An Analysis of the Use of DNS for Malicious Payload Distribution

Submitted in partial fulfilment
of the requirements of the degree of

Master of Science

of Rhodes University

Ishmael Dube

Grahamstown, South Africa
December 2018

Abstract

Abstract: The Domain Name System (DNS) protocol is a fundamental part of Internet activities that can be abused by cybercriminals to conduct malicious activities. Previous research has shown that cybercriminals use different methods, including the DNS protocol, to distribute malicious content, remain hidden and avoid detection from various technologies that are put in place to detect anomalies. This allows botnets and certain malware families to establish covert communication channels that can be used to send or receive data and also distribute malicious payloads using the DNS queries and responses.

Cybercriminals use the DNS to breach highly protected networks, distribute malicious content, and exfiltrate sensitive information without being detected by security controls put in place by embedding certain strings in DNS packets. This research undertaking broadens this research field and fills in the existing research gap by extending the analysis of DNS being used as a payload distribution channel to detection of domains that are used to distribute different malicious payloads.

This research undertaking analysed the use of the DNS in detecting domains and channels that are used for distributing malicious payloads. Passive DNS data which replicate DNS queries on name servers to detect anomalies in DNS queries was evaluated and analysed in order to detect malicious payloads. The research characterises the malicious payload distribution channels by analysing passive DNS traffic and modelling the DNS query and response patterns. The research found that it is possible to detect malicious payload distribution channels through the analysis of DNS TXT resource records.

Keywords: DNS, Malicious Payload Distribution, Passive DNS, Covert DNS Tunneling.

Supervisor : Prof. George. C. Wells

Department : Department of Computer Science
Degree : Master of Science

Acknowledgements

I would like to thank :

- my supervisor for his patience and guidance,
- all those who believed and supported me during my studies (family, friends and colleagues).

Contents

List of Figures **ix**

List of Tables **1**

1 Introduction **2**

1.1 Background 2

1.2 Context of Research 2

1.3 Research Question 4

1.4 Objectives 4

1.5 Approach 5

1.5.1 Dataset 5

1.6 Related Work 6

1.6.1 Attacks on the DNS 6

1.6.2 DNS Tunnelling 7

1.6.3 Fast-Flux Networks 7

1.6.4 Domain Generation Algorithm (DGA) 8

1.6.5 Payload Distribution Channels 8

1.7 Conclusion 9

2 Literature Review **10**

2.1 Introduction 10

2.1.1 Domain Name System 10

2.2 The Abuse of the DNS 14

2.2.1 Protocol-Level Abuse of the DNS 15

2.2.2 Abuse of the DNS at System-Level 19

2.2.3 DNS Amplification Attack 22

2.2.4 Security Measures Put in Place to Protect the DNS from Abuse . . 23

2.3 Passive DNS 24

2.4 Payload Distribution Using the DNS 25

2.4.1 Malicious Payload Distribution Using the DNS 26

2.5 Placement of this Research in the Wider Field of DNS-Based Attacks . . . 26

2.6 Conclusion 28

3 Research Methodology **29**

3.1 Introduction 29

3.2 Background 29

3.3 Previous Studies 30

3.4 Overview of the Approach 32

3.4.1 Datasets 33

3.5 Using Passive DNS to Identify Payload Distribution Channels 34

3.6 General Detection Techniques 35

3.6.1 Payload Analysis 35

3.6.2 Traffic Analysis 36
3.7 Features of EXPOSURE 38
3.7.1 Time-Based Features 38
3.7.2 DNS Answer-Based Features 39
3.7.3 Time to Live (TTL) Value-Based Features 40
3.7.4 Domain-Name-Based Features 40
3.8 System for Analysing Multi-Purpose Payload Distribution Channels Using the DNS 41
3.8.1 Analysing Query and Response Patterns 43
3.8.2 Payload Distribution Detection 44
3.8.3 Filtering 46
3.9 Conclusion 47

4 Analysis **48**

4.1 Introduction 48
4.1.1 Tools and Resources Used 48
4.1.2 Datasets 49
4.1.3 Sources of Data Enrichment 49
4.1.4 Passive DNS Database 51
4.2 Extracting the Passive DNS Dataset 51
4.3 Pre-Analysis Filtering 52
4.4 Overview of the Approach and the Analysis 54
4.5 Analysis 55
4.5.1 DNS Query and Response Pattern Analysis 55

4.5.2 Detecting the Payload Distribution Channels . . . 58
4.6 Malware Families Under Consideration . . . 60
4.6.1 Wekby . . . 62
4.6.2 DNSMessenger . . . 63
4.7 Botnet . . . 67
4.7.1 Targets . . . 71
4.7.2 Infrastructure Analysis . . . 72
4.7.3 Cobalt Strike . . . 72
4.8 Analysis of the Other Domains . . . 72
4.9 Conclusion . . . 75

5 Results and Discussion 76

5.1 Introduction . . . 76
5.2 Background . . . 76
5.3 Answering the Research Questions . . . 78
5.3.1 Research Problem . . . 78
5.3.2 Research Sub-Questions . . . 79
5.3.3 Multi-Purpose Payload Distribution . . . 81
5.4 Further Discussion . . . 81
5.4.1 Passive DNS Data . . . 81
5.4.2 Analysis . . . 81
5.4.3 Advanced Persistent Threats . . . 83
5.5 Limitations of this Research . . . 83
5.6 Conclusion . . . 84

6 Conclusion **85**

6.1 Introduction 85

6.2 Dataset 86

6.3 System for Analysing Payload Distribution Channels 86

6.3.1 Analysing Query and Response Patterns 87

6.3.2 Payload Distribution Detection 87

6.4 Summary of Findings 87

6.5 Future Work 88

6.6 Conclusion 90

List of Figures

2.1 An Example of a Zone File 13

2.2 Iterative and Recursive Queries for example.com (Adapted from Aitchison (2011)) 14

2.3 DNS Tunnelling (Adapted from Merlo et al. (2011)) 15

2.4 Comparison Between a Normal Network and a Fast-Flux Network 18

2.5 DNS Cache Poisoning Attack 20

2.6 An Example of a Domain Generation Algorithm (from Sood and Zeadally (2016)) 22

2.7 Passive DNS. 25

2.8 How this Research Relates to the Wider Field of DNS-Based Attacks. . . . 28

3.1 System Overview (Adapted from Kara et al. (2014)) 42

4.1 Details of TXT Records for Domain oakwoodsys.com using Splunk 52

4.2 Python Query on TXT Records 53

4.3 DNS Entry with an Empty TXT Record 54

4.4 Access Count of TXT Records for the .de ccTLD 56

4.5 Output of the $ dig +trace s01.1yf.de Command 57

4.6 Domains that use dns.resolution.de as a Name Server 57

4.7 Alexa Top 1000 Resource Record Distribution 59

4.8 Malware Domains Resource Record Distribution 59

4.9 Resource Record Count in the Top 20 Shady TLD domains 60

4.10 Search for the "www" in TXT records 64

4.11 Search for "stop" in TXT Records 65

4.12 Search for "mail" in TXT records 65

4.13 Search for "idle" in TXT Records 66

4.14 Content Delivery Network 68

4.15 WHOIS Information 69

4.16 IP Address Resolution 69

4.17 IP Address Resolution 70

4.18 Geolocation 72

List of Tables

2.1 Fields/Labels in a Resource Record 11

2.2 Resource Record Types 12

Chapter 1

Introduction

1.1 Background

This chapter discusses the background to the topic, and gives a brief outline of previous works that have been conducted around the subject of malicious payload distribution using the Domain Name System (DNS). A gap in the existing literature is identified leading to the formulation of the research problem. After the research problem is identified, the problem statement helps to formulate the research question. The research question and the sub-questions are the basis of this research. The chapter concludes by giving the motivation and goals of undertaking this research on malicious payload distribution channels that use the DNS.

1.2 Context of Research

Previous research has shown that botnet masters use different methods to remain hidden and avoid detection from various technologies that are put in place to detect anomalies. Various communication protocols have been misused in the past, and the DNS has become a target for botnet masters in conducting their malicious activities (Soltani et al., 2014). It is essential to analyse how DNS can be used as a channel to distribute malicious payloads.

Cybercriminals use the DNS to breach highly protected networks, steal and exfiltrate sensitive information without being detected by security controls put in place by organisations (Mc Carthy et al., 2016). According to Liu et al. (2017), DNS traffic is considered

harmless, is often not monitored or less effort is put in monitoring this traffic and is allowed to bypass monitoring measures being implemented. This allows botnets and certain malware families to establish covert communication channels that can be used to send or receive data and also distribute malicious payloads using the DNS queries. The architecture and vulnerability of the DNS protocol present an opportunity for cybercriminals to distribute malicious payloads thereby rendering traditional monitoring tools ineffective. The proposed research aims to analyse the use of DNS for distributing malicious payloads, and investigate methods for detecting domains and channels that are used in this way.

The DNS is a fundamental part of Internet activities which makes it attractive to botnet masters in maintaining their malicious networks. In its simplest form, the DNS is a powerful yet simple database management system that contains Internet Protocol (IP) addresses of every domain name (Koc et al., 2012). It is a service that translates Internet requests for accessing a particular domain name to a corresponding IP address. This enables Internet users to work with domain names that can be easily remembered without needing to know and remember numerical IP addresses.

As a fundamental part of Internet activities, the DNS is attracting cybercriminals who can exploit its vulnerabilities and also use it as part of their malicious network/infrastructure. The DNS can be abused, and there is substantial evidence that the DNS is not only attacked but can also be used as a fundamental component of botnets (Li et al., 2017). The DNS can be used as a covert communication channel between the command and control servers and the bots. DNS traffic bypasses traditional security monitoring giving an opportunity to botnets to send and receive data in highly protected networks by embedding certain strings in the DNS packets (Sharma et al., 2016). The simple architecture of the DNS protocol allows and facilitates the transfer of data in query and response packets. Cybercriminals can also use this facility and vulnerability to establish covert communication channels that allow them to send and distribute malicious payloads. The same channels are used by compromised hosts to communicate back with their masters. As a result, communication channels that can be used to distribute payload using DNS are established.

Payload distribution channels can be detected if DNS logs are analysed, but as previously mentioned, these are often unmonitored. It is, therefore, essential to analyse DNS activity to understand, detect and set up appropriate defence strategies against such emerging threats. This can be achieved through analysing Passive DNS datasets which are databases of actual historic DNS traffic. Passive DNS implementations can be used for security research and have been used for detecting anomalies in DNS queries (Marchal

et al., 2012).

According to Binsalleeh et al. (2014), research in the analysis of the DNS as a payload distribution channel is limited and often concentrates on specific malware families. The proposed research aims to broaden this research field and fill in the existing research gap by extending the analysis of DNS being used as a payload distribution channel to detection of multi-purpose domains that are used to distribute different malicious payloads. Multi-purpose domains are used for different malicious activities such as sending spam messages, phishing campaigns, and distribution of malware among others (Kara et al., 2014; Brumaghin, 2016)

1.3 Research Question

While the subject of malicious payload distribution channels using the DNS has been studied before, there is still a research gap when it comes to multi-purpose domains used for various malicious activities. In detecting these domains, it is hoped that the supporting network infrastructure used in establishing the covert channels will also be identified. The research question to be investigated is:

- Can domains used for distributing multiple malicious payloads using covert channels be characterised and modelled based on the data transferred using the DNS?

Sub-questions to be answered are:

- How can the abuse of the DNS be quantified and all the involved parties and infrastructure be identified?
- Can encrypted payload distribution channels that are used for various purposes be detected and monitored through analysis of DNS traffic?

1.4 Objectives

The aim of the research was to widen the research into the use of DNS for malicious payload distribution by filling in the existing research gap through analysing DNS traffic

generated by domains used to distribute multiple payloads. In answering the research questions above, the researcher's goal was to also characterise DNS messages associated with malicious networks and investigate various ways that malicious networks are using DNS to distribute attack payloads.

1.5 Approach

The research undertaking characterises the malicious payload distribution channels by analysing passive DNS traffic data, and investigating the DNS query and response patterns. The approach taken then analyses the passive DNS dataset for DNS queries and responses so that malicious payload distribution channels in the DNS communication channels could be detected. The analysis happens in two steps:

- DNS query and response pattern analysis and,
- Detection of multipurpose malicious payload distribution channels.

The passive DNS traffic was divided into different regular time segments, and aggregated for each given domain which is used for distributing multiple payloads. These malicious payload distribution domains were identified through analysing malware datasets for the given period to identify domains used for such services. The traffic identified to be containing malicious payload was then analysed further by examining the Resource Record. The identified payload was then correlated with the DNS profile obtained from known malware samples. The known malware samples were analysed in a controlled environment to observe the DNS traffic. A lot of filtering was involved so that the researcher could only deal with specific domains that are relevant to the research.

1.5.1 Dataset

To investigate the phenomenon above, it is important to analyse DNS logs and the most fitting way, in this case, is to analyse passive DNS data. Passive DNS datasets are ideal for security research because they offer the opportunity to replicate DNS queries that can be stored and used for both historical and near real-time security research (Bilge et al., 2014). They can be used to correlate with other data sets like the malware databases, and model the behaviour of malicious payload distribution channels used by botnets.

There are various implementations of passive DNS by different organisations, but the most widely used one is by Farsight Security which provides a subscription service to the passive DNS database through its Security Information Exchange initiative (Liu, 2015; Lee, 2015). Additionally, the Passive DNS data at the Computer Incident Response Center Luxembourg (CIRCL) was also used (CIRCL, 2017). Other data sets that were used to correlate the passive DNS information were sample malware datasets that are publicly available or from security vendors. Malware samples were analysed in a controlled environment so that their dynamic behaviour could be modelled.

1.6 Related Work

The DNS is a powerful yet simple system comprised of name servers that are deployed globally. Almost every Internet transaction is initiated by a DNS lookup. The DNS has become a target for exploitation due to the vulnerabilities in the protocol and system. According to Kara et al. (2014), compared to other methods used by cybercriminals to distribute payload, the DNS is an inefficient method. However, there is a growing number of malware families that are exploiting and using the DNS as an attack vector due to its wide use and availability (Zhang et al., 2014). Previous research on DNS exploits has primarily focused on specific malware families and botnets.

1.6.1 Attacks on the DNS

DNS security is a widely researched area that has been investigated from various angles ranging from security to performance. Research on DNS as a medium for distributing payload has increased in recent years. Although these studies have approached the subject from different angles, these studies can be categorised under DNS tunnelling and payload distribution channels (Xu et al., 2013; Liu et al., 2017). In addition to these aforementioned studies, there are studies that use passive DNS data in near real-time to detect malicious activity (Martinez-Bea et al., 2013; Yu et al., 2014). Most of these studies have utilised replicated passive DNS traffic to understand various methods used by botnets for communication through DNS tunnelling and fast-flux networks.

