                           ident=d7nit module=agent
INFO[0031] Got command data to execute, processing       cmd-data=1f8b08000000000002ff32606c7872a7eedd6bdd9b0cf189474e9f
5b1575f46fc559cdfd4b9b01010000ffffe8743d8d1a000000 ident=d7nit module=agent
INFO[0031] Decoded command                               cmd=whoami ident=d7nit module=agent
INFO[0031] Command interpreted as OS command             ident=d7nit module=agent
INFO[0031] Sending command output back                   cmd-output-len=24 ident=d7nit module=agent request-count=4
INFO[0033] Successful request made                       ident=d7nit labels=ab05.be.0.00.1.0.0.0.0 module=agent response
=1.1.1.1
INFO[0033] Successful request made                       ident=d7nit labels=ab05.ef.1.6b51b634.1.3.1f8b08000000000002ff0
08e0071ffce6c1d531e8d645dc87e1cbd8f0235.af27996735d60a43557cf89d939405095bcbab4a1ece1e2cf11a4bf12463.27a47eefd005bd2ad39
a7d61110ece7f90edd1e452877b0f9b799f26cc89 module=agent response=1.1.1.1
INFO[0033] Successful request made                       ident=d7nit labels=ab05.ef.2.3ab0cf91.1.3.8e615ceb6d2f2b7973d58
ad0322ab9070aae3d044ff042d4dd6d878b424e.da109b779b3311299163fdc9440e128b2648f00c63db12b4d47e0b33956a.9c42602c5247ef01000
0ffff0f5bb6c48e000000 module=agent response=1.1.1.1
INFO[0033] Successful request made                       ident=d7nit labels=ab05.ca.3.00.1.0.0.0.0 module=agent response
=1.1.1.1
```

Figure 7: GoDoH Client

4. Additional Methods to Detect Encrypted DNS

4.1. TLS Inspection

Organizations can use TLS inspection to analyze encrypted sessions, including DoH, to find malicious traffic. TLS inspection is the process of using a middlebox to decrypt communications encrypted with TLS or SSL protocols for analysis purposes. In many implementations, the middlebox intercepts encrypted traffic, decrypts it, and presents the traffic for analysis. The middlebox then encrypts the traffic again with a new key and certificate that the endpoint that initially started the session trusts.

The Internet is moving to adopt TLS Version 1.3, which introduces significant new privacy features which will hinder the effectiveness of TLS inspection technologies. Ian Levy (2018), technical director of the UK Government's National Cyber Security Centre, warned that TLS 1.3 makes traffic inspection harder for enterprises because "it's impossible to whitelist sites anymore because server certificates (the things that authenticate a site) are encrypted." Another enhancement to TLS 1.3, encrypted Server Name Indication (eSNI), will further obscure the destination of encrypted traffic by hiding the domain name in the SNI field of the ClientHello request during the TLS handshake (Ghedini, 2018). Organizations with policies prohibiting the decryption of specific traffic, such as a privacy policy preventing the organization from inspecting employee visits to healthcare websites, may find the inability to selectively decrypt TLS 1.3 traffic as problematic for the implementation of TLS inspection technology.

Drew Hjelm, drew@vets.io

Inspecting TLS traffic may also break some applications dependent on using specific certificates to secure end-to-end connections (certificate pinning). Some organizations will view these TLS inspection challenges as tough to manage without downgrading traffic to TLS 1.2, which would abandon security improvements provided with TLS 1.3 (Du Toit, 2017). The various security and privacy concerns with inspecting TLS 1.3 traffic require organizations to carefully consider how they will implement TLS inspection.

Because of upcoming changes to TLS that would affect TLS inspection, the research will consider other methods for detecting and identifying encrypted DNS traffic. The current state of TLS inspection should be sufficient for allowing organizations to inspect encrypted DNS traffic and alert on traffic encrypted with TLS 1.2. TLS inspection may become unwieldy in the future for some organizations, especially given the advances in investigating metadata of encrypted traffic.

