Global Information Assurance Certification Paper

Challenges in Effective DNS Query Monitoring

GIAC (GCIA) Gold Certification

Author: Caleb Baker, caleb@calebbaker.net
Advisor: David Hoelzer

Accepted: September 27, 2019

Abstract

Domain Name System (DNS) queries are fundamental functions of modern computer networks. Capturing the contents of DNS queries and analyzing the logged data is a recommended practice for gaining insight into activity on a network and monitoring for unusual behavior. Multiple solutions and approaches are available for monitoring DNS queries. Some methods add the capability to redirect queries identified as malicious, stopping an attack. This paper investigates the effectiveness of solutions that utilize the monitoring of DNS queries to detect and block behavior DNS queries identified as potential indicators of compromise. The performance of each tool will be evaluated against a sample of real-world threats that utilize DNS queries. As the prevalence of DNS query monitoring increases, attackers will need to take steps to bypass monitoring by obfuscating DNS queries. Accordingly, this paper will also assess the capabilities of each tool to detect techniques for DNS query obfuscation.

1. Introduction

Capturing the contents of DNS (Domain Name System) queries and analyzing the captured data provides valuable insights into activity on a network. DNS query monitoring has several unique advantages compared to other methods for monitoring network activity. It is relatively simple to put in place, requiring minimal configuration changes to setup in most cases. DNS queries are mostly unencrypted, making the analysis process less complex when compared to encrypted application protocols.

Due to the usefulness of this data, DNS query monitoring solutions have developed which include the feature of monitoring requested domains against a blacklist of domains the solution has categorized as indicators of compromise. If a DNS query is identified that matches a blacklist entry, the solution will then provide an alert or notification.

A fundamental concern when considering DNS query blacklists is in evaluating the effectiveness of the threat intelligence behind each solution. Can DNS query blacklists be trusted to block most real-world threats from commodity malware, or are detection rates too low to be reliable?

Public collections of malicious domains are available online that can be integrated with open-source software such as Security Onion at no cost. This research will investigate whether these open-source blacklists provide a similar level of effectiveness to the commercial blacklists without the cost of a commercial solution.

This paper will explore the challenges to DNS query monitoring solutions, examine the specific strengths and weaknesses of existing solutions, and present what must be done to maximize the effectiveness of a DNS query monitoring solution.

1.1. Value of DNS Query Monitoring for Intrusion Detection

DNS queries are fundamental to the operation of the internet. Client computers use DNS to translate a domain name into an IP address that can be used to establish a connection to a server elsewhere on the internet. DNS domain queries are used almost universally on the internet for client connections to a server. For readers unfamiliar with DNS, a recommended resource is “Is Anyone Out There? Monitoring DNS for Misuse”

(Fornero, 2016), which provides an excellent overview of the DNS query process and attacker methods that use DNS records.

Attackers are incentivized to use DNS due to the flexibility it provides. If an attacker manually specifies an IP address in an attack, the attack will only be effective so long as the specific IP address is reachable by the target. Using DNS-based attacks allows an attacker to quickly move servers between different IP addresses, forcing a "whack-a-mole" approach to block IP addresses.

Because of the value of DNS query data, logging DNS queries in a network is considered a best practice (Todd, 2017). When properly implemented, these logs allow a forensic investigator to review what network devices queried for a specific domain name at what specific times. These logs can be used to trace the entry point and spread of an attack on a network.

Due to the usefulness of DNS query data, solutions have been developed that take this concept one step further by actively monitoring for DNS queries requesting domains on a blacklist. Detection can take the form of generating an alert or notification, subjecting the query to a DNS sinkhole, or both.

1.2. Examples of DNS Query Monitoring for Intrusion Detection

An example of a threat that can be countered using DNS query monitoring is botnet command and control traffic. The Mirai botnet utilized 67 domains for communication with at least 484 separate IP addresses. The command and control servers hosted at these addresses would send commands to a variety of internet-connected devices (Antonakakis et al., 2017). Monitoring for devices querying for these domains allows for passive identification of compromised devices. Once identified, the blacklisted DNS queries can be disrupted by subjecting them to a DNS sinkhole. The DNS sinkhole prevents successful communication from the compromised device back to the botnet command and control server.

The CryptoWall ransomware attempts to use DNS to resolve domain names pointing to servers that generate and store the private encryption keys used by CryptoWall to encrypt files on the target system (Cabaj, Gawkowski, Grochowski, &

Osojc, 2015). Detecting and alerting on queries for these domains would enable faster isolation of a compromised system. In the implementation used for CryptoWall, successfully subjecting all queried domains to a DNS sinkhole results in the payload not reaching the attacker's encryption key management servers, halting the ransomware execution.

Another example of a real-world threat that DNS query monitoring can address is lookalike domain phishing. In this type of attack, an attacker registers a domain very similar to the target domain, such as examp1e.com rather than example.com (1 replaces lowercase L). Using this technique, an attacker sends what appears to be a link to a partner or internal site, but the link instead directs to infrastructure controlled by the attacker (Krebs, 2015). DNS query monitoring can be configured to address this by blocking newly seen domains when using the DNS sinkhole approach.

1.3. Potential Weaknesses in DNS Monitoring

There are two fundamental keys to effective DNS query monitoring. The blacklist must have reliable and up-to-date threat intelligence information, and the solution must accurately capture all DNS queries.

Complete knowledge of all malicious domains is a practical impossibility; therefore, the accuracy rate for blacklisted domains is always less than 100% given a sufficiently large sample. Accuracy rates for blacklists vary considerably, with the most accurate blacklists having around 75% accuracy (Kührer, Rossow, & Holz, 2014).

Successful obfuscation of DNS queries allows for bypassing DNS query monitoring. If the same method is in use for both DNS query logging and monitoring, this means that logs of the obfuscated queries are also unavailable.