1.6.2 DNS Tunnelling

Research on the DNS being used as a channel for distribution of payload has mainly dwelt on DNS tunnelling (Sheridan and Keane, 2015; Soltanaghaei and Kharrazi, 2015; Almomani, 2018). DNS tunnelling refers to the use of the DNS as a covert channel to bypass traditional security mechanisms (Marrison, 2014; Yu et al., 2016). In DNS tunnelling, data is encapsulated into DNS packets. Research has shown that DNS tunnelling allows cybercriminals to exfiltrate data using DNS queries and receive data using DNS responses. This gives the cybercriminals the ability to distribute malicious payload although this method has limitations due to low rates of data transmission achieved when using Resource Records (Binsalleeh et al., 2014). In spite of the low transmission rates, DNS tunnelling research offers prospects of establishing covert channels in the DNS which may be used for both legitimate and malicious cases. Previous research on DNS tunnelling has traditionally focused on detecting covert channels that exhibit characteristics of DNS tunnels and has not focused on covert channels that do not reflect characteristics of DNS tunnels. Analysing strings on DNS queries might not help differentiate between legitimate and malicious DNS queries. These studies indicate promising results in detection of malicious payloads in near real-time using passive DNS data and their findings could be used to complement this research undertaking.

1.6.3 Fast-Flux Networks

Another area of research that is related to the proposed research undertaking is the use of fast-flux networks. In fast-flux networks, a different IP address is returned for each DNS query thereby mapping the same domain to multiple IP addresses (Truong and Cheng, 2016). This makes it a challenge to detect and identify the machine whose domain name is being used. Botnets use this method because it allows them to establish an additional layer of protection in front of their command and control servers (Soltanaghaei and Kharrazi, 2015; Dharna, 2017). By adding this complexity, detection becomes challenging, and by the time they are discovered and taken down, the botnets would have achieved their goals. In addition to adding delays in detection, adopting flux networks gives botnets the freedom to establish an additional layer of protection in front of command and control servers. Several studies have been done on malicious fast-flux networks and how these can be used to distribute malicious payloads (Haneef, 2016; Pomorova et al., 2016; Celik et al., 2016).

1.6.4 Domain Generation Algorithm (DGA)

The use of domains based on the Domain Generation Algorithm is another related field that can be linked to the subject under consideration. Bots randomly generate domains, and these bots query these with the aim of getting a response from one of these domains (Sharifnya and Abadi, 2015; Kumar et al., 2016). The responses from these domains usually contain malicious content or further instructions and commands. This method, called domain fluxing, enables the botnet masters to deliver their malicious payloads and gives rise to the challenge of detecting and identifying these domains. A botnet master controls and manages the botnet infrastructure and is responsible for sending and receiving instructions. Domain fluxing is generally used either as a primary or secondary evasion strategy during the distribution of malicious payload.

1.6.5 Payload Distribution Channels

Due to its architecture, the DNS protocol can be used as a payload distribution channel although it is not an efficient method due to the limits in size of each query and response. The flexible fields in the protocol can be abused for malicious payload distribution. The payload data can be stored in various Resource Records, and cached in resolvers which allows the payload to be accessible even if the command and control server is unreachable. While distributing payload via the DNS is a relatively new concept that has few legitimate use cases, some organisations have begun using DNS and DNS tunnelling as channels for enhancing their systems.

Nadler et al. (2017) argues that the DNS can be used as an alternative to conventional anti-virus update mechanisms by distributing anti-virus signature updates to hosts through the DNS. Anti-virus software vendors perform signature code updates using the File Transfer Protocol (FTP) or the Hyper Text Transfer Protocol (HTTP). Binsalleeh et al. (2014) argues that these methods involve considerable costs as they require special infrastructure on the security vendor side and that the DNS can be used as reliable and a low cost signature code update service. Authentication parameters of public Wi-Fi hotspots can also be transferred between the authenticating server and mobile devices using the DNS protocol (Kara et al., 2014). While there are legitimate use cases to distribute payloads using the DNS, cybercriminals use the same mechanism to ensure that their communication methods are less-easily detected and resilient. As a result, the DNS protocol is used by botnets to distribute, update and to issue and receive commands and instructions.

According to Binsalleeh et al. (2014), research on malicious payload distribution has mainly focused on domains that distribute a single family of malware or are used for specific purposes. Research on domains that are used for distributing multiple malicious payloads has, according to the literature reviewed, not been conducted. The goal of this proposed research undertaking is to fill the existing gap and answer the research questions proposed above.

1.7 Conclusion

This chapter briefly discussed the context of this research undertaking. The chapter discussed how DNS can be used to conduct criminal activities and various methods that have been used in the past to achieve this goal. A gap was identified in literature resulting in the formulation of the problem statement and the research questions. The chapter concluded by discussing the motivation for undertaking the research and the brief approach that was used to achieve the goal.

The next chapter will discuss, in detail, the fundamentals of the DNS and various related research undertakings that have been conducted around the subject of payload distribution using the DNS.

Chapter 2

Literature Review

2.1 Introduction

This chapter introduces fundamental concepts that are central to the work that is presented in this thesis. The chapter begins with a presentation on the DNS, highlights how the DNS protocol can be abused, and ends with an explanation of the passive DNS technique. Related work and research gaps are also discussed in this chapter.

2.1.1 Domain Name System

According to Dooley and Rooney (2017), the DNS protocol is a translation service used by the Internet protocol. To access a web page on the Internet, a request to access a domain name begins with a DNS query to obtain an IP address which corresponds to the requested domain name. Doing so simplifies the process of remembering domain names instead of remembering numerical IP addresses.

The DNS protocol utilises the Fully Qualified Domain Name (FQDN) to specify the location of a computer in the hierarchy of the DNS. Munkhbaatar et al. (2017) argue that the FQDN can be likened to a tree with multiple branches and each branch being ordered in a hierarchical format. To illustrate the hierarchical format above, an example of the `ross.ru.ac.za` FQDN will be used. This FQDN has multiple parts that are separated by dots with the rightmost part (`za`) being the highest level and the leftmost part (`ross`) being the lowest level of the hierarchy. Accordingly, the highest level/the rightmost part

(`za`) is considered as the Top-Level Domain (TLD) or country code Top-Level Domain (ccTLD), while `ac.za` is considered as the Second-Level Domain (2LD), and `ru.ac.za` is considered as the Third-Level Domain (3LD). According to Alenazi et al. (2017), any part of the domain or label that comes below the Second-Level Domain is usually considered as a subdomain of the Second-Level Domain. Below these levels, the next component/label in the domain name is used to designate a particular function or host server. Therefore, `ross` is intended to perform only the implied function.

According to Liska and Stowe (2016), data used to respond to DNS queries for a particular domain name is stored in authoritative name servers called Start of Authority (SOA). These authoritative name servers store original information about a domain in a text file called a zone file. Dooley and Rooney (2017) argue that while these authoritative name servers store information for zones they are responsible for, these servers are usually configured to be master name servers with the slave name servers configured and set up to respond to any DNS query. A DNS zone is part of a hierarchical domain name structure and is often a single domain. This makes zone files a fundamental component of an authoritative zone in the DNS hierarchy as it outlines services that run under a specific domain (Anagnostopoulos et al., 2013). A zone file is comprised of entries or mappings between IP addresses, domain names and other resources. These entries in a zone file are organised using a textual representation called a Resource Record (RR). Each line in a zone file is a text description that defines a single RR. A zone file for a 2LD name usually has many Resource Records that are used for various purposes

Each Resource Record consists of five fields, as shown in Table 2.1, that are separated by white space/tabs.

Table 2.1: Fields/Labels in a Resource Record

Name	Time to Live (TTL)	Record Class	Record Type	Record Data

These fields or components are briefly defined below:

- Name: This field is in the FQDN form but may be left blank, in which case the record automatically inherits the name from the previous record.
- Time to Live (TTL): This value specifies the time period after which DNS servers can discard cached responses and perform new queries to obtain up-to-date information. If this value is not specified, the DNS uses the global TTL value that is defined at the top of the zone file instead.

- Record Class: This field defines namespaces of the record information that are used for various purposes within the DNS protocol. According to Liu et al. (2016), the most commonly used namespace is that of the Internet, indicated by the IN parameter, though other namespaces are in existence and are used in some instances.

- Record Type: This field is an abbreviation of the type of information stored in the record data field (the last field). This record type field specifies the type of information that will be carried by the DNS message. Table 2.2 lists some of the record types that were considered in this research undertaking.

Table 2.2: Resource Record Types

RR Type	**Description**
A/AAAA	IPv4/IPv6 address
NS	Nameserver
MX	Mail server
TXT	Text information associated with a name
CNAME	Canonical name or an alias name

- Record Data: The record data field consists of one or more information components, depending on each record type. This is essentially the response information assigned to a Resource Record name.

In a zone file, Resource Records may occur in any order, with a few exceptions (Jin et al., 2015). According to Kalafut et al. (2011), the zone file may contain comments by preceding the text in the zone file with a semicolon. The semicolon may be placed at the beginning of the line, or on a blank line, or after the last field of any line. The zone file may also contain directives marked with a keyword and preceded by a dollar sign character ($) with the most prominent keyword being the $ORIGIN keyword (van Beijnum, 2006). This keyword indicates where the zone starts in the DNS hierarchy.

Figure 2.1 shows an example of a zone file for the domain `example.com`.

```
$ORIGIN example.com.     ; designates the start of this zone file in the namespace
$TTL 1h                  ; default expiration time of all resource records without their own TTL value
example.com.  IN  SOA   ns.example.com. username.example.com. ( 2007120710 1d 2h 4w 1h )
example.com.  IN  NS    ns                    ; ns.example.com is a nameserver for example.com
example.com.  IN  NS    ns.somewhere.example. ; ns.somewhere.example is a backup nameserver for example.com
example.com.  IN  MX    10 mail.example.com.  ; mail.example.com is the mailserver for example.com
@             IN  MX    20 mail2.example.com. ; equivalent to above line, "@" represents zone origin
@             IN  MX    50 mail3              ; equivalent to above line, but using a relative host name
example.com.  IN  A     192.0.2.1             ; IPv4 address for example.com
              IN  AAAA  2001:db8:10::1        ; IPv6 address for example.com
ns            IN  A     192.0.2.2             ; IPv4 address for ns.example.com
              IN  AAAA  2001:db8:10::2        ; IPv6 address for ns.example.com
www           IN  CNAME example.com.          ; www.example.com is an alias for example.com
wwwtest       IN  CNAME www                   ; wwwtest.example.com is another alias for www.example.com
mail          IN  A     192.0.2.3             ; IPv4 address for mail.example.com
mail2         IN  A     192.0.2.4             ; IPv4 address for mail2.example.com
mail3         IN  A     192.0.2.5             ; IPv4 address for mail3.example.com
```

Figure 2.1: An Example of a Zone File

In its simplest form, the DNS protocol is a simple database lookup with queries and responses. The queries are initiated on the host machine by a simple resolver known as the stub resolver by delegating the query to a local DNS resolver defined in the host machines settings. The local DNS resolver will then interact with other DNS resolvers in the hierarchy until it receives the answer and response to the query.

To illustrate the scenario above, a typical user types an FQDN into the web address bar of a web browser to begin browsing the Internet. When the user has finished typing the FQDN and pressed the appropriate button to initiate the browsing session, the stub resolver on the host computer sends a query to the local DNS resolver. If the local DNS resolver has a cached copy of the records then it responds to the host machine with an answer to the query but if it does not have a cached copy, it then begins querying the DNS hierarchy beginning with the root DNS servers. DNS root servers form the backbone of the DNS infrastructure and contain information about TLD authoritative name servers. Using the TLD part of the query, the DNS root servers will respond to the DNS resolver with the corresponding IP address of the name server. The DNS resolver will then query this name server to obtain the IP address and information about the name server that has authority over the 2LD part of the query. This process will go on recursively until the final zone is reached. The final zone will contain the response to the original DNS query. The scenario described above is highlighted in Figure 2.2.

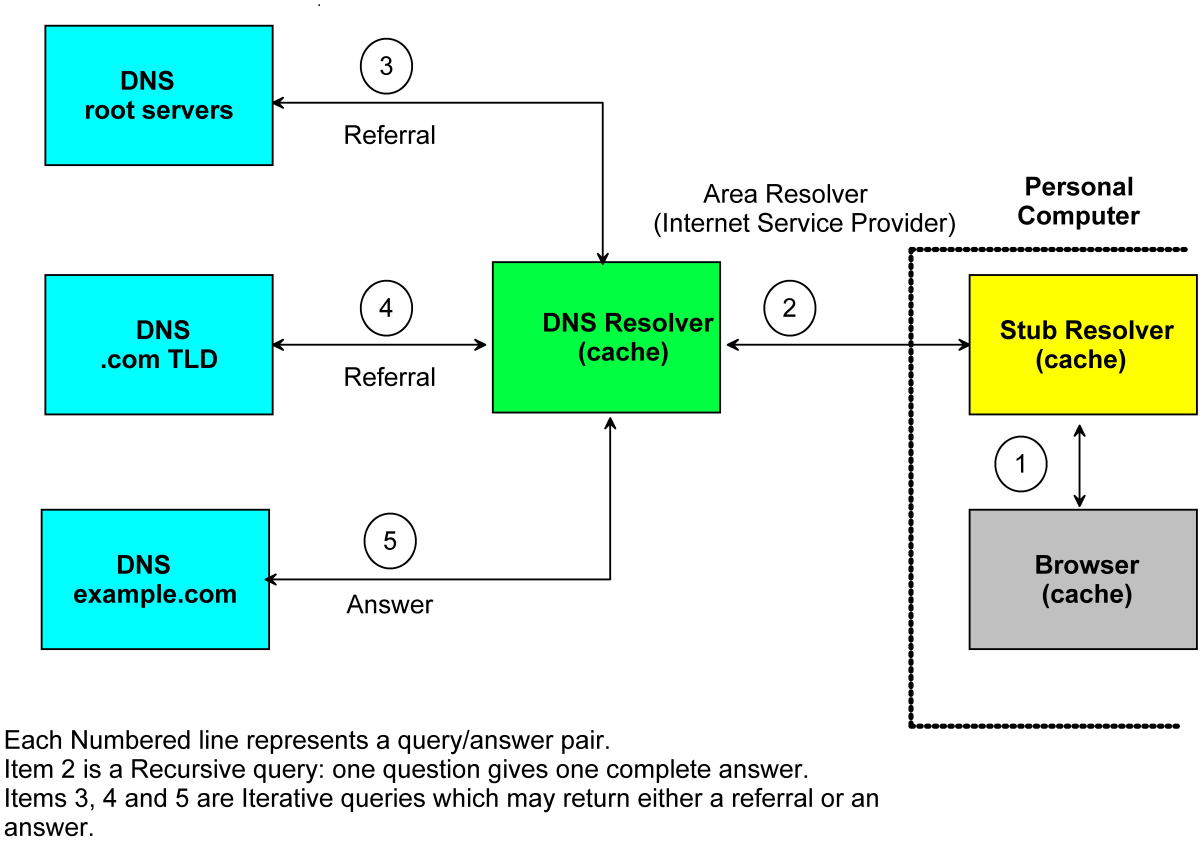

Figure 2.2: Iterative and Recursive Queries for example.com (Adapted from Aitchison (2011))

2.2 The Abuse of the DNS

According to Soltani et al. (2014), the DNS has been and is a target for criminals for different purposes. Due to the simplicity and the underlying architecture of the DNS, the protocol has been used for malicious purposes by various actors. Sharma et al. (2016) noted that whatever the form of abuse of the DNS, cybercriminals use the DNS to achieve malicious goals. Based on the form of abuses in the past, Kara et al. (2014) categorised these DNS abuses into system-level and protocol-level abuses. The following sections will explore and discuss previous work and research that have been undertaken in detecting and mitigating DNS abuse at system and protocol level.