4.2. Application Logging

If an application can make DNS queries over an encrypted DNS channel, configuring the application to save the DNS to a log file can give organizations another tool to analyze information. Organizations can save these log files locally or ingest the logs into a Security Information and Event Monitoring (SIEM) solution through tools like NXLog. These log files allow an organization to both examine known traffic and find which traffic flows do not have DNS information attached to them. In other words, if an organization can determine which application made the DNS request, they can rule that application out as a source for unknown or unusual traffic.

4.2.1. Mozilla Firefox

Mozilla published a guide to debugging Firefox on Windows, Linux, and Mac OS X showing how to use environment variables to enable logging. One configurable flag will log DNS queries made by Firefox, even for DoH (Mozilla [2], 2019). Organizations can configure endpoints running Firefox to log DNS queries on Windows with the setx command.

```
setx MOZ_LOG timestamp,rotate:200,nsHostResolver:4
setx MOZ_LOG_FILE C:\Logs\%USERNAME%-Firefox-DNS-log.txt
```

Figure 8 shows an example log file from the Firefox Trusted Recursive Resolver (TRR), which is the process Firefox uses to query DNS over HTTPS.

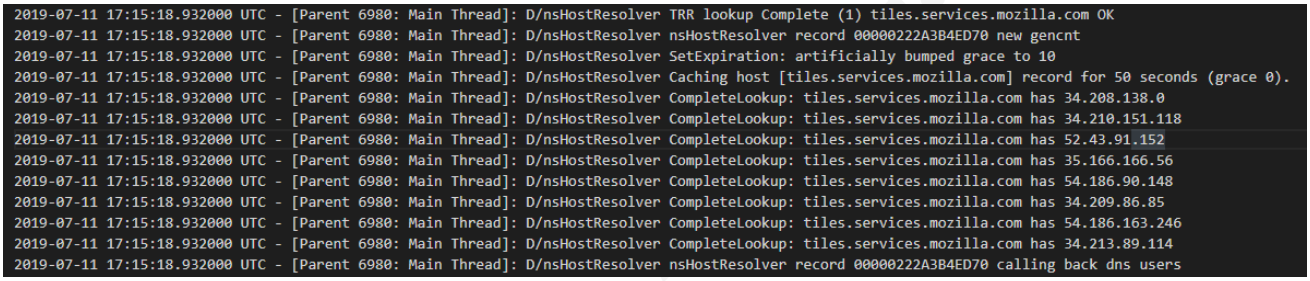

Figure 8: Firefox DNS Resolver Log

4.3. Zeek

The test cases in this paper present two different challenges for finding data in Zeek logs: web browsing and C2. Zeek is an open-source Intrusion Detection System (IDS) previously known as Bro. Zeek ingests pcap files and creates log files based on the network traffic. Analysts can interpret these log files with native tools such as bro-cut. Analysts can also send network logs into other tools for analysis such as a SIEM (such as Elasticsearch-Logstash-Kibana on Security Onion) or RITA.

4.3.1. Web browsing using DNS over HTTPS

Analysts should consider searching Zeek logs to find workstations browsing the web without using defined DNS infrastructure. For example, Zeek logs can show all the known DNS queries made on a network:

```
cat dns.log | bro-cut answers | tr , '\n' | sort -V | uniq
```

The information from this list would give an analyst all of the results of DNS lookups known on the network. The analyst could then use this against a list of SSL and HTTP sessions to determine if any endpoints made those requests to hosts not known by DNS. If endpoints made HTTP and SSL requests to hosts missing from the DNS log, they could be results of DoH queries.