1.3.1. Methods for bypassing DNS monitoring

One method of DNS query obfuscation is using DNS over HTTPS or DNS over TLS. Both of these methods use Transport Layer Security (TLS) to encrypt DNS queries; DNS over TLS uses port TCP 853, while DNS over HTTPS encapsulates DNS queries in HTTPS using port TCP 443.

Malware abusing DNS over HTTPS for command and control is not just a theoretical threat – a variant of the Godlua trojan discovered in early 2019 uses a DNS over HTTPS mechanism as a method of obfuscating command and control traffic (Turing, 2019).

An additional method of obfuscating DNS queries is bypassing the DNS resolver specified for the system and sending DNS requests directly to a different DNS server. Blocking this method at the network level is easily accomplished by blocking outbound connections with a destination port of UDP 53. However, some public DNS resolvers listen on non-standard ports, allowing for queries to complete even if outbound port UDP 53 is blocked. For example, the Quad9 DNS resolver accepts DNS queries on the standard port of UDP 53, and also accepts queries on an alternate port of UDP 9953. Figure 1 shows an example of this query.

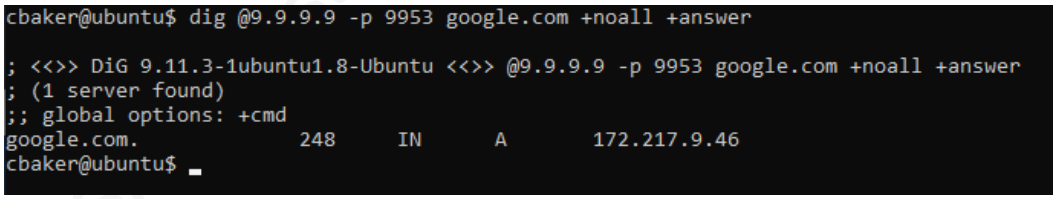


Figure 1. Querying the Quad9 DNS resolver using non-standard port UDP 9953

Another method to bypass DNS query monitoring is to host payloads on services that host content from user submissions on a single public domain. Some real-world botnet operators have used Twitter to command and control compromised hosts (Dwyer, 2009). It is not possible to block this vector using DNS query sinkhole without completely cutting off access to Twitter.

2. Overview of DNS Query Monitoring Solutions and Capabilities

Current commercial DNS sinkhole solutions include Webroot DNS Security, Infoblox DNS Firewall, Mimecast Web Security, Verizon DNS Safeguard, and SafeDNS. Each of these services has similar basic DNS Sinkhole functionality. Technical differences between the services come down to additional functionality offered (such as

roaming client software or web proxy functionality), the threat intelligence information powering them, and options for reporting and alerting.

Some consumer-focused DNS Resolvers such as OpenDNS and Quad9 provide sink holing of DNS queries. These consumer-focused options do not include any integrated reporting or alerting capability. A possible point of confusion is the relationship between OpenDNS and Cisco Umbrella; Cisco acquired OpenDNS in 2015. After the acquisition, the enterprise-focused version was re-branded as Cisco Umbrella, while the consumer-focused version remained branded as OpenDNS (Cisco Systems Inc, 2016).

Many software packages for packet inspection and analysis include a feature to detect DNS queries identified within a packet capture. The capabilities for DNS query detection vary from basic blacklisting of DNS queries in Cisco Firepower to full DNS application layer inspection in Zeek (which was formerly named Bro). Various enterprise firewalls such as Sonicwall, pfsense, and Palo Alto also include this DNS query detection feature.

2.1. DNS Sinkhole approach

The DNS Sinkhole approach consists of configuring DNS queries to be pointed to a resolver provided by the solution. The resolver includes a blacklist of domains configured to not resolve to the authoritative information, but instead to resolve to a sinkhole IP containing a message the domain is blocked. The domain information contained in the blacklist is categorized based on a proprietary set of intelligence sources. These could include public DNS blacklist services, an internal Threat Intelligence team, and user reports. The DNS Sinkhole approach also may be referred to as the "DNS Firewall" or "DNS Blackhole" approach.

These approaches may include optional client software which can be loaded on monitored endpoints to allow for more effective capture of DNS queries if the client computer is a laptop that connects to multiple networks while traveling. The client software also assists in tracking a query to a specific endpoint. The process of identifying the individual device initiating a DNS query can be a challenge if an internal DNS resolver is utilized, which is typical in an enterprise network.

One advantage of this model is that it is a simple way to roll out coverage quickly across a network without requiring altering network traffic flow or hardware. Since the minimum requirement for this method to function is a DNS server entry to be set in the network DHCP settings, it allows for monitoring at branch offices that may not have packet capture equipment. Installing the client software allows DNS queries to be logged and monitored on devices that travel outside the enterprise network.

The disadvantage of this model is that without the client software installed, bypassing monitoring can be accomplished with ease, as all that is required is changing the DNS resolver on a system.

2.2. IDS/IPS DNS Packet Inspection approach

The DNS Packet inspection approach to DNS query monitoring requires capturing network traffic containing DNS queries using an ethernet tap or a span port on a network switch. The captured traffic is then analyzed by an intrusion detection system (IDS) or intrusion prevention system (IPS). Under a typical configuration, an IDS would alert on a detected blacklisted DNS query or obfuscation attempt, while an IPS would block a detected blacklisted DNS query or obfuscation attempt.

An advantage of the DNS packet inspection approach is that, because of the integration with a network monitoring tool, all network traffic is visible and available for inspection. Inspecting all network traffic increases the ability of DNS packet inspection to detect DNS query obfuscation attempts since the inspection process will have more network traffic available than just the DNS query portion. This additional information is not available to a DNS Sinkhole, which can only inspect the DNS query portion without additional client software installed on all endpoints.