2.2.1 Protocol-Level Abuse of the DNS

Protocol-level DNS abuses target the DNS protocol and exploit the weaknesses in the architecture and the way the protocol is designed. Li et al. (2017) argue that criminals are able to establish covert communication channels using the DNS because the architecture of the DNS protocol is naïve and allows simple tweaks to the protocol. These abuses, according to Nadler et al. (2017), allow attackers to exfiltrate and transfer data using packets in the DNS query and response messages. Malicious networks can also be created by pointing multiple compromised computers to a single domain (Liu et al., 2016). From the literature reviewed, DNS Tunnelling and Fast-Flux networks were identified as protocol-level abuses of the DNS.

1. **DNS Tunnelling**

   According to Kara et al. (2014), distribution of payload (for malicious or non-malicious purposes) using the DNS as a medium of communication is relatively new. Nuojua et al. (2017) note that research activities on the topic of payload distribution using the DNS are relatively limited and scattered. Although these research activities are dispersed, they can be categorised into the DNS tunnelling category, and the malicious payload distribution channels using the DNS protocol category. Figure 2.3 shows how DNS tunnelling can be used to bypass network security devices and communicate with rogue DNS resolvers (Malicious traffic is bidirectional).

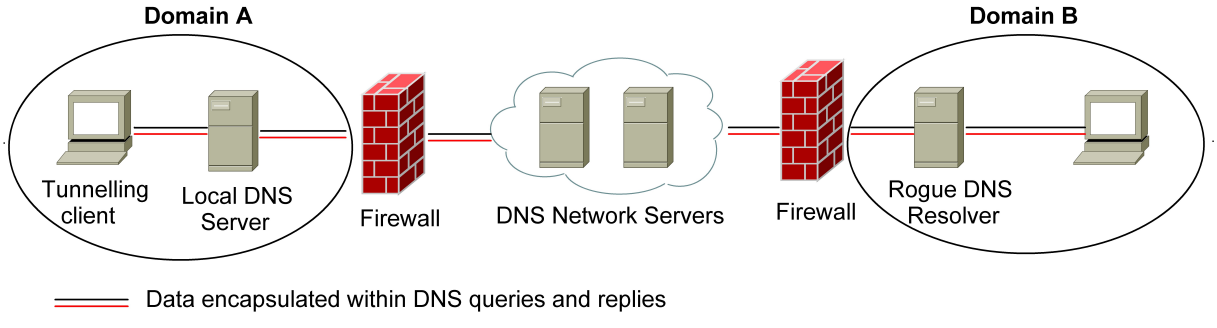

Figure 2.3: DNS Tunnelling (Adapted from Merlo et al. (2011))

From the literature reviewed, Dietrich et al. (2011) were the first to discuss botnets that use the DNS as a channel for communicating with the command and control servers. They claim, to the best of their knowledge, to be the first to analyse DNS as

a carrier of command and control messages. They discovered and reverse engineered a botnet called Feederbot that utilises the DNS to carry its command and control messages. Feederbot exfiltrates data using labels of domain names in the DNS query and infiltrates malicious attack payloads through response packets. To illustrate this, a 3TLD label and subsequent top-level domain labels could be instructions or actual data being exfiltrated. According to Kara et al. (2014), the detection method used by Dietrich et al. (2011) is limited to the detection of aggressive DNS tunnels and is ineffective for malware families, like the MortoWorm, that utilise more resilient mechanisms to receive attack payloads through DNS instead of using DNS tunnelling (Mullaney, 2011). Kara et al. (2014) argue that the study by Dietrich et al. (2011) assumes that there will be a significant amount of traffic whereas some malware families actually use the DNS to receive limited amounts of attack payload. In some instances, the payload sent is not encoded but is sent in clear text records (Nadler et al., 2017).

In the analysis of stealthy communication channels, Butler et al. (2011) introduced the concept used by stealthy botnets to create covert communication channels. They designed a covert command and control architecture that has two modes: one mode supports unidirectional communication that pulls the attack payload while the tunnel mode supports a bidirectional channel between the bot and the command and control server. They also discovered some mechanisms used by some malware families to increase the stealthiness of these channels making it difficult to detect them from the perspective of a compromised host: DNS Tunnels and covert channels. In addition to this, Raman et al. (2012) proposed a method that uses DNS tunnels as a technique for network penetration and distribution of attack payload.

According to Nuojua et al. (2017), DNS tunnelling is a widely researched area that has gained a lot of interest in academia due to its ability to offer multiple ways of establishing covert channels using the DNS. DNS tunnelling studies in the information security field have focused on different perspectives ranging from detection of such channels using through analysis of DNS traffic (Nuojua et al., 2017; Shafieian et al., 2017; Sammour et al., 2017), ways of improving detection mechanisms (Aiello et al., 2015) and techniques that can be used to protect and mitigate threats posed by DNS tunnels (Hands et al., 2015; Dooley and Rooney, 2017). These research undertakings present innovative ways for detecting DNS tunnels. Binsalleeh et al. (2014) note that the biggest challenge with malicious payload distribution channels is that they usually do not show characteristics of DNS channels because of the relatively low volume of upstream data. As a result, traditional analysis and methods

used to detect DNS tunnels are not sufficient as they may fail to show the difference between malicious and regular DNS queries.

There are various DNS tunnelling tools that are in existence (Nadler et al., 2017) and a number of ways of detecting such tunnels and the tools used (Sheridan and Keane, 2015). These detection methods are usually based on a set of rules or signatures used by intrusion detection systems (Xu et al., 2013), and behaviour and machine learning (Aiello et al., 2015; Engelstad et al., 2017). According to Alieyan et al. (2017), rule and signature-based detection mechanisms are usually prone to and plagued by false positives. Several other studies have also discussed the concept of DNS tunnel analysis from a performance perspective (Aiello et al., 2013). Sheridan and Keane (2015) argues that DNS tunnels often introduce some performance concerns due to limitations of DNS packet sizes. These DNS tunnelling mechanisms, however, create a significant amount of traffic overhead (Nadler et al., 2017). Finally, Sammour et al. (2017) investigated new ways that criminals are using to avoid detection when using DNS tunnels. They concluded that behaviour and machine learning are the best ways to detect such covert communication channels. They further conclude that these new DNS tunnelling techniques can be detected accurately using analysis of network traffic flow and can be studied just like any other covert communication channels.

According to Kara et al. (2014), Boolean data can also be exfiltrated using fixed values in the subdomains of the DNS query and by timing queries in a certain way. DNS packets can also be piggy-backed without altering the structure of the packet (Sheridan and Keane, 2017). This method is an alternative DNS tunnel whose implementation differs from the traditional implementations of DNS tunnels. Several studies have been conducted on such alternative DNS tunnel implementations and are based on the possibility that different covert channels exist within the DNS (Binsalleeh et al., 2014).

2. **Fast-Flux Networks**

According to Chahal and Khurana (2016), fast-flux networks are a result of a different set of IP addresses being returned for each DNS query. As a result, a single domain name is mapped to multiple IP addresses which makes it difficult for the machine behind that domain name to be detected. The fast flux botnet operator hosts a domain name on one of these IPs and then, after a short period of time, rotates the domain to another IP address in the group in order to evade detection and maintain its infrastructure. The basic concept of a fast flux network is having

multiple IP addresses associated with a domain name, and then constantly changing them in quick succession. Most machines that make up this type of network are not actually responsible for hosting and downloading malicious content for victims. This task is reserved for a few machines that act as servers of this malicious content; the rest just act as redirectors that help to mask the real addresses of these systems controlled by criminals. They argue that criminals are constantly adapting fast-flux networks as this gives them the ability to establish an additional layer of protection in front of the command and control servers. With the passage of time, fast-flux networks have become complex making it even more difficult to detect and trace them (Soltani et al., 2014). Even when eventually detected, by that time the criminals would have caused damage and achieved their goals before the domain is taken down. Figure 2.4 shows a comparison between a normal network and a fast-flux network. The compromised host forwards queries to the fast flux infrastructure each time a query is made and the response is then sent to the client.

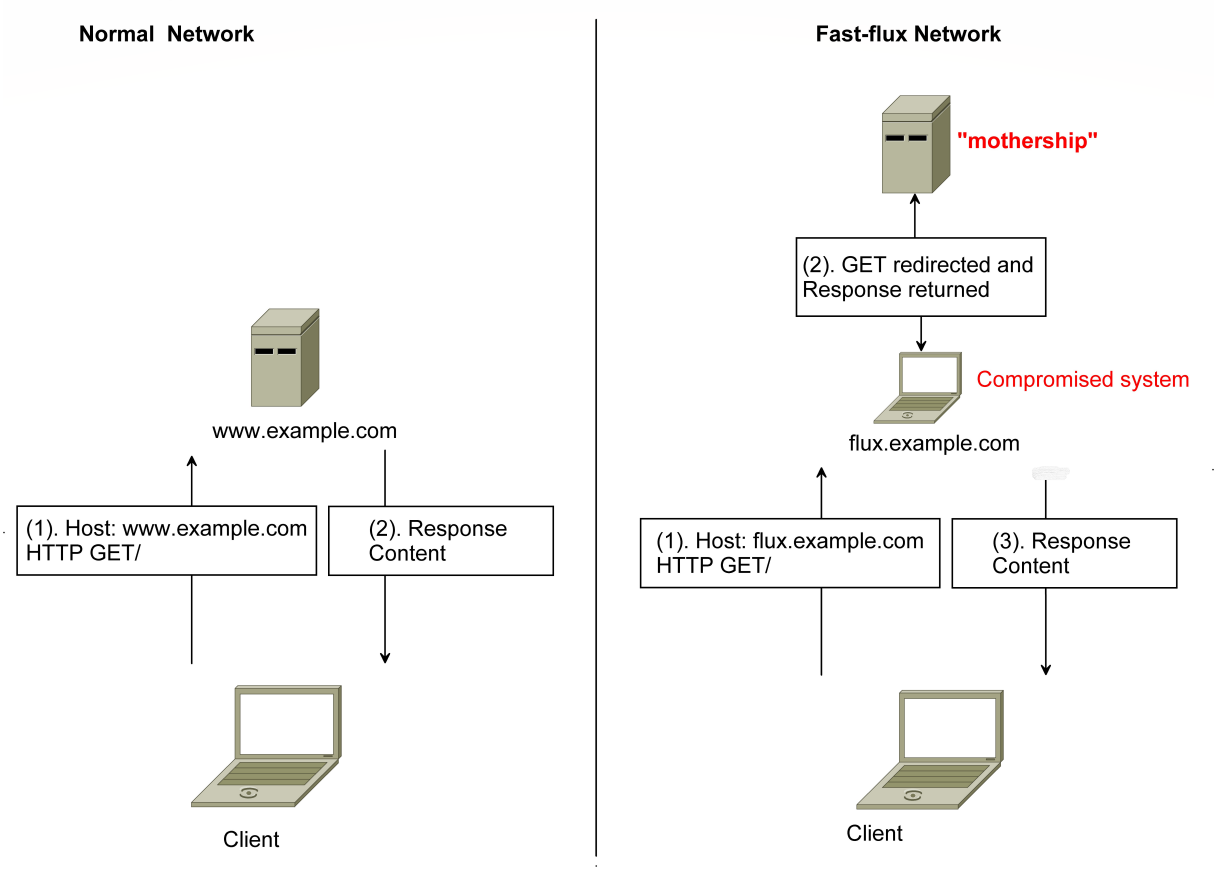

Figure 2.4: Comparison Between a Normal Network and a Fast-Flux Network

The aim of fast-flux networks is to map FQDNs to multiple IP addresses and swap

these IP addresses at a high frequency. The swapping is achieved through setting the TTL field in a DNS Resource Record to a very short time and using IP addresses in a round-robin manner. Another layer of security is added to ensure failover. This is achieved through the use of blind proxy redirection. The controlling element is the mothership which is equivalent to the command and control servers used by most botnets. Figure 2.4 shows a comparison between a single-flux network and a normal network. In a single flux network, a victim browsing `http://flux.example.com` will actually be communicating with a fast-flux network redirector. This redirector then forwards the requests to the targeted domain. Single-flux networks have the ability to change front end nodes DNS records frequently so that even if a flux redirector system is taken down, other compromised systems quickly take over the function. Double-flux networks provide an additional layer of redundancy by continuously changing both the authoritative Name Server records for the malicious domain and DNS A record set and advertising these to the fast-flux network service.

Several studies have been conducted on the analysis and detection of fast-flux networks (Soltani et al., 2014; Kwon et al., 2016; Fu et al., 2017). These studies have mainly dwelt on the characteristics of fast-flux networks and how these networks can be detected. Analysis of passive DNS data does not limit the detection of such networks to those that are already known and gives the ability to detect new networks.