```
cat ssl.log | bro-cut id.resp_h | sort -V | uniq
```

Drew Hjelm, drew@vets.io

```
cat http.log | bro-cut id.resp_h | sort -V | uniq
```

Analysts can also query Zeek logs for the existence of connections to various DoH providers. The following Zeek command would show connections to Quad 9, Google, and Cloudflare:

```
cat conn.log | bro-cut -d ts id.orig_h id.resp_h id.resp_p
proto service duration | grep
"9\.9\.9\.9\|9\.9\.9\.10\|149\.112\.112\.112\|149\.112\.112\.9\|1
49\.112\.112\.10\|8\.8\.8\.8\|8\.8\.4\.4\|1\.1\.1\.1\|1\.0\.0\.1\
|104\.16\.249\.249\|104\.16\.248\.249"
```

4.3.2. DoH Command and Control Detection

Analysts can use the Zeek conn.log to find various types of C2 communications. For example, a long-running connection would show a constant connection and data exfiltration, where many short-running connections at various intervals can show beaconing activity. In contrast to typical DoH requests, the connections to Quad9 by the GoDoH were much longer (Figure 9). The longer connections show that an agent connected to a server for longer C2 sessions, as opposed to an agent performing beaconing or a process simply performing DoH queries.

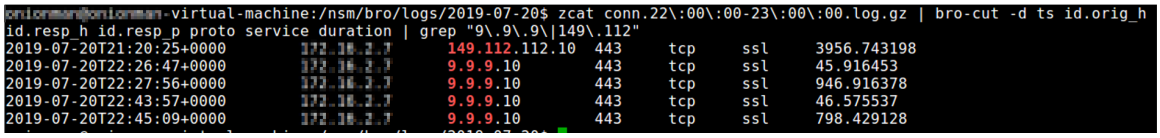

Figure 9: DoH C2 in Zeek

4.3.3. Application Fingerprinting

Zeek also parses out the JA3 and JA3S fingerprints made during an SSL/TLS session. JA3 is a methodology designed by Salesforce (2019) to identify known and unknown programs by their SSL/TLS handshake characteristics. Even though IP addresses can change, it is more difficult for attackers to change how their malware makes connections. Defenders can use the JA3 fingerprint of programs present in Zeek logs to find unique programs. In this case, GoDoH has a distinctive JA3 fingerprint: d3e1de2ca313c6c0a639f69cc3e924a4, shown in Figure 10.

Drew Hjelm, drew@vets.io

```
[redacted]virtual-machine:/nsm/bro/logs/2019-07-20$ zcat ssl.22\:00\:00-23\:00\:00.log.gz | bro-cut -d ts id.orig_h id.resp_h server_name ja3 ja3s | grep quad9
2019-07-20T22:26:47+0000        [redacted]      9.9.9.10        dns10.quad9.net d3e1de2ca313c6c0a639f69cc3e924a4        04b04ede2cbc37f957aeecab9ff8049a
2019-07-20T22:27:56+0000        [redacted]      9.9.9.10        dns10.quad9.net d3e1de2ca313c6c0a639f69cc3e924a4        04b04ede2cbc37f957aeecab9ff8049a
2019-07-20T22:43:57+0000        [redacted]      9.9.9.10        dns10.quad9.net d3e1de2ca313c6c0a639f69cc3e924a4        04b04ede2cbc37f957aeecab9ff8049a
2019-07-20T22:45:09+0000        [redacted]      9.9.9.10        dns10.quad9.net d3e1de2ca313c6c0a639f69cc3e924a4        04b04ede2cbc37f957aeecab9ff8049a
2019-07-20T22:58:41+0000        [redacted]      149.112.112.10  dns10.quad9.net d3e1de2ca313c6c0a639f69cc3e924a4        04b04ede2cbc37f957aeecab9ff8049a
```

Figure 10: DoH C2 - JA3 Fingerprints

4.4. RITA

Analysts can use tools that interpret encrypted traffic for statistical patterns as a means for finding encrypted DNS traffic. RITA, by Active Countermeasures, is an open-source tool capable of showing unusual activity, such as beaconing, DNS tunneling, and traffic to blacklists (Active Countermeasures, 2019). Organizations do not need to decrypt traffic for RITA to analyze the traffic.

Tools that interpret encrypted traffic for long-term activities such as beaconing require collecting full packet capture or interpreting network flows. RITA consumes Zeek (Bro) logs and analyzes them for statistical patterns. Analysts can use a Network Intrusion Detection System like Security Onion to capture traffic and parse the traffic with Zeek. The analysts can then analyze the Zeek logs with RITA to find less-obvious malicious traffic such as beaconing. They can use RITA to analyze the Zeek logs even if the sample is encrypted traffic.