A key disadvantage is that an IDS is not able to prevent the resolution of a blacklisted domain name since preventing the resolution of a blacklisted domain requires an IPS to modify the query. When a network firewall includes a DNS packet inspection feature, the firewall typically operates in IPS mode.

3. Research Method - Testing Methodology

3.1. Discussion of Testing Selections

This paper will explore five solutions; two solutions using a DNS sinkhole approach, two solutions using DNS packet inspection, and an additional solution using the packet inspection approach with publicly available blacklist information.

Cisco Umbrella and DNSFilter were selected for examination as examples of DNS sinkhole solutions. Cisco Umbrella is the enterprise-focused version of OpenDNS, which was acquired by Cisco in 2015. OpenDNS was one of the first public DNS sinkhole providers (Bruneau, 2010) and Cisco Umbrella remains one of the most popular commercial DNS sinkholes. DNSFilter is a Cisco Umbrella competitor with a very similar set of features. DNSFilter was chosen for its competitive feature set as well as the ease of signing up for an account for testing purposes. Both Cisco Umbrella and DNSFilter feature client software for Windows and Mac that can be installed to provide DNS blacklist protection and DNS query logging for computers connected to any network.

LogRhythm NetMon Freemium and a Fortinet FortiGate firewall are examples of the DNS packet inspection approach that will be utilized in this testing. The blacklisting feature is not available in LogRhythm NetMon Freemium, as it functions as an IDS and does not include a blacklisting feature. The Fortinet FortiGate line of network firewalls includes a DNS packet inspection option, and the Fortinet FortiGuard add-on service includes a blacklist of domains. The FortiGate Firewall operates in Intrusion Prevention mode, blocking DNS queries on the FortiGuard blacklist.

Security Onion will be used as an open-source option and tested with freely available public domain blacklists. Security Onion uses the Zeek intrusion detection software for packet analysis. The data from Zeek will be analyzed directly as well as with the RITA tool that can be easily added into Security Onion (Haselhorst, 2019). RITA includes blacklisted domains from MalwareDomainList.com and MalwareDomains.com. Additional custom domains were added to the blacklist in RITA using domains from the Unified Hosts project. This project combines multiple public blacklists into a single list (Black, 2019). As of August 10, 2019, this list contains 41,749 domains identified with

adware and malware. The list was refreshed daily during the testing process. Security Onion will be utilized to provide a baseline for the level of insight available using freely available information.

3.2. Test Environment Overview

All testing will be done utilizing virtual machines running in the VMWare ESXi Hypervisor. The client endpoints will be running Windows 10 release version 1809. The client will be the base install from Microsoft, with the only customizations being the installation of VMware tools 10.3.10. Additional software will be installed for various testing scenarios, as specified below.

Each testing environment will be separated within the hypervisor, and network segmentation will be employed to prevent data from each testing scenario from influencing other test results.

For solutions utilizing the DNS sinkhole approach, testing endpoints will be configured to send DNS queries directly to the global resolvers provided by the solution. A local DNS forwarder will not be utilized during testing to prevent interference with the test results, even though such a configuration is typical. Figure 2 below shows a diagram of the network topology for testing clients utilizing the DNS sinkhole approach.

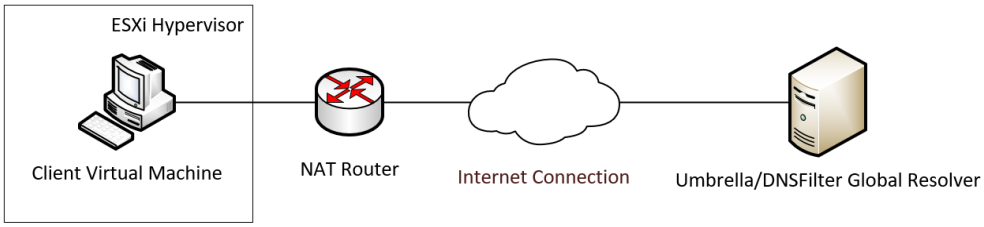


Figure 2. DNS Sinkhole Approach for DNS Query Monitoring

For solutions utilizing the Packet Inspection approach, testing endpoints will direct queries to a NAT router running a DNS resolver. The NAT router will query the DNS root servers to resolve the queries. Packets will be captured between the client and the DNS resolver. Figure 3 shows a network diagram of this testing topology.

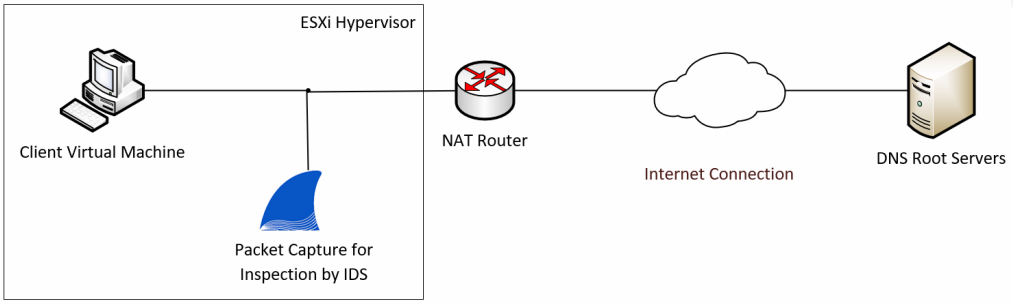


Figure 3. Packet Inspection Approach for DNS Query Monitoring (Intrusion Detection System)

The firewall-integrated packet inspection approach is very similar to the packet capture approach. The critical difference is the packets containing DNS queries are inspected by the firewall device rather than on a separate device. For this configuration, the firewall will be configured as the local DNS resolver. A network diagram for this topology is shown in Figure 4.