2.2.2 Abuse of the DNS at System-Level

The DNS is a simple system composed of globally organised name servers. Every Internet transaction is initiated by a lookup on the DNS and criminals use every tool that is available to them to exploit all the vulnerable parts of the system. According to Lu et al. (2016), the DNS has a number of yet to be discovered vulnerabilities that exist at every layer in the DNS. These unknown vulnerabilities present an opportunity for criminals to abuse the DNS system whenever they can (Marrison, 2015).

Studies have been conducted on cache manipulation attacks that rely on caching DNS information in the hierarchy of the DNS (Mohan et al., 2015; Wu et al., 2015; Klein et al., 2017). These studies have ranged from methods used to poison DNS caches (Hussain et al., 2016) to ways of detecting cache poisoning attacks by scanning DNS resolvers (Wang, 2014). DNS Cache Poisoning, (DNS Spoofing), is a cyber-attack that exploits vulnerabilities in the domain name system (DNS) by diverting Internet traffic away from

legitimate servers and towards fake ones. DNS cache poisoning enables an attacker to pollute the data in DNS servers with bogus information that re-routes traffic to the attackers sites by changing data in the DNS to point to their IP address. Figure 2.5 shows a DNS cache poisoning attack. In addition to these vulnerabilities that exploit the DNS cache system, Kikuchi and Arimizu (2014) studied a newly discovered vulnerability which allowed domains that had been deleted to be kept alive. These vulnerabilities that manipulate the cache system in the DNS, highlight serious concerns and issues about the architecture of the DNS and its implementation. These issues allow attackers to resolve a domain which the attacker owns, using an authoritative name server effectively enabling the cache to be kept alive in open resolvers that are being targeted.

There are solutions, however, to mitigate DNS cache poisoning attacks and these involve forcing caching resolvers to utilise mixed case encoding (Mohan et al., 2015). To circumvent this implementation, the attacker has to guess the encoding used in order to access cached data and update the cache with harmful response data. Although this may be an efficient method, the length of the domain name has an impact on the effectiveness of the implementation. It would be easy to guess the mixed-case domain name if it is short (Mohan et al., 2015).

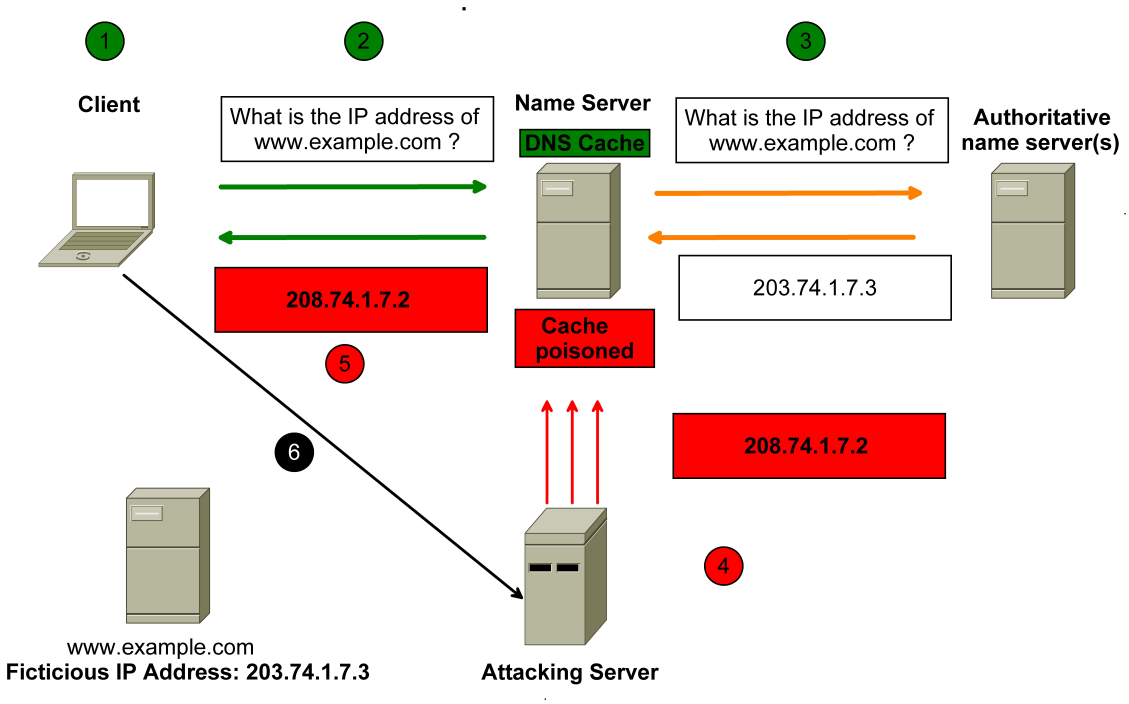

Figure 2.5: DNS Cache Poisoning Attack

In addition to the above abuses, Trevisan et al. (2017) studied the impact of rogue resolvers

in the DNS hierarchy. These resolve domain names to malicious IP addresses that can be under the control of the attackers. This attack targets stub resolvers instead of caching resolvers in cache poisoning attacks. This effectively alters DNS settings on a compromised host computer to use rogue resolvers in delegating DNS queries. As a result, legitimate domains are mapped to IP addresses that are used for malicious purposes.

Further studies on detecting DNS-based botnets reveal that malicious domains that have just been registered have distinct features when compared to regular domains. These domains resolve to particular IP address ranges which are under the control of the attackers and the lifetime of these malicious domains is relatively short since these domains are used for a particular short-lived purpose. Liu (2016) observed that domains used for malicious purposes are also found in authoritative name servers. Previously used methods to detect such domains could detect only those malicious domains that are similar to ones that have been previously discovered. There have been proposals around algorithms that can be used to detect malicious activities based on DNS queries (Fukuda et al., 2017). These proposals are focused on the similarity of queries. However, the proposed work in this research undertaking focuses on the patterns in the DNS queries and responses, and on the activities of the DNS zone.

Recent emerging studies on system-level abuses of the DNS have focused on the domains that are based on the Domain Generation Algorithm (DGA). These domains are randomly generated domains (Kumar et al., 2016) and botnets send queries to them hoping that they are already registered. Criminals also register their own set of domains, in the same manner, making it difficult to trace and find a malicious domain among numerous domains that are registered almost every day. This method, called domain fluxing, has received a lot of attention in academia with a lot of research undertakings focusing on specific malware families that use this approach (Sood and Zeadally, 2016; Bertino and Islam, 2017). According to Sood and Zeadally (2016), domain generation algorithms are either script-based or binary based with the latter being the most common. Figure 2.6 shows the execution flow of a binary-based DGA. When a user visits a malicious domain, the malware initiates the inbuilt DGA module to pseudo randomly create website domains if the binary code in the infected computer system is executed. The attacker then queries these algorithmically generated domains (AGDs) until a communication channel is established with the server that is managed by the attacker (Sood and Zeadally, 2016).

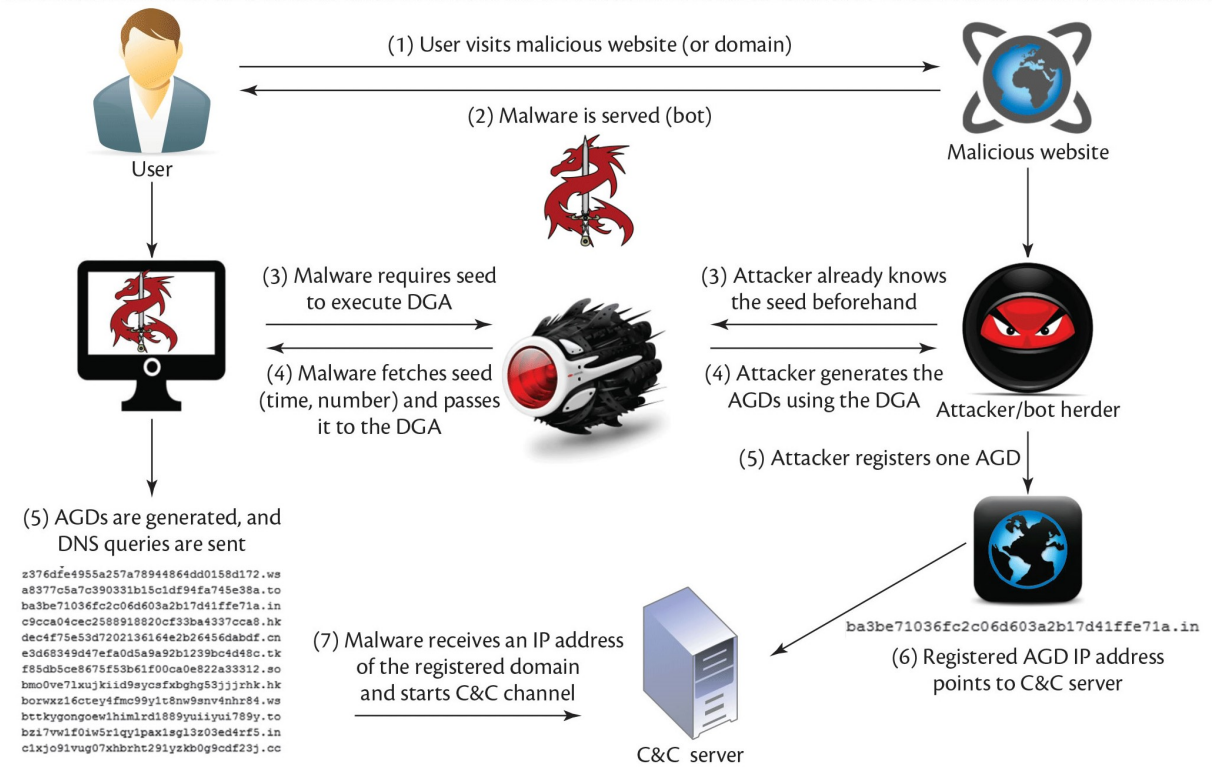

Figure 2.6: An Example of a Domain Generation Algorithm (from Sood and Zeadally (2016))

2.2.3 DNS Amplification Attack

A DNS amplification attack is a reflection-based distributed denial of service (DDos) attack (Anagnostopoulos et al., 2013). The attacker spoofs look-up requests to domain name system (DNS) servers to hide the source of the exploit and direct the response to the target. Through various techniques, the attacker turns a small DNS query into a much larger payload directed at the target network.

The attacker sends a DNS look-up request using the spoofed IP address of the target to vulnerable DNS servers (Anagnostopoulos et al., 2013). Most commonly, these are DNS servers that support open recursive relay. An open DNS server is a DNS server which is willing to resolve recursive DNS lookups for anyone on the Internet. This is similar to the better known “open relay” term used about mail servers which will forward e-mails (including spam and viruses) for anyone. When a DNS server resolves a recursive DNS lookup, it tracks down information about a domain name hosted on some other DNS server somewhere else on the Internet (a recursive process involving several other

DNS server). Unfortunately hackers have also found this feature useful in performing DNS amplification attacks. The original request is often relayed through a botnet for a larger base of attack and further concealment. The DNS request is sent using the EDNS0 extension to the DNS protocol allowing for large DNS messages. It may also use the DNS security extension (DNSSEC) cryptographic feature to add to the size of the message (van Rijswijk-Deij et al., 2014).

This requires that requests be broken down for transmission and then reassembled, requiring further target network resources. A botnet's many amplified requests enable an attacker to direct a large attack with little outgoing bandwidth use. The attack is hard to protect against as it comes from valid-looking servers with valid-looking traffic.

Proposed methods to prevent or mitigate the impact of DNS amplification attacks include rate limiting, blocking either specific DNS servers or all open recursive relay servers, and tightening DNS server security in general (Krämer et al., 2015).Rate limiting is used to control the rate of traffic sent or received by a network interface controller and is used to prevent Denial of Service attacks. Common ways to prevent or mitigate the impact of DNS amplification attacks include tightening DNS server security, blocking specific DNS servers or all open recursive relay servers, and rate limiting. However, these methods do not eliminate attack sources, nor do they reduce the load on networks and switches between name servers and open recursive servers. Also, blocking all traffic from open recursive servers can interfere with legitimate DNS communication attempts. Source IP verification stops spoofed packets from leaving the network.

2.2.4 Security Measures Put in Place to Protect the DNS from Abuse

There are various mechanisms and measures that have been put in place to protect the DNS against security threats (Hock and Kortiš, 2016), such as traffic shaping, flow filtering and prioritization. These security measures are used to protect computer systems against malware families that utilise the DNS in their implementation.

Jalalzai et al. (2015) proposes a number of measures to protect the DNS. These measures are meant to address the flaws and mitigate the impact of attacks on the DNS. The first suggestion is that of hardening the environment hosting the DNS. The DNS hosting environment encompasses all the components of the servers, from their operating systems and applications to the DNS data they store, access and manipulate. Securing

hosting environments is generally straightforward. It includes hardening the operating systems and applications, configuring access controls so only the necessary activities are permitted for authorised users and properly maintaining the environment through patching, reconfiguring, monitoring, auditing and more. In addition to this, they argue that only programs associated with the DNS should be installed on machines hosting the DNS infrastructure and only required ports should be left open. They further suggest that Name servers should not be placed on the same subnet or behind the same router to avoid a single point of failure. Organisations should have multiple DNS servers that are placed in various locations. Dooley and Rooney (2017) also recommend that recursion be disabled on master and primary DNS, and restriction of recursive queries on slave servers to mitigate spoofing attacks. When DNS recursion is enabled on a server configuration, the DNS server allows recursive queries for other domains that are actually not real master zones located on the same name server. This simply allows third-party hosts to query the name servers as they want. This setting can also increase exposure to DNS amplification attacks.

According to Jalalzai et al. (2015), one of the recommended best practices is restricting zone transference and updates to only authorised slave servers. Dooley and Rooney (2017) further suggest that the best way is to segregate DNS queries based on their origin. The origin of a DNS query could be an indicator of a compromised host, and segregating queries based on origin could enable an organisation to come up with customised defence mechanism for any attacks that could be coming from a certain source. Dooley and Rooney (2017) argue that internal and external queries should be split. External queries originate from outside an organisation whereas the internal queries come from within an organisation.

The DNS with security extensions (DNSSEC), a suite of specifications used for securing information supplied by the DNS, provides a set of security extensions. These extensions authenticate DNS data, authenticate denial of existence and data integrity (Chung et al., 2017). According to van Rijswijk-Deij (2017), these extensions also address most DNS vulnerabilities that are known, but its usage is not as widespread as expected.