In the lab set up for this exercise, the researcher installed RITA on the Security Onion server to determine if RITA could detect DoH traffic. Using a browsing test in Firefox like earlier, RITA identified the DoH requests to Cloudflare (1.1.1.1) as a form of beaconing traffic (Figure 11). In a production configuration, an organization can configure a server running RITA to send its analysis of beacons and DNS tunneling to a SOC or SIEM regularly using cron jobs.

SCORE	SOURCE IP	DESTINATION IP	CONNECTIONS	AVG BYTES	INTVL RANGE	SIZE RANGE	TOP INTVL	TOP SIZE	TOP INTVL COUNT	TOP SIZE COUNT
0.999	[redacted]	1.1.1.1	2211	5731	127	1870592	1	769	1074	2146
0.748	[redacted]	9.9.9.9	21	166	306	3	1	84	5	10
0.667	[redacted]	23.52.167.93	23	40818	1	13162	1	132	4	2
0.633	[redacted]	72.167.239.239	36	3116	324	202	1	624	8	12
0.577	[redacted]	69.147.92.11	49	11035	356	5992	1	1425	6	10
0.509	[redacted]	72.21.91.29	28	6252	416	15286	1	172	2	4

Figure 11: RITA Beaconing Analysis

For the DNS C2 over DoH test using GoDoH, RITA listed the C2 sessions as long-running connections. Long connections may not necessarily be typical behavior for malware beaconing, but in this case, the GoDoH connection stayed open during the extent of testing (Figure 12).

SOURCE IP	DESTINATION IP	DSTPORT:PROTOCOL:SERVICE	DURATION
[redacted]	149.112.112.10	443:tcp:ssl	3956.74s
[redacted]	172.217.5.226	443:tcp:ssl	1015.96s
[redacted]	172.217.164.130	443:tcp:-, 443:tcp:ssl	1014.59s
[redacted]	9.9.9.10	443:tcp:ssl	946.916s
[redacted]	173.241.242.220	443:tcp:ssl	942.913s
[redacted]	152.195.32.112	443:tcp:ssl	905.504s
[redacted]	34.218.190.17	443:tcp:-, 443:tcp:ssl	900.376s

Figure 12: RITA Long Connections

RITA can also help with analyzing unusual user agent strings, which it parsed from the dataset using the JA3 fingerprints. RITA used the JA3 data from the ssl.log in its analysis of user-agents and found the frequencies of fingerprints. The GoDoH JA3 fingerprint was present six times in the data analyzed by RITA as seen in Figure 13.

```
[redacted] virtual-machine:/nsm/bro/logs/2019-07-20$ rita show-useragents dataset1 -H
```

USER AGENT	TIMES USED
b20b44b18b8[redacted]773e921b03422	709
334da957304[redacted]063e36bc90a47	506
Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:68.0) Gecko/20100101 Firefox/68.0	128
d470a3fa301d80227bc5650c75567d25	35
ce5f3254611a[redacted]d821d44539877	20
e70e3fa3254b[redacted]96f17ec5dfb16	19
10ee8d30a5d0[redacted]d7b2b205facc4	15
bd0bf25947d4[redacted]0424edf4db9ad	10
d3e1de2ca313c6c0a639f69cc3e924a4	6
839bbe3ed07fe[redacted]aaf714d6842	2
28a2c9bd18a11[redacted]5a160da29e4	2
a0e9f5d64349f[redacted]781f81f42e1	2
Microsoft BITS/7.8	1
3b5074b1b5d0[redacted]f69f9f700ff0e	1
839868ad711d[redacted]d37a87f14740d	1
f557b8e5f8ed[redacted]bfb8efb300a94	1
fb6542ef0d88[redacted]3a58b521d6912	1

Figure 13: RITA - User Agents and JA3 Fingerprints

Drew Hjelm, drew@vets.io

5. Encrypted DNS Mitigation Recommendations for Organizations

5.1. Block Egress to Well-Known Encrypted DNS Providers

Organizations should configure firewalls, routers, or other switching devices to block access to publicly known encrypted DNS providers. Network administrators will find blocking DoT easier to implement than blocking DoH because they must only block outbound access to TCP port 853. DoH will be more difficult to block because it uses a well-known port carrying HTTPS traffic. Admins may also find DoH to be more difficult to block because it would be possible to create a new public DoH resolver at any IP address. Attackers or other users of DoH may try to obfuscate their queries even further with techniques such as domain fronting. According to MITRE (2018), domain fronting is a technique to "obfuscate the intended destination of HTTPS traffic or traffic tunneled through HTTPS." Several proof-of-concept DoH C2 applications, such as GoDoH and dnsbotnet, include the ability to user domain fronting for DoH queries. For example, GoDoH allows a user to send DoH queries to https://www.google.com/resolve which then are forwarded to https://dns.google.com/resolve because the application added the latter FQDN to its HTTP Host header. Organizations should not treat blocking known DoH providers as an all-encompassing control but as a first step toward detecting and blocking encrypted DNS.