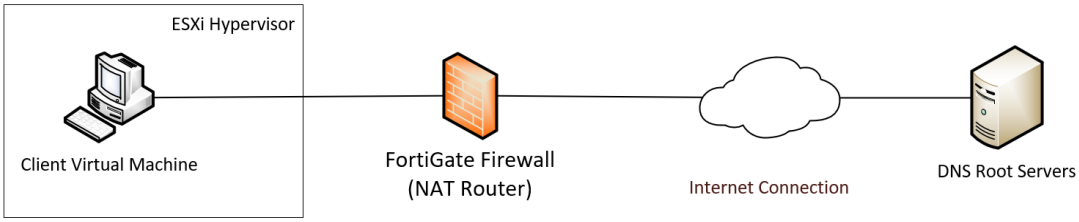


Figure 4. Packet Inspection Approach for DNS Query Monitoring (Integrated with Network Firewall)

3.3. Testing Process - Detection of Malicious Behavior

It is important to note that for the malicious domain detection, the objective is to examine accuracy rates in real-world scenarios, not to perform a statistically significant comparison.

LogRhythm NetMon Freemium does not include a feature to query a domain blacklist, so it will be omitted from this portion of the testing. The Quad9 DNS resolver will be substituted to provide an additional data point.

3.3.1. Testing Process - DNS Tunneling Activity

The dnscat2 tool will be utilized to create a DNS tunnel between the testing client and a dnscat2 control server configured just for use in this test. The client will remain

configured to send DNS queries to the configured DNS resolver – i.e. not directly to the dnscat2 server.

3.3.2. Testing Process - DNS Queries from Malware Samples

DNS queries from real-world malware samples will be tested by obtaining public samples submitted to the Hybrid Analysis sandboxing service. The samples obtained will be categorized for the number of detections against multiple anti-malware tools for that file using the VirusTotal website. Specific interest will be paid to relatively unknown samples with 10 or fewer detections on VirusTotal. Statistics will also be tracked separately for samples with more detections on VirusTotal since the detection of these types of malware is still valuable. Fifty samples will be utilized, with at least 10 being samples with 10 or fewer detections. Figure 5 below shows an example of a suspected malware sample that would be examined. The double arrow symbol in the lower right indicates that the sample has network activity, which may include DNS queries.

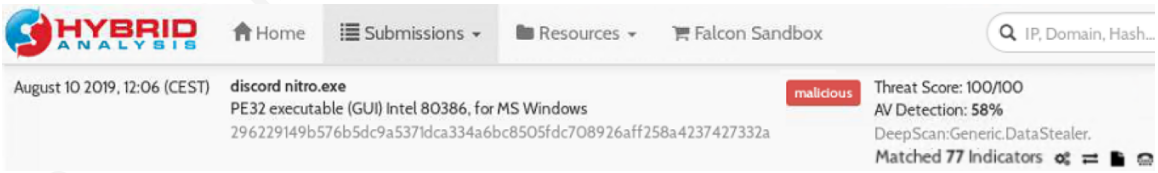


Figure 5. Suspected Malware Sample from the Hybrid Analysis Sandbox *(edited for clarity)*.

A maximum of one query for a domain (including subdomains) will be tested. For example, if a sample queries subdomain.example.com, a second sample queries example.com, and a third sample queries subdomain2.example.com, only subdomain.example.com would be tested as it was the first example seen for the domain example.com.

3.3.3. Testing Process - DNS Queries from Phishing Samples

Fifty domains obtained from samples of identified phishing will be queried on a test client. The domains will be obtained from a combination of submissions investigated by the author and supplemented with additional sites obtained from the Hybrid Analysis sandboxing service. Phishing pages will be examined to verify that the phishing activity is still active at the time of the test.

3.4. Testing Process - Detection of DNS Query Obfuscation

For all solutions that support blocking blacklisted queries, a manual block will be put into place for the domain "dnstest.ugottasecure.com". The domain "dnstest.ugottasecure.com" will then be queried from the test client to determine if the authoritative IP address information is returned, or if the response is a sinkhole IP address. If the response is a sinkhole IP, this indicates that the DNS query has been blocked.

3.4.1. Detection of DNS Queries Manually Pointed to a DNS Server

A query will be manually directed to the Google Public DNS resolver at 8.8.8.8 using the nslookup command. Solutions will be evaluated on based whether the DNS query is detected, blocked, or responded to with the authoritative IP address without alteration.

3.4.2. Detection of DNS Queries on Destination Ports other than UDP 53

A DNS server will be configured using the BIND software to listen on a randomly chosen ephemeral UDP port. The client will then be configured to match the established settings through several tests. The dig package from the Windows version of the BIND project will be installed to allow for testing lookups over non-standard ports, as the DNS Client in Windows 10 does not natively support destination ports other than UDP 53. The testing process will be repeated five times using randomly chosen UDP ports. Solutions will be evaluated based on whether the DNS query is detected, blocked, or responded to with the authoritative IP address information.

3.4.3. Detection of DNS Queries using DNS over HTTPS

The Firefox web browser will be utilized and configured to query the Cloudflare DNS servers using DNS over HTTPS. The DNS over HTTPS feature was first introduced in Firefox version 62 in September 2018. The feature is activated in the connection settings dialog, as shown in Figure 6 below. It is not anticipated that a DNS query monitoring solution would be able to detect the content of the query, as monitoring the content of the query would require SSL inspection. However, it is possible to block the connection to the DNS over HTTPS server. Solutions will be evaluated on based whether

the DNS over HTTPS traffic query is detected, blocked, or responded to with the authoritative IP address information.