2.3 Passive DNS

According to Khalil et al. (2016), Passive DNS is a system of record that stores DNS resolution data for a given location, record, and time period. Passive DNS can be used

to replicate DNS activity which can be analysed in near real-time. DNS data can be replicated and stored in historical databases that can be used for further investigations. Liu et al. (2016) argues that due to the increasing number of studies that use passive DNS information, passive DNS data will eventually become the primary and default source for analysing threats. Historically, there have been a number of studies using passive DNS to detect malicious activities happening in near real-time. Most of these studies utilise or have adapted Weimers passive DNS implementation to understand methods used by attackers (Alieyan et al., 2017). Figure 2.7 shows the process of collecting passive DNS data and the placement of the passive DNS data collector in the DNS infrastructure.

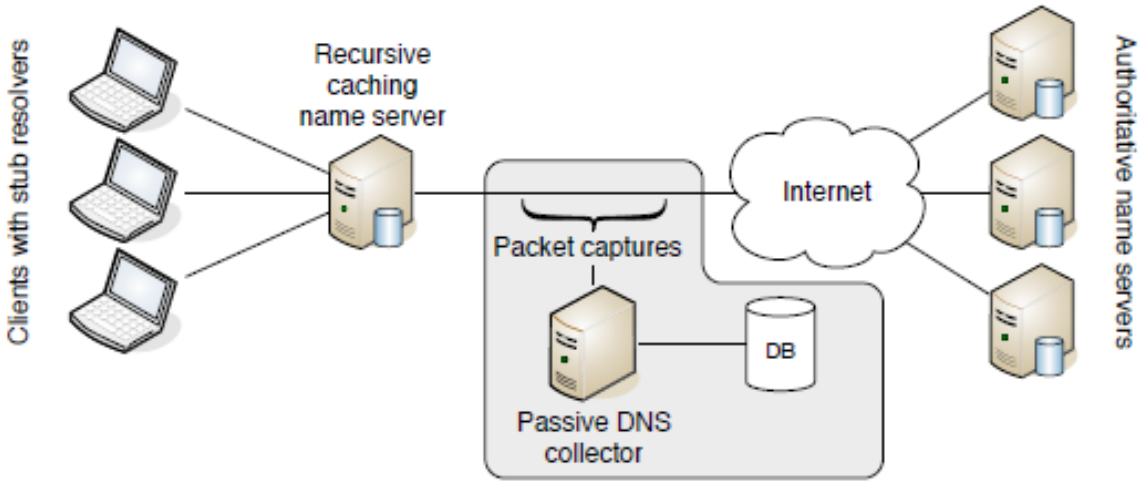

Figure 2.7: Passive DNS.

2.4 Payload Distribution Using the DNS

According to Kara et al. (2014), there are limited legitimate uses of using the DNS to distribute payload. These include antivirus updates, and mobile device authentication system for public hotspots. Nadler et al. (2017) further argues that some organisations use DNS tunnelling to distribute part of their data so as to optimise their systems. Most of these are general purpose, thus allowing various types of data exchange (e.g., web browsing, file transfer and remote desktop control). According to Shah (2017) there are legitimate and non-malicious use cases. They argue that DNS can be used for payload distribution by security products vendors as an alternative way of distributing malicious code signature updates to client software. Gordon (2013) introduced another legitimate use case using DNS to distribute payload. He implemented a mobile device authentication

system for public hotspots that used DNS as a channel for transferring authentication messages between the authenticating server and mobile devices attempting to connect.

2.4.1 Malicious Payload Distribution Using the DNS

According to Sheridan and Keane (2017), the method of communication is the most important element of an efficient and resilient malicious network. Detection of malicious activities is heavily influenced by the way/method traffic is transmitted from one end to the other. Some methods are easy to detect, whereas some methods add layers of complexity thereby increasing efficiency and resilience to the communication method. They observed that the common method used by attackers to circumvent network defence devices is to tunnel their communication through existing channels. This makes the DNS protocol a choice for attackers to exploit due to its wide usage. Since the DNS is fundamental to all Internet communication, it is a perfect target for criminals who want to distribute their malicious payload. This invites criminals to abuse the DNS for various malicious purposes. Liska and Stowe (2016) argue that the DNS is a perfect choice because DNS traffic is usually allowed to bypass network inspection as it is considered part of core Internet activity. The DNS is also chosen for malicious distribution due to flexible fields in the DNS protocol which can be abused for malicious use. Malicious payload data is usually stored in resource records and cached in resolvers so that the payload can even be accessed when the command and control servers are down.

2.5 Placement of this Research in the Wider Field of DNS-Based Attacks

The possibility of using DNS resource records as a means of distributing payload has been demonstrated through the use of DNS tunnels in a number of studies mentioned above. These studies show that the DNS tunnelling technique can be utilised to transmit information using simple encoding techniques. However, due to limitations in the size of DNS response packets and low transmission rates when using resource records, this is not an efficient way to distribute payload (Kara et al., 2014). Except in exceptional cases when Extension Mechanisms for DNS are used, the response packets are 512 bytes long (Damas et al., 2013). Extension Mechanisms for DNS increase DNS packet size to 4096 bytes so as to enhance network capacity. Increasing the DNS packet size to 4096 bytes

allows DNSSEC and larger query responses. This presents another challenge since some network defence devices, such as firewalls, do not allow DNS response packets exceeding 512 bytes. This is a drawback that has forced many malware families to find novel ways to use DNS to distribute malicious payload through the Resource Records.

According to Kara et al. (2012), research in the analysis of the DNS as a payload distribution channel is limited and often concentrates on specific malware families. The proposed research aims to broaden this research field and fill in the existing research gap by extending the analysis of DNS being used as a payload distribution channel to detection of multi-purpose domains that are used to distribute different malicious payloads.

The researcher hopes also to characterise DNS messages and investigate various ways that malicious networks are using DNS to distribute attack payloads. How this research relates to the wider field of DNS-based attacks is shown in Figure 2.8.

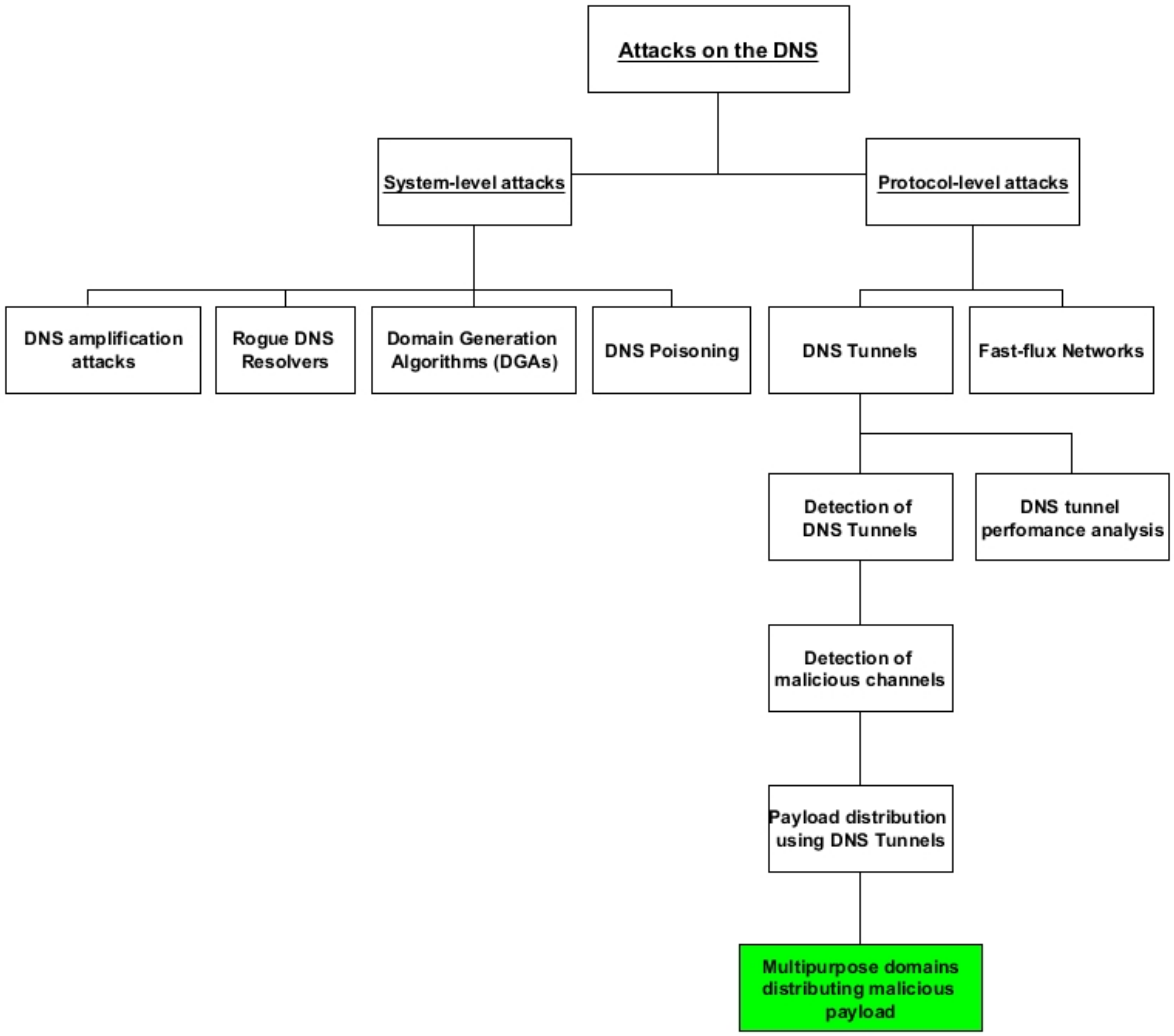

Figure 2.8: How this Research Relates to the Wider Field of DNS-Based Attacks.

2.6 Conclusion

This research analysed the detection mechanisms for multipurpose payload distribution channels that use the DNS by utilising features that are inherent in DNS and are used by malicious domains.

This chapter discussed, in detail, the fundamentals of the DNS protocol and explained various ways that the protocol can be abused. Related work was analysed and the resulting gap helped to highlight how this research relates to the wider field of DNS-based attacks.

The next chapter discusses the Research Methodology used.

Chapter 3

Research Methodology

3.1 Introduction

This chapter discusses the method and the approach that was used during the analysis and processing of data. The chapter also discusses mechanisms and methods that have been used in prior and related studies and highlights how these methods could be used to achieve the goal of analysing malicious payload distribution using the DNS.

3.2 Background

A widely used source of information for the detection of malicious domains is the DNS data (Pa et al., 2015). Malicious domains are web domain names that have been used to operate malicious networks such as botnets, and other types of malicious software (malware), and have been involved in malicious activities such as phishing and spamming. There are a number of methods and approaches that have been proposed in the past and are used to detect such domains that are used to distribute malicious payloads. The approaches can complement each other to increase the level of accuracy. These can be generally categorised as classification-based approaches and inference-based approaches.

A classification-based approach involves building a classifier using features of a domain that have been extracted from DNS data. The classifier may be enriched further using data obtained from network or host machine features. The classifier can then be further trained using known datasets of malicious and benign domains so that it can detect

new and unknown malicious domains. An inference-based approach involves building associations between domains using DNS data to highlight and determine if there are any significant connections based on a certain criteria. Should there be any association between the domains, an inference algorithm is then used to determine the maliciousness of the domain based on its association (direct or indirect) with malicious domains that are known. Maliciousness is defined as the intent to harm (King et al., 2018).

The goal of this research undertaking is to refine and define a new and stronger set of rich features and associations that can be used to improve the accuracy of detecting malicious domains, in particular, domains that are used for distributing various malicious payloads using the DNS.

3.3 Previous Studies

As mentioned in the previous chapter, there exists in the literature a significant body of research that is devoted to detecting malicious domains using statistical analysis. According to Zhauniarovich et al. (2018), this research can be classified into two categories. These two techniques are network-based or host-based detection mechanisms. Host-based mechanisms are reliant on the detection of malware signatures in applications and programs that are running on endpoint hosts, while network-based mechanisms rely on detecting malicious domains through identifying specific fingerprints and patterns in the monitored network traffic.

This research followed a network-based approach that utilised implementations that have been used in the past and relied on DNS data to detect malicious domains. Network-based approaches can be divided further into inference-based approaches and classification-based approaches. The paragraphs below will first discuss the inference-based approach and then discuss the classification-based approach.

Inference-based approaches can be used to detect malicious domains using only aggregate DNS data. Inference-based approaches can be used to complement classification-based approaches (Khalil et al., 2017). An inference-based approach is based on a simple concept or principle. The approach is based on the assumption that if a domain is strongly associated with a malicious one, it is also highly likely to be a malicious domain as well (Khalil et al., 2017). The key to achieving high accuracy and validity in this malicious domain detection approach is highly dependent on the definition of associations between

domains. In an ideal situation, for an inference-based approach to be effective, it should satisfy two conditions.

The first condition is that the link/association between domains should be accurate and relevant to the inferred maliciousness. It is reasonable to associate two domains if they are deployed and controlled by one entity. In this case, if the first domain is malicious, it is reasonable to associate the other domain with a malicious one thereby raising the probability of the maliciousness of the second domain (Manadhata et al., 2014). Conversely, it would be unreasonable to associate two domains because they have weak and irrelevant associations. Weak associations result in a high rate of false positives.

The second condition is that the defined association should have a fair coverage and should show as many appropriate links between domains as possible (Khalil et al., 2017). Using strict associations would overlook many relevant links between domains and result in a low detection rate of malicious domains and the converse is true. The ideal situation is to define associations between domains to have a high coverage and also be highly accurate. There is, however, a challenge in associating a domain with another malicious domain because of the way DNS data is collected which might limit the scope of information obtained from such DNS data. DNS data collected may not provide enough information to make informed conclusion about the maliciousness of that domain.

There are various classification-based approaches that have been proposed and implemented in the past. These implementations will be briefly explored and the relevant aspects of those approaches will be incorporated into this research methodology.

Notos (Antonakakis et al., 2010) and EXPOSURE (Bilge et al., 2011) are two of the widely used classification-based approaches that have been proposed and are used for identifying malicious domains through building classifiers that use local features obtained from passive DNS data in combination with other network information obtained from other sources such as WHOIS records. The effectiveness of these classification-based approaches is dependent on whether the local features that are used in the classification process have not been manipulated. Khalil et al. (2017), however, argue that many local features (TTL-based features and patterns) can be easily manipulated, thus making these detection techniques less effective. The success and accuracy of these approaches are dependent on gaining access to sensitive DNS queries, which though difficult to gain access to, the researcher has obtained access to and has been authorised to use for this research purpose. The DNS queries are considered to be sensitive when they contain private information that identifies the party involved. The information could compromise the identity of individuals/organisations that are making DNS queries.