Organizations should monitor lists of public DoH resolvers. They should update their network access control lists and firewall rules to block outbound connections to well-known DoH resolvers. John Bambenek of Bambenek Consulting (2019) published a list of public encrypted DNS resolvers on GitHub. IP addresses (IPv4 and IPv6) and FQDN of several public DoH resolvers include:

Service	Domain	IPv4	IPv6
Quad9	dns.quad9.net, dns9.quad9.net, dns10.quad9.net	9.9.9.9, 9.9.9.10, 149.112.112.112, 149.112.112.9, 149.112.112.10	2620:fe::fe, 2620:fe::fe:9, 2620:fe::9, 2620:fe::fe:9, 2620:fe::10, 2620:fe::fe:10

Service	Domain	IPv4	IPv6
Google	dns.google, dns.google.com	8.8.8.8, 8.8.4.4	2001:4860:4860::8888, 2001:4860:4860::8844
Cloudflare	cloudflare-dns.com, mozilla.cloudflare-dns.com	1.1.1.1, 1.0.0.1, 104.16.249.249, 104.16.248.249	2606:4700:4700::1111, 2606:4700:4700::1001, 2606:4700::6810:f9f9, 2606:4700::6810:f8f9

Table 2: Public DNS over HTTPS Recursive Resolvers

5.2. Application Whitelisting and Configuration Standards

As more developers begin to add encrypted DNS capabilities to their applications, organizations should be evaluating applications for their encrypted DNS capabilities and configuration settings. Organizations using applications with the ability to make encrypted DNS queries (such as Firefox) should investigate how the application can log its DNS queries for investigation purposes. Analysts can also ingest these logs into a Security Information and Event Monitoring (SIEM) solution. During testing, organizations should ensure they are profiling applications not only for external connections but also for characteristics of the connections, such as JA3 and JA3S signatures. Keeping track of these configuration settings and application fingerprints will aid organizations in finding unusual or unwanted connections in their environment.

For applications that supply the capability to enable encrypted DNS, organizations should update configuration management standards to prevent unauthorized usage of encrypted DNS technologies. For example, creating following two files on a Windows endpoint running Firefox will block DoH (Mozilla [4], n.d.).

File: “C:\Program Files\Mozilla Firefox\defaults\pref\autoconfig.js”

```
pref("general.config.filename", "firefox.cfg");
pref("general.config.obscure_value", 0);
```

File: “C:\Program Files\Mozilla Firefox\firefox.cfg”

```
// Disable Firefox Trusted Recursive Resolver (DoH)
lockPref("network.trr.mode", 5);
```

Drew Hjelm, drew@vets.io

Organizations should use application whitelists to prevent unauthorized applications from running, which may also communicate using encrypted DNS. Various security frameworks, such as the CIS Critical Security Controls (Center for Internet Security, 2019), push organizations to perform application whitelisting and stop unauthorized applications from running on endpoints. One whitelisting methodology published by Microsoft is its AppLocker utility included in Windows and a set of tools called AaronLocker (Margosis, 2019).