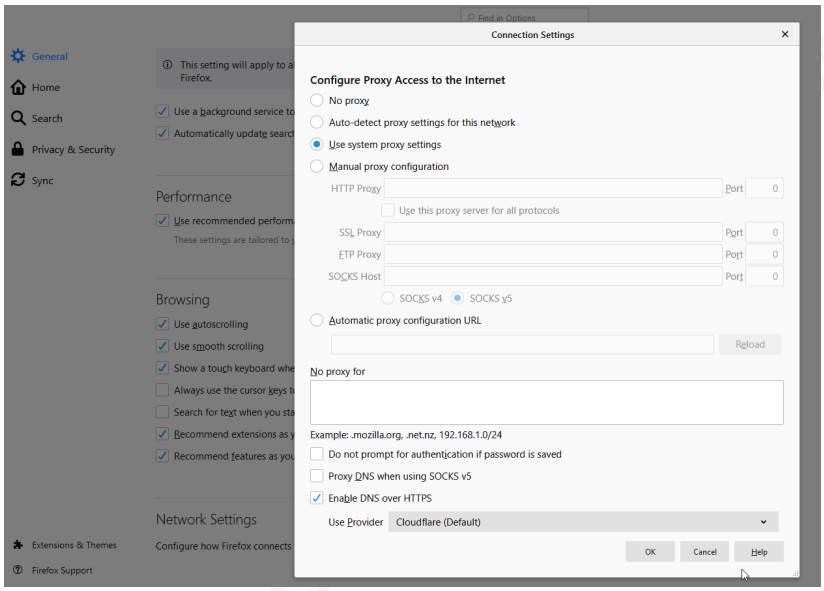


Figure 6. DNS over HTTPS setting in Firefox 68.0

3.4.4. Monitoring Bring Your Own Device (BYOD) Networks

Cisco Umbrella and DNSFilter will be tested both with the roaming client software installed as well as on a device with no custom configuration outside of the DNS servers assigned via DHCP. The roaming client software configures a DNS resolver on the client endpoint to capture DNS queries. The Cisco Umbrella roaming client version used in testing is 2.2.150, and the DNSFilter roaming client version is 1.3.5.0.

This method will be used to determine if some types of detections are possible only with the agent software running on a client and therefore would be missed in a BYOD network if the monitoring agent is not installed. The agent software also installs a Cisco certificate into the trusted root certificate authority store on the client. The root certificate allows for proxying of some HTTPS requests that the Umbrella client determines to be suspicious to a Cloud-based web proxy operated by Cisco.

4. Test Results & Analysis

4.1. Detection of Malicious Domains Testing Results

Testing was carried out for each scenario as was specified in Section 3.3. Table 1 is an overview of the results of the testing for each solution. The same sample domains were used with each solution. Each sample was tested in the five separate solutions at the same time, so as not to give any solution a time advantage in adding the domain in the malicious sample to its blacklist.

	Umbrella	DNSFilter	Quad9	FortiGate	Security Onion
Phishing Domains (out of 50)	14 (28 %)	5 (10%)	3 (6%)	32 (64%)	0 (0%)
Malware Domains (out of 50)	26 (52%)	1 (2%)	7 (14%)	34 (68%)	0 (0%)
Malware Domains, unknown samples (out of 10)	4 (40%)	0 (0%)	0 (0%)	5 (50%)	0 (0%)
Cumulative (out of 100)	40 (40%)	6 (6%)	10 (10%)	66 (66%)	0 (0%)

Table 1. – Results of testing for domains from malicious samples. Number and percent of detections are listed.

Overall, 76 out of the 100 domains tested were categorized as malicious by at least one of the blacklists. This result for the blocking percentage is very similar to the results from "Paint It Black: Evaluating the Effectiveness of Malware Blacklists" (Kührer, Rossow, & Holz, 2014), with the most effective solution in that testing blocking 75% of malware domains.

The data showed substantial differences in blocking percentage between the blacklists tested. The commercial blacklists showed significant differences in blocking performance. The freely available blacklists used for testing also fared significantly worse than the commercial blacklists.

4.1.1. Analysis of Malicious DNS Query Test Results

The real-world testing shows that DNS blacklists are not an adequate measure in preventing access to malicious domains; even the most effective blacklists fail to block too many malicious domains.

DNS query blacklists were generally more effective against malware than against phishing. During the data gathering, it was observed that phishing sites frequently seemed to rely on compromised web hosting accounts for hosting the page. The domains associated with these compromised sites are likely the domains being used for the DNS query. Reusing legitimate domains makes identification of phishing at the domain level more difficult. A phishing page hosting on a domain may only be a small portion of a legitimate site. The domain for the compromised site is likely to have already been categorized by the blacklist based on genuine content. Prior categorization increases the difficulty for the blocklist since the domain must first be identified as potentially compromised, and then have the domain category reevaluated before the blacklisting is effective.

Approximately 50% of the phishing samples examined had already been taken offline by the time they were examined. This result corresponds with research Webroot published in 2016, showing that 84% of phishing pages were online less than 24 hours (Webroot, 2016).

Dynamic DNS services were heavily utilized in samples tested. 14% of the malware samples used dynamic DNS domains with known dynamic DNS providers. The prevalence is understated in the data, as additional subdomains under the same domain are not considered. Both Cisco Umbrella and the Fortinet FortiGate are configured by default to block known common dynamic DNS domains.

Some malware samples were observed attempting to defeat DNS blacklists by having multiple domains resolve to the same IP address. Figure 7 shows an example of the multiple domains and the dynamic DNS tactics being used simultaneously. In the example, two different subdomains are configured on different dynamic DNS domain services (hopto.org and duckdns.org). Both domains queried by this malware sample resolve to the same IP address, which indicates the attacker is attempting to defeat DNS

blacklisting since both domains must be blacklisted to prevent communication from the compromised system back to the attacker.