3.4 Overview of the Approach

This research undertaking proposes a mechanism for detecting multi-purpose malicious payload distribution channels that use inherent DNS features that are used by both malicious and non-malicious domains. This approach aims to detect payload distribution channels that are used for distributing multiple malicious payloads through the analysis of DNS resource records and the intensity of the payload distribution.

The research adopted and adapted previous approaches that have been used to detect malicious payload distribution channels and domains, but expanded it to include those domains that are used for various malicious purposes such as spamming, phishing, botnets, among others. The approach is an adaptation of previous work by Bilge et al. (2011) and Bilge et al. (2014) who proposed a system named EXPOSURE. This research also expands on the work by Kara et al. (2014) which also identified this research undertaking as a gap that needed to be examined further. EXPOSURE was designed with the goal of detecting domains used to perform malicious activities such as botnet command-and-control, and phishing among others. EXPOSURE achieves this through passive DNS analysis and leveraging of machine learning to build detection rules which have the capability to differentiate the DNS behaviour of malicious domains from non-malicious domains (Bilge et al., 2014). The design uses four sets of DNS features that are obtained from DNS records. These features are time-based features, domain name-based features, TTL-based features, and DNS answer-based features, discussed in the following Sections.

In addition to the features proposed in the EXPOSURE model, historical malicious-site data obtained from various blacklist sources was used. This enabled the adopted approach to detect malicious domains that are dynamically mapped to new address spaces and are utilised for malicious purposes.

The assumption is that by monitoring real network traffic generated by users, some of these users are running systems that have already been infected or have malicious content in their traffic. As such, it is highly likely that these infected components will contact these malicious domains that may or may not be already blacklisted in spam blacklists and malware domain lists. By using passive DNS data to study the behaviour of known and unknown malicious and benign domains, the researcher hopes to identify and distinguish the behaviour of domains that are used for multiple purposes and thus define the maliciousness of that domain.

3.4.1 Datasets

The datasets to be used in this research undertaking are from multiple sources. This is done to check for accuracy and validity of the information or data provided by different sources. The primary dataset is the passive DNS data obtained from two organisations: FarSight Security (commercial subscription) and Computer Incident Response Centre Luxembourg (an incident response centre driven and supported by government, which offers a passive DNS data repository to the information security analysts community). However, the FarSight Security dataset was used as the primary source since FarSight Security Passive DNS data is generally considered to be the most comprehensive passive DNS data available (Noroozian et al., 2016). These datasets were queried via the Application Programming Interfaces (APIs) provided, using supplied login details.

The rationale for choosing passive DNS data is based on the following reasoning. A passive DNS system logs all DNS data, performs all analysis passively and the results are produced in the desired format. While using an active DNS system can help organisations and researchers to gain a real-time overview of malicious networks, it comes at a risk of being discovered by administrators of these networks if the organisation is actively probing the malicious network. The administrator of that malicious network could discard the requests thereby ending the analysis or usefulness of the research undertaking. However, passively observing and analysing DNS data without making active DNS queries allows the analysis to remain undetected and stealthy.

Passive DNS analysis, when combined with systems that use reputation scores, can form a powerful combination that can be used to detect multipurpose payload distribution channels (da Luz, 2013). Having a historical view of DNS can assist in making a timeline of malicious IP addresses and domains thereby assisting researchers in understanding their behaviour which can be extracted from features of the DNS. Features such as average TTL values, where these domains resolved in the past, among other attributes can be used in the analysis of DNS. Making use of this information can help in the detection of multipurpose payload distribution channels.

A Malware Database was also used for this research undertaking. For the given period being analysed, malware samples obtained from freely available malware research repositories were analysed in a controlled and sandboxed environment. The aim of this analysis was to obtain insight into the dynamic behaviour of malware, and considered only malware families that conduct their activities using TXT Resource Records.

In addition to the above, malicious domain lists were collected from various reputable sources. These malicious domain lists that were considered are:

- `http://www.malwaredomains.com`,
- `https://www.malwaredomainlist.com`,
- `https://www.phishtank.com`,
- `https://zeltser.com/malicious-ip-blocklists`,
- `http://www.surbl.org/lists`.

The above malicious domain lists, while not exhaustive, represent a good coverage of malicious activity ranging from phishing sites to botnet command-and-control servers. This list was used conservatively to construct a domain list. Preliminary checks were applied before assigning a domain as being malicious. Some of the above malicious domain lists repositories rely on ordinary users and volunteers. As a result, while most domains submitted are malicious, not all domains in those lists are malicious in nature. In addition, some third-level domains of a domain name in a Uniform Resource Locator (URL) may be malicious, the rest of the domain may not be malicious. As an example, `a.b.com` may be malicious, while `b.com` is not malicious but just benign.

3.5 Using Passive DNS to Identify Payload Distribution Channels

There are systems that have been proposed in the past that are used for identifying and detecting malicious domains using passive DNS monitoring techniques. Notos detects malicious domains by dynamically assigning a reputation score to a given domain name whose maliciousness is not yet known (Antonakakis et al., 2010). A system developed by Bilge et al. (2014) makes use of passive DNS data to detect domain names that are used for malicious activities. EXPOSURE addresses the shortcomings of Notos as it does not need a complete overview of malicious Internet activities. EXPOSURE is able to detect domains that Notos would not classify as malicious and takes less time to train. EXPOSURE is discussed in detail in Section 3.7.

3.6 General Detection Techniques

According to Wang (2016), many utilities that use DNS tunnelling do not even attempt to remain stealthy because DNS traffic is often not monitored. While many detection techniques have been proposed, there seems to be limited research that analyses the payload and the traffic at the same time. In particular there is limited research into payload distribution channels that are used for malicious purposes. The proposed detection technique analyses both the payload and the network traffic over time to characterise and understand how these payload distribution channels operate. To analyse the payload, the payload carried by the DNS was analysed for features in the requests and responses, and the behaviour that indicates tunnels that are used for distributing malicious content. In network traffic analysis, the frequency, count and other attributes of the DNS requests and responses was considered.

3.6.1 Payload Analysis

The methods and techniques that were used to analyse the payload of DNS queries and responses were borrowed from the research on Domain Generation Algorithms (DGAs). Domains generated through DGAs exhibit similar behaviour that is useful in the analysis of malicious payload distribution. The attributes below were considered in the analysis of DNS payload.

3.6.1.1 Size of Request and Response

A proposal by Engelstad et al. (2017) suggests one method, which involves analysing the size of DNS requests and responses. They proposed a technique which identifies suspicious DNS tunnelling traffic through analysing the ratio of the source and destination bytes. Another way was to look at the number of the DNS queries and responses. The assumption is that utilities that use DNS tunnels attempt to send as much data and information as possible in the DNS requests and responses. It is, therefore, highly likely that DNS queries with malicious payloads would have request and responses that have long labels and names.

3.6.1.2 Entropy of Hostnames

Butler et al. (2011) recommended a method of detecting DNS tunnels and payload distribution channels using the entropy of requested hostnames. Non-malicious, legitimate and popular DNS names often contain meaningful and dictionary words. However, Engelstad et al. (2017) argue that encoded names often have even use of character sets and have a higher entropy. Although exceptions should be noted, this is usually the case and can be used as an indicator of DNS tunnels and payload distribution channels.

3.6.1.3 Uncommon Record Types

Another method that can be used is that of searching for and identifying records that are not normally used during a DNS request and response session (Engelstad et al., 2017). The commonly used record types include the A, CNAME, MX and TXT records. These are the most common DNS records that perform actions that have the most everyday use for people. But, going past these records, there is a large list of DNS records and record types that are rarely used. Usage of these rarely used records may be an indicator of some anomaly, but there should be an investigation before a conclusion is reached.

3.6.2 Traffic Analysis

This technique involves analysing the request and response pairs over a period of time. A high frequency and a large number of DNS requests are usually an indication of payload distribution channels, and this technique would use the following features to help in the detection of payload distribution channels (Engelstad et al., 2017).

3.6.2.1 The Volume of the DNS Traffic per IP address

The most basic method is to analyse the amount of DNS traffic that is generated by a specific IP address. Due to typical DNS tunnel data being limited to 512 bytes per request, for the tunnel to send a lot of data on the channel requires many requests in order to transmit as much data and information as possible (Engelstad et al., 2017). Many continuous requests are also expected if the client is polling the host server.

3.6.2.2 The Volume of the DNS Traffic per Domain

Large volumes of traffic querying the DNS for a domain is another way that can be used to detect payload distribution channels. Payload distribution channels are most likely to set up a domain name that will distribute payload, as a result all traffic will be pointing to that domain name set up for that purpose (Engelstad et al., 2017). There are however exceptions when these channels could use different domain names thereby decreasing the volume of traffic per given domain.

3.6.2.3 Number of Hostnames per Domain

The number of hostnames per domain could be an indication that there is a payload distribution channel because utilities using DNS tunnelling can also request a different hostname on each request (Engelstad et al., 2017). This leads to a higher than usual number of hostnames compared to a non-malicious, legitimate domain name.

3.6.2.4 History of the Domain

The history of a domain can raise suspicions when analysing DNS traffic. A simple check on when an A or NS record was added can reveal useful and interesting information about whether a domain is used for malicious purposes. This information should be used in combination with other features in order to arrive at a meaningful conclusion. While this method is used primarily to detect malicious domains, it could also be used to detect malicious payload distribution channels and tunnels.

3.6.2.5 The Volume of Non-Existent Domain Responses

Excessive Non-Existent domain (NXDomain) responses can be used for detecting domain names generated by DGAs, and has been used in the past to detect certain malware families that generate huge volumes of NXDomain responses (Engelstad et al., 2017).

3.6.2.6 WHOIS-Based Features

According to Frosch et al. (2013), the date of registration provided by WHOIS domain registries can be used to calculate the age of a domain. Based on this, malicious domains,

which typically have a shorter lifespan, can be observed and differentiated from non-malicious domains. It follows that malicious domains are expected to have a lower age when compared to non-malicious domains.

3.7 Features of EXPOSURE

The DNS features needed to distinguish between malicious and non-malicious domains, and to be able to train a classifier typically require a lot of data. The EXPOSURE system identified fifteen different features that can be used to detect malicious domains (Bilge et al., 2014). Of those fifteen features, six of them had already been used in prior research and, to be precise, in the classification of malicious URLs and in the detection of malicious fast-flux domains.

3.7.1 Time-Based Features

According to Li et al. (2017), the short life, daily similarity, repeating patterns, and access ratio are the time-based features that may be indicative of malicious behaviour. The time at which a request is made is one of the components of DNS records that may appear to be insignificant by itself, however, when many requests are analysed over time, useful patterns that may be indicative of malicious behaviour may be observed (Bilge et al., 2014). Of particular note is a change in a volume or number of requests for a particular domain. The assumption is that malicious payload distribution channels would show a sudden spike in the number of requests followed by a sudden decline in the number of requests due to malicious services utilising the domain flux technique, and the resulting generated domains having a short lifespan.

The number of requests during a given time period for a particular domain was analysed by dividing the given time into fixed-lengths intervals. For each time interval, the number of DNS queries issued for that particular domain is then counted and these are converted into a time series. In other words, the collection of DNS queries for a domain are chronologically ordered as a sequence of data values. The time-series analysis is performed globally and locally. The global time-series analysis covers the start and the end of the whole period being monitored. In the local time-series analysis, the focus is on the start and end of the fixed time interval. The global time-series analysis was used for

detecting malicious domains whose behaviour would have changed for a short duration or those that have a shorter life.

A domain is considered as short-lived if it appears for a comparatively short duration. A domain that is suddenly observed in the global scope time-series but then disappears shortly after being highly active has a high likelihood of being considered as a malicious domain due to this abnormal behaviour which is associated with DGAs. The short duration makes it difficult to detect and they may even be detected only after damage has been done.

According to Frosch (2012), detecting domains with regularly repeating patterns and short-lived domains can be considered as a change point detection problem. The goal of change point detection algorithms is to identify points in a time-series where there is an abrupt change in data values. These algorithms help to output the point in time when a change is detected, and the behaviour during the selected time period. Abrupt changes in a domain can be detected by preparing a time-series representation of requests count over time. Similar behaviour can be detected by measuring the level of similarity of two time-series, and calculating the distance between the two.

3.7.2 DNS Answer-Based Features

According to Li et al. (2017), the following are features that are based on the DNS that are indicative of malicious behaviour: number of distinct IP addresses , number of distinct countries, reverse DNS query results, and the number of domains share the IP with. When a server for a domain answers a DNS query, the answer generally consists of many DNS A records that map a hostname to an IP address (Nadler et al., 2017). A domain can be mapped to multiple IP addresses, a technique usually implemented in load balancing. This allows the DNS server to cycle through multiple IP addresses in a round-robin manner returning a different IP address mapping for each query. According to Bilge et al. (2014), malicious domains also exhibit the above feature but they, however, resolve to infected hosts that are located in different Autonomous Systems and different geographic locations. As a result, attackers would use these domains that map to different IP addresses and also utilise IP addresses that are shared by different domains.

With the insight from the above, this research undertaking utilised the features proposed by Kara et al. (2014) to detect malicious domains used to distribute payload. These features to be analysed are the count of different IP addresses that resolve to a particular

domain. The other feature to be analysed is the number of various countries that these IP addresses reside in. The third feature to be analysed was the reverse DNS query results of these multiple IP addresses. The last feature to be considered is the count of distinct domains whose IP addresses resolve to a particular domain name. However, malicious domains may also resolve to IP addresses that are used by well known reputable and legitimate domains.

3.7.3 Time to Live (TTL) Value-Based Features

According to Li et al. (2017), the following features are indicative of malicious behaviour: average TTL, standard deviation of TTL, number of distinct TTL values, number of TTL change, and the percentage usage of specific TTL ranges. Each DNS record has a TTL value which specifies the time for which the corresponding response should be cached. For systems that are designed for high availability, the TTL values are set to a lower value and often use round-robin DNS. According to Andrews (1998), the recommended TTL value is between 1 and 2 days. Setting the TTL value to a lower value allows the domain to be reachable even if one IP address cannot be reached at a particular time, due to the quick expiry of TTL value and having another IP address that is reachable.

This approach enables attackers to also achieve high availability. Abnormal usage of round-robin has been used in the past to detect fast-flux networks, a feature that is commonly used by domains that distribute malicious payloads (Hachem et al., 2011). In this research undertaking, the TTL values and usage of round-robin by suspected malicious domains was also analysed as these are often used by attackers to distribute malicious payload. It has been observed in prior research that malicious domains tend to have significantly higher TTL values and TTL values change frequently compared to non-malicious domains. As such, TTL values above the recommended values may be an indication of malicious behaviour.