5.3. SIEM and IDS Alerting for encrypted DNS Indicators

Organizations can mostly automate the analysis of detecting encrypted DNS discussed in this research in some fashion with a Security Incident and Event Monitoring (SIEM) platform and Intrusion Detection System (IDS). Many organizations already have some form of SIEM platform they use to aggregate logs from various network devices, servers, and endpoints. Analysts can then correlate logs against malicious activity from threat intelligence or other signatures. With the addition of Zeek logs, an organization can identify if there are malicious flows to investigate that have unauthorized DoH or DoT activity.

6. Conclusions

This research is meant to show how attackers and other actors can use the increased privacy and security features included in encrypted DNS to bypass traditional controls organizations may have adopted to secure their environments. The research is not meant to cast aspersions on new protocols, but to ensure organizations are aware of new threats to look for in their environments. With proper preparation of an organization's detection infrastructure organizations do not need to fear encrypted DNS. The unmitigated usage of encrypted DNS, particularly DNS over HTTPS, could allow attackers and insiders to bypass organizational controls. Even without intercepting and decrypting encrypted DNS traffic, organizations have many options they can use to view how endpoints use the encrypted DNS traffic. Organizations can perform analysis on encrypted traffic to find malicious actors without decrypting the traffic and breaching the

privacy of those using these services. Organizations contemplating the risks to their data from encrypted DNS should consider implementing controls at the network-level and on their endpoints to ensure that malign actors cannot use the new protocols for unauthorized purposes.

References

360 Netlab. (2019, July 1). An Analysis of Godlua Backdoor. Retrieved July 14, 2019, from https://blog.netlab.360.com/an-analysis-of-godlua-backdoor-en/

Active Countermeasures. (2019, July 15). RITA. Retrieved July 17, 2019, from https://github.com/activecm/rita

Bambenek, J. (2019, July 2). Bambenek/block-doh. Retrieved July 11, 2019, from https://github.com/bambenek/block-doh

Bortzmeyer, S. (2015, August). RFC 7626 - DNS Privacy Considerations. Retrieved from https://tools.ietf.org/html/rfc7626

Chromium. (2019, July 18). Add DNS-over-HTTPS to chrome://flags. Retrieved July 21, 2019, from https://chromium-review.googlesource.com/c/chromium/src/+/1639663

Draeger, A. (2019). Digging for Gold: Examining DNS Logs on Windows Clients. SANS Reading Room. Retrieved from https://www.sans.org/reading-room/whitepapers/dns/digging-gold-examining-dns-logs-windows-clients-38975

Du Toit, R. (2017). Responsibly intercepting TLS and the impact of TLS 1.3. Retrieved August 7, 2019, from https://www.symantec.com/content/dam/symantec/docs/other-resources/responsibly-intercepting-tls-and-the-impact-of-tls-1.3-en.pdf

Farnham, G. (2013). Detecting DNS Tunneling. SANS Reading Room. Retrieved from https://www.sans.org/reading-room/whitepapers/dns/detecting-dns-tunneling-34152

Drew Hjelm, drew@vets.io

Fortinet. (2018, November 14). As the Holiday Season Draws Near, Mobile Malware Attacks Are Prevalent. Retrieved from https://www.fortinet.com/blog/industry-trends/as-the-holiday-season-draws-near--mobile-malware-attacks-are-pre.html

Ghedini, M. (2018, September 24). Encrypt it or lose it: how encrypted SNI works. Retrieved from https://blog.cloudflare.com/encrypted-sni/