Domain	Address	Registrar	Country
bylgay.hopto.org OSINT	152.246.120.88 TTL: 59	TLDS L.L.C. d/b/a SRSPlus Organization: No-IP.com Name Server: NF1.NO-IP.COM Creation Date: Thu, 17 Feb 2000 19:56:50 GMT	Brazil
soucdtevoceumcuzao.duckdns.org OSINT	152.246.120.88 TTL: 59	Gandi SAS Name Server: NS1.DUCKDNS.ORG Creation Date: Fri, 12 Apr 2013 19:58:56 GMT	Brazil

Figure 7. Screenshot from a sample on the Hybrid Analysis site *(edited for clarity)*

One potential issue with these results is that just because a malicious executable is issuing the query, it does not guarantee the queried host is directly tied to the malicious activity. For example, several samples queried for "checkip.amazonaws.com". This domain is for Amazon Web Service's Public IP check service and was likely used by malware to determine the public IP of the compromised system. Where it was identified that a domain was probably being queried incidentally, these domains were omitted.

Quad9 performed better than anticipated, having a higher blacklist percentage than one of the commercial solutions. A key downside to Quad9 is that blacklisted domains return an NXDOMAIN response. An NXDOMAIN response is the same response that would be returned if the domain queried did not exist. Because of the NXDOMAIN response, the blacklisted query responses are indistinguishable from actual invalid domain responses; therefore, detection of a blacklisted query response via packet inspection by an IDS is not possible using the Quad9 resolver. Quad9 performed well enough that its use in a home network would be warranted, or in a situation with an extremely constrained budget where a commercial solution is not feasible.

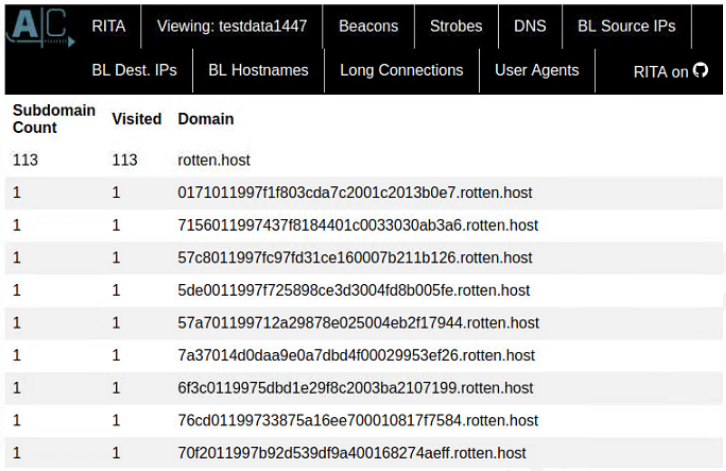
The Cisco Umbrella roaming client blocked four additional phishing domains and one additional "fresh" malware sample via the Web Proxy feature that is part of the roaming client software. These domains would not be blocked in a BYOD environment, as the Web Proxy feature requires installing a Cisco Umbrella root certificate on the client device. DNSFilter had the same results both with and without the roaming client installed; the roaming client had no impact on blocking for the tested domains. Table 2 contains a summary of the results with and without the Roaming Client software installed on the testing endpoint.

	Umbrella – with Roaming Client	Umbrella – without Roaming Client	DNSFilter – with Roaming Client	DNSFilter – without Roaming Client
Phishing Domains (out of 50)	14 (28 %)	10 (20 %)	5 (10%)	5 (10%)
Malware Domains (out of 50)	26 (52%)	25 (50%)	1 (2%)	1 (2%)
Malware Domains, unknown samples (out of 10)	4 (40%)	3 (30%)	0 (0%)	0 (0%)
Cumulative (out of 100)	40 (40%)	35 (35%)	6 (6%)	6 (6%)

Table 2. – Results of testing for domains from malicious samples. Number and percent of detections are listed.

Neither DNS sinkhole solution blocked the DNS tunnel. This result was unexpected, as both Cisco Umbrella and DNSFilter had been configured using the option to block newly seen domains, and the domain used for testing had been registered less than one week before the test.

Both Security Onion and LogRhythm NetMon detected the DNS queries associated with the DNS tunnel. Since these are both typically deployed as IDS solutions, it is not possible to block the DNS tunnel since that would require the query response to be modified. Figure 8 shows a screenshot of the DNS tunneling test being detected by the RITA tool used in conjunction with Security Onion.



Subdomain Count	Visited	Domain
113	113	rotten.host
1	1	01710119971f1f803cda7c2001c2013b0e7.rotten.host
1	1	71560119974371f8184401c0033030ab3a6.rotten.host
1	1	57c8011997fc97fd31ce160007b211b126.rotten.host
1	1	5de0011997f725898ce3d3004fd8b005fe.rotten.host
1	1	57a701199712a29878e025004eb2f17944.rotten.host
1	1	7a37014d0daa9e0a7dbd4f00029953ef26.rotten.host
1	1	6f3c0119975dbd1e29f8c2003ba2107199.rotten.host
1	1	76cd01199733875a16ee700010817f7584.rotten.host
1	1	70f2011997b92d539df9a400168274aeff.rotten.host

Figure 8. DNS tunneling activity in RITA

4.2. DNS Query Obfuscation Test Results

Testing was carried out for each scenario as was discussed in Section 3.4. Table 3 is an overview of the results of the testing for each solution. The testing of Cisco Umbrella and DNSFilter was performed with the roaming client software installed on the testing devices.