3.7.4 Domain-Name-Based Features

The main goal of the DNS protocol is to provide names that are human-readable, and easy to remember thereby saving users the pain of memorising IP addresses of host servers. Malicious domains also tend to try to choose domain names that can also be easily remembered and can deceive users (Bilge et al., 2014) to believe they are visiting legitimate non-malicious domains.

In some cases, attackers use Domain Generation Algorithms to come up with domain names that mimic or are similar to legitimate domain names (Bilge et al., 2014). To detect malicious domain names, this research undertaking analysed the ratio of numerical characters to the domain name length; and the ratio of the length of meaningful words/strings to the domain name length. The meaningful word or string could be a word in a dictionary. However, there are some legitimate, non-malicious domain names that do not necessarily have meaningful strings/words that are found in dictionaries. As a result, these were taken note of.

3.8 System for Analysing Multi-Purpose Payload Distribution Channels Using the DNS

The system analysed DNS queries and responses in the passive DNS dataset, with the aim of detecting multi-purpose payload distribution channels. The system is composed of a module for query and response analysis, and another for detecting the malicious payload distribution channels. The system is briefly described and discussed in the following paragraphs.

The system has a DNS query and response message analysis zone where the pattern of each channel is analysed, hereinafter referred to as the pattern analysis module. After the DNS query and response message analysis, the DNS query and response messages were also sent to a zone responsible for analysis of the presence of a potential payload distribution channel, hereinafter referred to as the payload analysis module. This zone extracted all the relevant DNS Resource Record (RR) activities for particular domains from the passive DNS data. The zone or module is responsible for determining the intensity of the payload distribution using the activities of DNS resource records. The intensity of a payload distribution channel measures the level of activity in a particular channel. Further to that, there is a need to do some filtering so that legitimate domains that resemble multipurpose payload distribution channels may be filtered out. Figure 3.1 shows the system that was developed, including the modules involved and the flow of data from one module to the other.

The following sections indicate how the pattern analysis module and the payload analysis module interact and work together in characterising and detecting multi-purpose payload distribution channels using the passive DNS data.

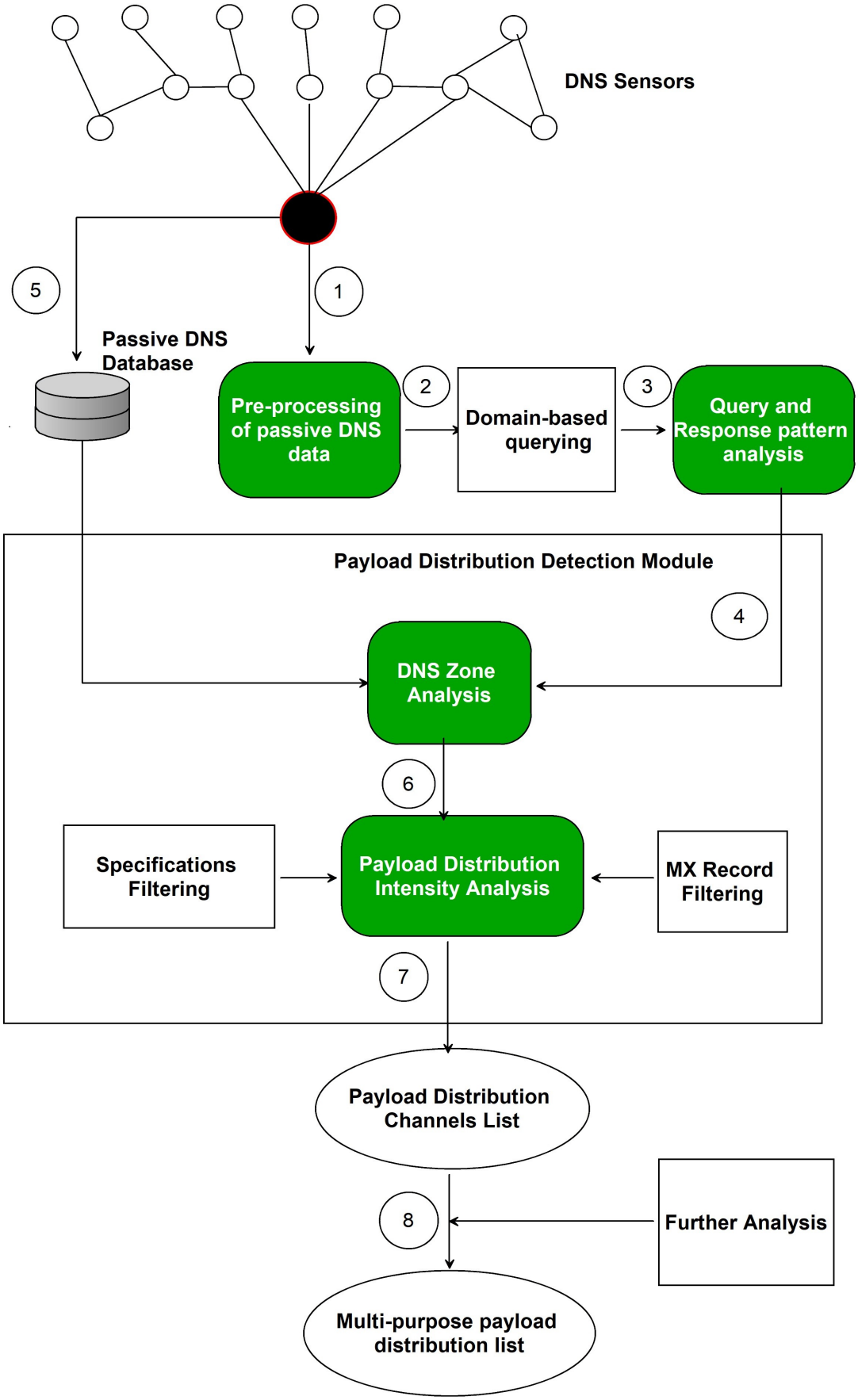

Figure 3.1: System Overview (Adapted from Kara et al. (2014))

3.8.1 Analysing Query and Response Patterns

The DNS protocol uses the query and response messages to map human readable domain names to IP addresses. Clients can query name servers for different pieces of information or fields, and the name servers respond with the required information. Using the information observed from the communication between a client and the server using DNS requests and responses, it is possible to model the relationship between the query and response messages. These messages can then be used to identify and observe different behaviours of channels that distribute payload (malicious or non-malicious).

To identify payload distribution activities in the DNS, certain parameters were analysed as they are often used for establishing DNS tunnels and channels (Kara et al., 2014). The three parameters that are used to implement payload distribution are the second-level domain, the sub-domain and the TXT record. The second-level domain is used to coordinate the payload distribution. The sub-domain is then used to transfer data from a client host to the targeted host. The response from the server is carried by the TXT record.

The server and the client pre-arrange the domain name that can be used during the session for any activity that distributes payload. The second-level domain name is agreed upon before a session is established. The other two parameters that are used for establishing the communication channel have different behaviours depending on the nature of the channel distributing the payload.

The goal of the pattern analysis module is to differentiate various behaviours (Single-to-Many, Many-to-Many, Single-to-Single, and Many-to-Single) of channels distributing payload. This is achieved through analysing the behaviour during the exchange of DNS queries and response messages. The analysis is based on the fact these channels are forced to exchange only small amounts of information within each DNS message due to the limitation in the size (Cheshire and Krochmal, 2013) of the DNS response packet (512 bytes). This can be overcome by using Extension mechanisms for DNS (EDNS), but as already noted this is not widely supported. The other factor to be considered is the fact that there is a significant number of DNS queries and responses when there is a significant amount of traffic being transported via the DNS protocol.

3.8.2 Payload Distribution Detection

After the DNS query and response patterns have been recognised, they are analysed using the module responsible for DNS zone analysis.

3.8.2.1 DNS Zone Analysis

During the lifetime of a DNS query, name servers have a central role in the processing of DNS query messages and responding accordingly with the appropriate response messages. Name servers play an important role in the DNS protocol and as such, malicious networks require access to these servers in order to be able to distribute their payload. Once a name server has been configured to be an authoritative domain name server that manages payload distribution, the cybercriminals configure the zone file of that particular domain to contain all payloads for the information/commands to be delivered through the DNS.

The module responsible for analysing the DNS zones looks at the behaviour of the domain name servers. Each Resource Record (RR) located within the zone file of a domain indicates a particular service that is associated with that domain. A passive DNS database allows the number of times a record has been queried to be presented as an aggregate which is referred to as the access count. Domain names that are used for payload distribution are expected to exhibit behaviour that is different from legitimate and regular domains.

It was expected that legitimate, normal domains would receive DNS queries for different resource records but for malicious domains that are used to distribute malicious payload, it was expected that they would be accessed solely to receive payloads that are used to launch attacks. As such these malicious domains are expected to use only particular resource records that are commonly used by malicious payload distribution channels. As mentioned previously, the most commonly used resource record is the TXT record. These malicious domains are not expected to frequently use resource records that are usually used by legitimate domains, such as the MX, A and AAA resource records. The DNS zone activities can be profiled by observing the resource records and the number of times they have been queried.

3.8.2.2 Extracting DNS Zones

Name servers are regarded as the payload distributors when investigating channels that distribute malicious payload using the DNS. The domain names can have many zones

and it is, therefore, necessary to identify those zones that are responsible and associated with the distribution of payload. A DNS query may be composed of many sub-domain labels which could be pointing to sub-zones that are in a second-level domain. After the sub-zone has been extracted, the DNS zone activities need to be profiled.

3.8.2.3 Profiling DNS Zones

Analysing resource records of a sub-zone can help to understand and determine if a sub-zone has been used for malicious distribution of payload. The activities can be measured and analysed as a function of the count of queries (access count).

3.8.2.4 Analysis of Payload Distribution Intensity

One of the ways to analyse the intensity of a payload is to assign a rating value to a domain. Ratings can be a range of values which show the increase in the intensity of the payload. Domains with higher intensity are likely to be malicious. This can, however, give uncertain information. When investigating the intensity of a payload distribution channel, domains that have been detected as malicious need to be prioritised so as to facilitate the process of investigating the intensity and level of its maliciousness. The rating values can be transformed into descriptive words like low, medium, high, among others, to make it simple to analyse.

To validate the system above, malware datasets were used to validate expected behaviour and the passive DNS data was investigated for any differences that may exist between regular legitimate domains and the malicious payload distribution channels. For regular domains, the Alexa Top 1000 sites list was utilised as these websites are used for various legitimate purposes. Malware domains used for payload distribution were extracted from various sources as mentioned previously in Section 3.4. The access count for resource records for each domain (malware domain or regular domain) was retrieved to get a measurement that could be used to gain a deeper understanding of individual resource record activity of a particular domain.

It was expected that the domains from the Alexa Top 1000 sites would receive DNS queries for various resource records since these domains utilise the DNS to enable access to various services. For malicious domains, the expectation is that they would receive huge volumes of DNS queries for the TXT records as these are the most suitable resource

record type for distributing payload. Another record type to be examined is the CNAME record type. Access to this record would be investigated as this record type is responsible for redirecting to malicious domains, and cybercriminals use this record type to maintain their malicious networks that distribute payloads using the DNS.

3.8.3 Filtering

The primary passive DNS dataset used in this research undertaking is from FarSight Security, and there are huge volumes of DNS traffic and queries processed at any given time. Processing and analysing such volumes of network traffic data without applying some filters is not practical and as such, there is a need to apply some filtering.

From the passive DNS dataset, there are two steps that were used for filtration. As mentioned in the Literature Review section, there are genuine legitimate use-cases for using the DNS to distribute payload, and these channels may behave like malicious payload channels. The most likely use-case regards the usage of TXT records for security measures for some mail servers. As these specifications are intended for mail server activities, it was expected that the zone file would have MX resource records.

Legitimate domains and services were, therefore, filtered through the activity of the MX resource record and recognition of the specification being used. The first step used to filter legitimate domains was to choose the most accessed TXT resource record for each domain and apply regular expressions based on the specification to determine if there were any strings that correspond to a particular specification. The second step involved examining the attributes of the MX resource records. Those domain names associated with MX resource record activities were excluded for further analysis, and were generally considered as being non-malicious for this research purpose. This, however, does not mean that these domains cannot be used for distributing malicious payload. These methods of filtering are meant to reduce the average number of domains to be considered in the analysis.

From the malicious domain list dataset, filtering was applied. This filtering process also involves two steps. The first step involves whitelisting popular and well-known legitimate domains that are highly unlikely to be involved in malicious activities. This whitelist is derived from the Alexa Top 1000 Global Sites list. The assumption is that the most popular websites and domains are unlikely to be associated with malicious activity because these sites usually have many visitors and or users and these domains are well monitored

and maintained. As such, by using this filtering it is expected that the volume of data being analysed would be reduced. The second filtering step involves filtering domains that are older than a specific age. This is based on the premise that malicious domains are usually uncovered or discovered after a certain period, after which they are blacklisted. As a result, cybercriminals then use DGAs to make it difficult for these domains to be detected and even when they are detected, their goal would have been achieved.

3.9 Conclusion

In conclusion, the research methodology presented above aims to provide a comprehensive analysis of the distribution channels that are used for multiple purposes through categorisation of DNS messages. The characterisation involves techniques that identify the patterns of these channels through DNS query and response pattern analysis. The channels are also identified using the resource record activities. Distribution channel intensity is then determined to highlight the level of activity on the channel that is under investigation.

The next chapter will discuss the actual analysis of the dataset.

Chapter 4

Analysis

4.1 Introduction

This chapter describes the evaluation and analysis phase of this research undertaking. The following sections analyse the passive DNS data obtained from FarSight Security using the model proposed in the previous Research Methodology chapter.

4.1.1 Tools and Resources Used

Below is a list of resources and tools, among others, that were used during this analysis:

- Ubuntu 18.04: This machine was used for the analysis and querying of the FarSight security database so as to extract the Passive DNS dataset.
- Windows Server 2012 64GB RAM: This server was used by the Tableau application for analysis of the Passive DNS data.
- Farsight DNSDB App for Splunk: This plugin for Splunk was used to validate some of the FarSight DNSDB queries as this application provides a user-friendly graphic interface.
- DNSDB API (dnsdb_query.py): This application programming interface allows authorised users to query and extract data from the FarSight DNSDB.
- Tableau: The analysis tool was used to filter and helped in the visual graphics.