Google. (2019, June 26). JSON API for DNS over HTTPS (DoH) | Public DNS | Google Developers. Retrieved July 21, 2019, from https://developers.google.com/speed/public-dns/docs/doh/json

Hoffman, P., & McManus, P. (2018, October). RFC 8484 - DNS Queries over HTTPS (DoH). Retrieved from https://tools.ietf.org/html/rfc8484

Hu, Z., Zhu, L., Heidemann, J., Mankin, A., Wessels, D., & Hoffman, P. (2016, May). RFC 7858 - Specification for DNS over Transport Layer Security (TLS). Retrieved from https://tools.ietf.org/html/rfc7858

Kennedy, D. (2019, July 1). Using Sysmon and ETW For So Much More. Retrieved July 21, 2019, from https://www.binarydefense.com/using-sysmon-and-etw-for-so-much-more/

Lee, M., Mercer, W., Rascagneres, P., & Williams, C. (2017, May 12). Player 3 Has Entered the Game: Say Hello to 'WannaCry'. Retrieved from https://blog.talosintelligence.com/2017/05/wannacry.html

Let's Encrypt. (2019). Let's Encrypt Stats. Retrieved July 21, 2019, from https://letsencrypt.org/stats/

Drew Hjelm, drew@vets.io

Levy, I. (2018, March 9). TLS 1.3: better for individuals - harder for enterprises. Retrieved from https://www.ncsc.gov.uk/blog-post/tls-13-better-individuals-harder-enterprises

Magisterquis. (2019, July 1). dnsbotnet. Retrieved July 14, 2019, from https://github.com/magisterquis/dnsbotnet

Margosis, A. (2019, July). AaronLocker. Retrieved July 21, 2019, from https://github.com/microsoft/AaronLocker/blob/master/Documentation/AaronLocker.docx

MITRE. (2018). Technique: Domain Fronting - Enterprise. Retrieved August 6, 2019, from https://attack.mitre.org/techniques/T1172/

Mockapetris, P. (1987, November). RFC 1035 - Domain names - implementation and specification. Retrieved from https://tools.ietf.org/html/rfc1035

Mozilla. (2019, March 23). Gecko Logging. Retrieved July 11, 2019, from https://developer.mozilla.org/en-US/docs/Mozilla/Developer_guide/Gecko_Logging

Mozilla. (2019, March 18). HTTP logging. Retrieved from https://developer.mozilla.org/en-US/docs/Mozilla/Debugging/HTTP_logging

Mozilla. (2019, February 7). Trusted Recursive Resolver. Retrieved July 21, 2019, from https://wiki.mozilla.org/Trusted_Recursive_Resolver

Mozilla. (n.d.). Customizing Firefox Using AutoConfig | Firefox for Enterprise Help. Retrieved July 11, 2019, from https://support.mozilla.org/en-US/kb/customizing-firefox-using-autoconfig

Drew Hjelm, drew@vets.io

MyOnlineSecurity. (2019, June 10). It looks like another DNS compromise hack happening. Retrieved July 14, 2019, from https://myonlinesecurity.co.uk/it-looks-like-another-dns-compromise-hack-happening/

Russinovich, M. (2019, June 28). Sysmon - Windows Sysinternals. Retrieved July 21, 2019, from https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon

Salesforce. (2019, April 8). ja3. Retrieved July 20, 2019, from https://github.com/salesforce/ja3

Sandvine. (2018, October). The Global Internet Phenomena Report. Retrieved from https://www.sandvine.com/hubfs/downloads/phenomena/2018-phenomena-report.pdf

Sensepost. (2019, February 6). goDoH. Retrieved July 14, 2019, from https://github.com/sensepost/godoh

SpiderLabs. (2018, November 13). DoHC2. Retrieved July 14, 2019, from https://github.com/SpiderLabs/DoHC2

SwiftOnSecurity. (2019, June 13). SwiftOnSecurity/sysmon-config. Retrieved July 21, 2019, from https://github.com/SwiftOnSecurity/sysmon-config

Vixie [PaulVixie], P. (2018, October 21). DoH is an over the top bypass of enterprise and other private networks. But DNS is part of the control plane, and network operators must be able to monitor and filter it. Use DoT, never DoH. [Tweet]. Retrieved July 14, 2019, from https://twitter.com/paulvixie/status/1053886628832382977

Drew Hjelm, drew@vets.io