	Umbrella	DNSFilter	NetMon	FortiGate	Security Onion
DNS query to 167.99.202.100 UDP Port 53	Query not detected, returned successfully	Query not detected, returned successfully	Query Detected	Query Detected and blocked	Query Detected
DNS query to 167.99.202.100 on random ephemeral UDP Port	Query not detected, returned successfully	Query not detected, returned successfully	Query Detected	Query not detected, returned successfully	Query not detected, returned successfully
DNS query using DNS over HTTPS	Query not detected, returned successfully	Query not detected, returned successfully	Query not detected, returned successfully	Query not detected, returned successfully	Query not detected, returned successfully

Table 3. – Results of testing for DNS query obfuscation techniques

4.2.1. Analysis of DNS Query Obfuscation Test Results

A significant finding from testing DNS obfuscation is that the agent-based solutions failed to block or capture even the simplest DNS obfuscation attempts. Merely pointing a DNS query to a resolver outside the one built into the agent software was adequate to bypass all the safeguards the agent software provides. This ease of bypassing is disappointing as some solutions in this space (such as Mimecast Web Security) do include this feature. The agent still has some usefulness, in that it allows for DNS queries that would typically not be captured in logs while the client computer roams away from the corporate network to be captured and subjected to blacklisting.

DNS over HTTPS queries are a significant concern, as these were not detected by any of the solutions tested and are challenging to detect as they appear to be HTTPS traffic. Mozilla Firefox has added DNS over HTTPS as a built-in feature and is considering enabling DNS over HTTPS by default in the future (McManus, 2018). If other major browsers follow this path, DNS query monitoring may become much more challenging to implement in the future.

5. Re-test Process

Based on the observed weaknesses to DNS obfuscation in the DNS sinkhole solutions, adjustments to the network and client configuration will be tested to see if they can improve on this identified weakness.

5.1. Network Modifications

Cisco Umbrella has created a guide with recommended adjustments that should be made at the network firewall level to reduce vulnerability to obfuscation (Prytuluk, 2019). The steps recommended in this guide are relevant to any DNS sinkhole solution, not just Cisco Umbrella.

To prevent direct queries to DNS servers, block outbound traffic with destination port UDP 53, except to the DNS resolvers provided by the solution (Prytuluk, 2019). If employing a local DNS resolver, the allowed source for DNS queries can be limited to systems functioning as local DNS resolvers. To prevent DNS over TLS queries, block outbound traffic with destination port TCP 853.

Blocking DNS over HTTPS queries is more challenging, as the destination port will be TCP 443, which is used for all HTTPS traffic. The block-doh project curates a list of known public DNS over HTTPS resolvers (Bambenek, 2019). Blocking outbound traffic to these servers with a destination port of TCP 443 will be adequate to block DNS over HTTPS in most cases (including the Godlua trojan mentioned previously). If the malware creator were to operate a dedicated DNS over HTTPS server, this traffic would not be blocked.

5.2. Client Configuration Modifications

For solutions utilizing DNS roaming client software, a local client firewall rule will be configured to block access to outbound traffic with a destination UDP port of 53. It is expected that this will function to allow for blocking of direct DNS queries to other servers.

5.3. Re-test Results Summary and Analysis

The adjustments were very effective at blocking DNS obfuscation attempts. After the adjustments, all solutions successfully blocked DNS queries to public DNS resolvers other than those associated with the tested solution. DNS over HTTPS queries to the Cloudflare DNS resolver were also blocked. Table 4 shows detailed results for the re-test process. Results in italics indicate a change from the initial test.

	Umbrella	DNSFilter	NetMon	FortiGate	Security Onion
DNS query to 167.99.202.100 UDP Port 53	*Query blocked*	*Query blocked*	*Query Detected and blocked*	Query Detected and blocked	*Query Detected and blocked*
DNS query to 167.99.202.100 on random ephemeral UDP Port	Query not detected, returned successfully	Query not detected, returned successfully	Query Detected	Query not detected, returned successfully	*Query Detected*
DNS query using DNS over HTTPS	*Query blocked*	*Query blocked*	*Query blocked*	*Query blocked*	*Query blocked*

Table 4. - Results of testing for DNS query obfuscation with network adjustments

The network adjustments made a significant difference in blocking obfuscated traffic. However, all solutions other than NetMon Freemium and Security Onion were still unable to detect outbound DNS queries when an alternate destination UDP port is selected. For Security Onion, it was necessary to adjust the Zeek intrusion detection system to monitor for DNS traffic on all UDP ports. This configuration change would be resource-intensive on a production installation of Security Onion; however, it could be utilized with a separate system used only for packet capture analysis.

The client firewall rule was effective in blocking DNS requests to different DNS servers, however for both Cisco Umbrella and DNSFilter it was necessary to exempt the IP Addresses for the public DNS resolvers provided by the solutions. The requirement to exempt the servers for the solutions indicates that in both cases the roaming client software is communicating with the solution DNS resolvers using port UDP 53.

6. Conclusion

Even under ideal conditions, a DNS query blacklist was not shown to be a reliable method of stopping threats. There is some value as a defense in depth measure, but it should be anticipated that at least 25% of malicious DNS queries will get through.

DNS sinkhole blacklist accuracy varies significantly. There are a considerable number of DNS sinkhole solutions beyond those covered in this research. When considering purchasing such a solution, performing a comparison with real-world malicious queries similar to the research in this paper is recommended. It is critical to verify that features work as expected and that the solution chosen is competitive in blocking percentage.

Without network measures to prevent query obfuscation in place, DNS sinkholes can be trivially bypassed. Malware creators show an awareness of DNS query monitoring and blacklisting and are taking steps to evade DNS query monitoring solutions. If network traffic monitoring and control is minimal, improving those areas will likely be more useful for improving security posture than implementing DNS query monitoring.

There is some value in the roaming client software, as it makes gathering DNS query logs from systems not connected to the enterprise network possible. However,

other solutions such as Microsoft's Sysmon are also adding this capability (Stevens, 2019), so this feature alone does not justify the expense.

DNS over HTTPS is already an effortless and effective way to bypass DNS query monitoring solutions. It is likely to make effective deployment of DNS query monitoring solutions significantly more complicated moving forward, and Godlua is unlikely to be the last malware that utilizes it as a method to obfuscate communication. Currently, blocking known public DNS over HTTPS resolvers on enterprise networks seems to be the only effective counter.

Open-source solutions like Security Onion have significant value for intrusion detection, but the freely available domain blacklists utilized in this research proved to be extremely lacking.

6.1. Research Caveats

The malware portion of this research used the Hybrid Analysis service. It is possible that the intelligence information from this service is shared with one or more of the DNS blacklist solutions in this paper. If this intelligence sharing has occurred, the performance for solutions with access is likely to be artificially inflated in this testing.

The open-source intelligence feed selected here performed particularly poorly – better feeds may exist. The public blacklist that was chosen was selected because it is a combined feed of several other sources that are highly regarded (including those already part of RITA). For further investigation of free public domains blacklists, see "Build Your Private Threat-blocking DNS Server" (Szathmari, 2017).

This research did not attempt to address some issues that could significantly hamper a real-world deployment. No attempt was made in this research to address false positives. Excessive false positives will add a significant amount of administrative overhead. Choosing a solution with a lower false-positive rate at the cost of a slightly lower detection rate may be desirable, depending on an organization's resources and threat appetite. No attempt was made in this research to examine DNS resolver response times, which can have a significant negative performance impact.

References

Antonakakis et al. (2017, August). *Understanding the Mirai Botnet*. Retrieved from https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-antonakakis.pdf

Bambenek, J. (2019, July 2). *Block-doh*. Retrieved August 12, 2019, from https://github.com/bambenek/block-doh

Black, S. (2019, August 10). *Unified Hosts Project*. Retrieved August 10, 2019, from https://github.com/StevenBlack/hosts

Bruneau, G. (2010, August 7). *DNS Sinkhole*. Retrieved from https://www.sans.org/reading-room/whitepapers/dns/dns-sinkhole-33523

Cabaj, K., Gawkowski, P., Grochowski, K., & Osojc, D. (2015, November). *Network activity analysis of CryptoWall ransomware*. Retrieved from http://red.pe.org.pl/articles/2015/11/48.pdf

Cisco Systems, Inc. (2016, November). *OpenDNS Cisco Umbrella*. Retrieved August 15, 2019, from https://umbrella.cisco.com/products/features/opendns-cisco-umbrella

Dwyer, D. (2009, September 3). *Twitter-Based Botnet Command and Control*. Retrieved August 10, 2019, from https://www.secureworks.com/blog/twitter-based-botnet-command-and-control

Fornero, K. (2016, December 29). *Is Anyone Out There? Monitoring DNS for Misuse.* Retrieved from https://www.sans.org/reading-room/whitepapers/dns/paper/37497

Haselhorst, D. (2019, January 2). *Onion-Zeek-RITA: Improving Network Visibility and Detecting C2 Activity*. Retrieved from https://www.sans.org/reading-room/whitepapers/detection/paper/38755

Krebs, B. (2015, March 10). *Spoofing the Boss Turns Thieves a Tidy Profit.* Retrieved August 1, 2019, from https://krebsonsecurity.com/2015/03/spoofing-the-boss-turns-thieves-a-tidy-profit/

Kührer, M., Rossow, C., & Holz, T. (2014, September). *Paint It Black: Evaluating the Effectiveness of Malware Blacklists*. Retrieved from https://christian-rossow.de/publications/blacklists-raid2014.pdf

McManus, P. (2018, June 1). *Improving DNS Privacy in Firefox*. Retrieved August 8, 2019, from https://blog.nightly.mozilla.org/2018/06/01/improving-dns-privacy-in-firefox/

Prytuluk, M. (2019, July 22). *Preventing Circumvention of Cisco Umbrella with Firewall Rules*. Retrieved July 28, 2019, from https://support.umbrella.com/hc/en-us/articles/230904088-Preventing-Circumvention-of-Cisco-Umbrella-with-Firewall-Rules

Rudis, B. (2019, February 26). *Researching the Landscape of DNS over Transport Layer Security (TLS).* Retrieved May 25, 2019, from https://blog.rapid7.com/2019/02/06/secure-that-query-researching-the-landscape-of-dns-over-transport-layer-security-tls/

Saxena, S. (2016, August 28). *Demystifying Malware Traffic.* Retrieved May 25, 2019, from https://www.sans.org/reading-room/whitepapers/malicious/paper/37222

Stevens, D. (2019, June 9). *Tip: Sysmon Will Log DNS Queries*. Retrieved June 25, 2019, from https://isc.sans.edu/diary/Tip: Sysmon Will Log DNS Queries/25016

Szathmari, G. (2017, December 06). *Build Your Private Threat-blocking DNS Server*. Retrieved August 12, 2019, from https://blog.cryptoaustralia.org.au/build-your-private-dns-server/

Todd, B. (2017, March 31). *Creating a Logging Infrastructure*. Retrieved from https://www.sans.org/reading-room/whitepapers/logging/creating-logging-infrastructure-38130

Turing, A. (2019, July 1). *An Analysis of Godlua Backdoor*. Retrieved July 29, 2019, from https://blog.netlab.360.com/an-analysis-of-godlua-backdoor-en/

Webroot Inc. (2016, December). *Webroot Phishing Threat Trends*. Retrieved from https://webroot-cms-cdn.s3.amazonaws.com/7314/8070/2914/Webroot_Threat_Trends_December_2016.pdf

Zeltser, L. (2019, February 27). *Tunneling Data and Commands Over DNS to Bypass Firewalls*. Retrieved July 25, 2019, from https://zeltser.com/c2-dns-tunneling/