4.1.2 Datasets

The collection of DNS data can be categorised along the following two dimensions: (1) where, and (2) how the data is collected.

4.1.2.1 Where and How the Data was Collected

Due to the distributed nature of the DNS infrastructure, multiple locations can be considered to collect information about DNS queries and replies. Among all servers involved, the resolver is unique as it is the only location which has access to queries coming directly from client machines.The FarSight Passive DNS dataset uses recursive name servers to log queries and responses from different authoritative name servers. The logged data is then copied to a central database which is used in this research. The near real-time capture of the majority of Passive DNS data is done above the recursive name server. This means that the Passive DNS database is comprised mainly of queries and answers from online authoritative name servers. The logged data is deduplicated, compressed, time-stamped, and copied onto a central database where it can be analysed and archived.

Obtaining information about existing associations between IPs and domain names at a given point of time can be done in two ways (passive or active). This research project utilised historical DNS information.

4.1.3 Sources of Data Enrichment

DNS data represents an important source of intelligence that has been successfully used by many approaches to discover and predict malicious activities. However, to provide deeper insights about malicious activities and to enhance the accuracy and coverage, the detection approach presented in this research undertaking utilised external sources of data to enrich DNS information. For example, mapping the IP address to a hosting country enables some approaches to use the trustworthiness of the country as a feature in classifying the maliciousness of domains/IPs.

4.1.3.1 Enrichment Information Types

1. **Geolocation:** The geolocations of IPs and domains are commonly used to understand the diversity of the origins of the DNS queries as well as of machines hosting

the domains. The most common source of IP geolocation information observed in the literature is the Maxmind Database[1].

2. **Autonomous System Number (ASN)**: An Autonomous System (AS) is a connected group of one or more IP prefixes, blocks of class A, B or C networks, under the control of a network operator, or multiple operators sharing a routing policy. Each AS is assigned a unique Autonomous System Number (ASN). An AS is not restricted to a single country and may contain geographically dispersed locations. This source of information enables us to understand the distribution and utilisation of malicious sites. For example, legitimate domains (except those using Content Delivery Networks) are usually hosted on one or few ASNs as opposed to malicious domains which hop from one ASN to another to evade detection. An ASN is a valuable source of information which helps to distinguish legitimate domains from malicious domains.

3. **Registration Records:** Even though domain registration records often are not verified by authorities, the information located there sometimes can be used as supportive evidence to link malicious domains controlled by the same adversary. Further, temporal information of registration records (e.g., their creation/expiration time) is critical to identify domains registered automatically in bulk to be used later for malicious activities. In fact, some previous works rely purely on registration records to identify malicious domains (Kuyama et al., 2016). The registration records information is usually obtained from servers which provide access to it through the WHOIS protocol (Lauinger et al., 2016).

4. **IP/domain blacklists/whitelists:** Domains are also often checked against well-known IP/domain blacklists.

In summary, the list below highlights some of the key datasets that were used for this analysis:

- DNSDB Passive DNS records accessed via the API supplied by FarSight Security (dnsdb_query.py),
- Known malware using DNS TXT Records,
- List of malicious domains obtained from various sources.

[1]https://www.maxmind.com/en/home

4.1.4 Passive DNS Database

The first step in the analysis was to extract relevant DNS entries from the DNSDB database. According to Farsight (2018), the DNSDB database currently has over 100 billion unique DNS records with over 200,000 new raw observations/second, totalling over 2TB of DNS data, collected daily. As a result of huge volumes of DNS data involved, it is important to filter and deal with only relevant entries. As a result, only DNS entries with TXT records are considered for this analysis. The researcher was granted access to an online/live dataset, and was allowed to perform up to 1000 queries per day.

Due to the inefficiency of using DNS as a payload distribution medium, the number of DNS entries with TXT records are expected to be lower than the number of other Resource Record entries. Nadler et al. (2017) noted that only a few malware families use DNS to spread their malicious code. However, new variants of malware families that use DNS to spread their payload have been discovered recently and this research undertaking analyses some of these new malware families and their mode of operation. This hopefully sheds light on the methods being used to avoid detection.

This research undertaking extracted DNS entries covering a period of six (6) months: from 01 January 2017 to 30 June 2017. The rationale behind choosing this time period is that the period is long enough to have include multiple malware families that exhibit the features that are relevant to this study and gives enough coverage to detect instances where DNS TXT records have been used to distribute payload. If there were newly discovered malware families that utilise these records, they were then compared with historic malware families that are known.

4.2 Extracting the Passive DNS Dataset

A total of 1542 top level domains were considered for this analysis. The list of the TLDs was obtained from the Internet Assigned Numbers Authority (IANA) whose role is to coordinate global IP addressing, the DNS root and other IP resources. To simplify the analysis, all Internationalised Domain Names (IDNs) were excluded although these are commonly used for phishing purposes as these domains can easily deceive users to visit a seemingly non-malicious website. An Internationalised Domain Name is an Internet domain name that contains at least one label that is displayed in software applications, in whole or in part, in a language-specific script or alphabet (Liu et al., 2018).

As a result, only DNS TXT records for 1391 top level domains were considered. Most of these top level domains did not show any activity during the period under consideration. TXT records for each top level domain were extracted in JavaScript Object Notation (JSON) format, and multiple JSON files for different top level domains were joined for analysis purposes. Due to limitations when extracting bulk records for the `.com` top level domain and the number of domains under the .com top-level domain, they were not considered for this analysis. The `.arpa` domain was also not considered as it is practically no longer used for any purposes that may be applicable to this research. Extracting all TXT entries using the DNSDB Python API proved to be challenging due to the limited nature of commands and options available through the API. The API supplied by FarSight Security and those supplied by third-parties have a few commands and these commands are limited to extracting only TXT records belonging to one top-level domain at a time. This had to be repeated for each top-level domain.

4.3 Pre-Analysis Filtering

Commercial tools became a necessity to facilitate the extraction of DNS entries with TXT records. Farsight DNSDB App for Splunk was used in conjunction with the trial version of Splunk Enterprise suite in some instances because of the user friendliness of the application. Figure 4.1 shows a screenshot of some of the TXT DNS entries for a domain `oakwoodsys.com`.

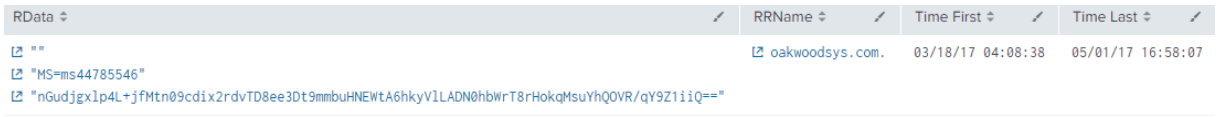

Figure 4.1: Details of TXT Records for Domain oakwoodsys.com using Splunk

The same can also be achieved via the query shown in Figure 4.2. Using Splunk to query the passive DNS dataset is much easier because of the user-friendly graphical user interface. However, during the analysis the researcher used the Python queries and commands supplied by FarSight Security which are well documented.

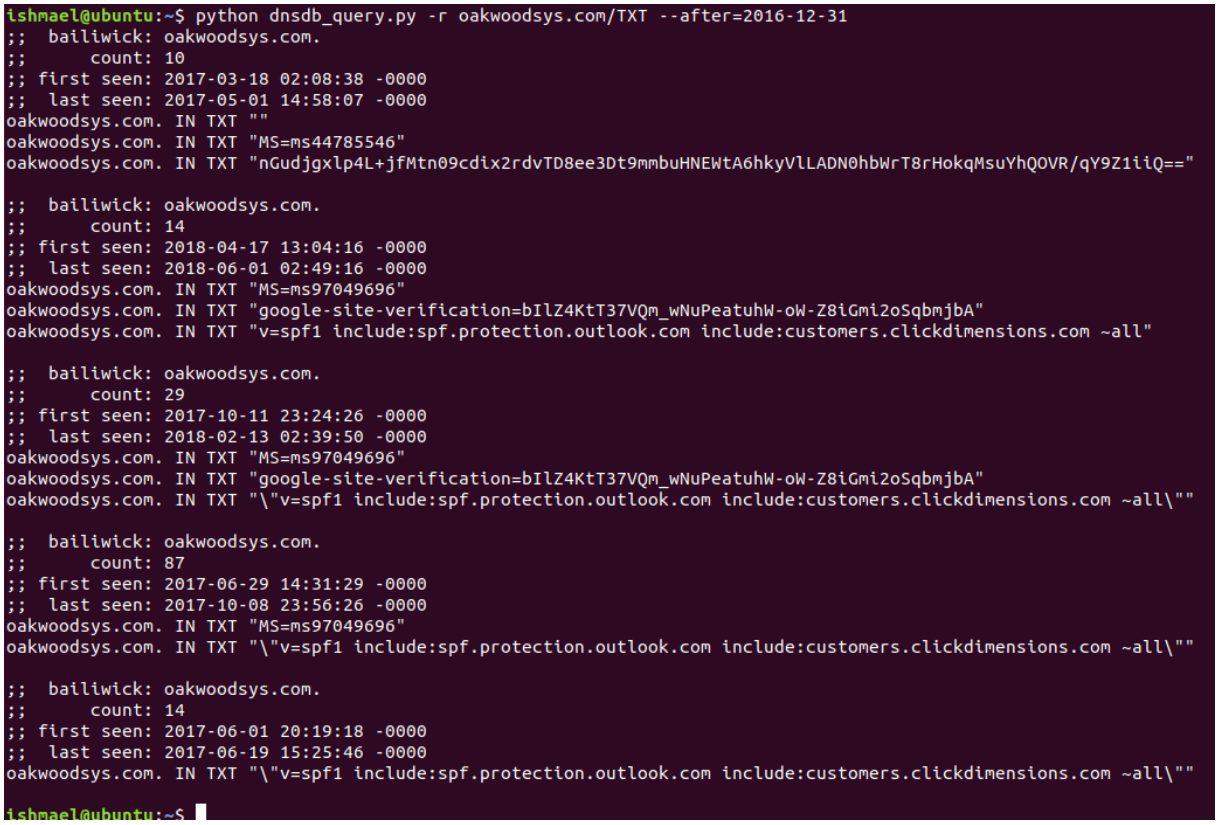

Figure 4.2: Python Query on TXT Records

When analysing TXT records, it is essential to consider the normal data that may be contained in a TXT record. TXT records hold free-form text of any type. A fully qualified domain name may have many TXT records. The most common legitimate uses for TXT records are the Sender Policy Framework (SPF), Domain Keys (DK), and Domain Keys Identified E-mail (DKIM). SPF records are used to inform other parties about IP addresses that are authorised to send emails from that domain (Scheffler et al., 2018). This information is normally published via the DNS TXT record and increases chances of having emails from a particular domain being whitelisted. According to Scott (2014), it is best-practise to publish SPF records via the DNS as a TXT and SPF record. TXT records historically have also been used to contain human readable information about a server, network, data centre, and other accounting information. In this data analysis, all TXT records that contain any of the above forms of information are not considered, though care should be taken as malicious information may be obfuscated in some text.

This filtering process shown in Figure 4.3 also includes DNS records that contain empty TXT records.

```
ishmael@ubuntu:~$ python dnsdb_query.py -r 967-28275.id-27107.down.sd.dyn.outflux.net/TXT --after=2016-12-31
;;  bailiwick: dyn.outflux.net.
;;       count: 2
;; first seen: 2017-05-19 16:08:09 -0000
;;  last seen: 2017-05-19 16:08:09 -0000
967-28275.id-27107.down.sd.dyn.outflux.net. IN TXT ""
```

Figure 4.3: DNS Entry with an Empty TXT Record

Once the above filtering criteria had been implemented, the dataset became smaller and further analysis was performed based on the model proposed in the Research Methodology Chapter.

4.4 Overview of the Approach and the Analysis

The proposed approach is composed of a module for query and response analysis and, another module for detecting and analysing the malicious payload distribution channels. The DNS query and response messages were sent to a module that analyses the patterns to determine the pattern for each domain. The DNS query and response messages were then sent to a module responsible for detecting payload distribution channels. This module extracted all the DNS Resource Record (RR) activities for particular domains from the passive DNS data. The module is responsible for determining the intensity of the payload distribution using the activities of DNS zones. Further to that, there is a need to do some filtering so that legitimate domains that may resemble malicious payload distribution channels may be filtered out.

To validate the system proposed above, a malware dataset was used for correlation, and the passive DNS data was investigated for any differences that existed between regular legitimate domains and the malicious payload distribution channels. For regular domains, Alexa Top 1000 sites that are used for various legitimate services, it was expected that the domains from the Alexa Top 1000 sites would receive DNS queries for various resource records since these domains utilise the DNS to enable access to various services. For malicious domains, the expectation is that they will receive huge volumes of DNS queries for the TXT records.

Another record type that was examined is the CNAME record type. Access to this record was investigated as this record type is responsible for redirecting to malicious domains and cybercriminals use these record types to maintain their malicious networks that distribute payloads using the DNS (Kara et al., 2014).

4.5 Analysis

This section explains the analysis, the outcomes of the analysis and the results of the experimentation implemented. The aim is to show that it is possible to detect payload distribution channels that are used for multiple purposes using passive DNS traffic. The experimentation involved using previously analysed malware that are used to distribute malicious payload using the TXT records of the DNS protocol. In some instances, the payload was obfuscated and could not be easily detected except through the use of decryption.

In summary, the domains that were accessed for TXT records were identified and processed further to determine the query and response pattern. The query and response pattern analysis reveal the nature of each domain and the type of activities it is used for. In the final step, the intensity analysis is performed to determine how the zone is used in payload distribution.

4.5.1 DNS Query and Response Pattern Analysis

During this phase, the probability of each query and response to be involved in the distribution of payload was analysed through comparison of the average distinct count of TXT messages to those that are identified to have been involved in malicious activities. Using previous work by Kara et al. (2014), the query and response pattern analysis allows the researcher to classify the pattern as either Many-to-Many, Single-to-Single or Single-to-Many. These patterns generally refer to the ratio of the number of command and control server to the number of target machines. As an example, Many-to-Many would refer to many command control servers targeting multiple victim machines, and Single-to-Single refers to a single command and control server targeting a single machine.

Many-to-Many patterns have the potential of distributing huge volumes of data as these patterns have many targets. Many-to-Many patterns are, however, more likely to be easily detected due to volumes of data being exchanged and the rate at which it is exchanged. This DNS query and response pattern is most likely to alert and be detected by security measures in place.

The other class is the Single-to-Single pattern which is targeted to specific systems while maintaining low footprint and visibility.

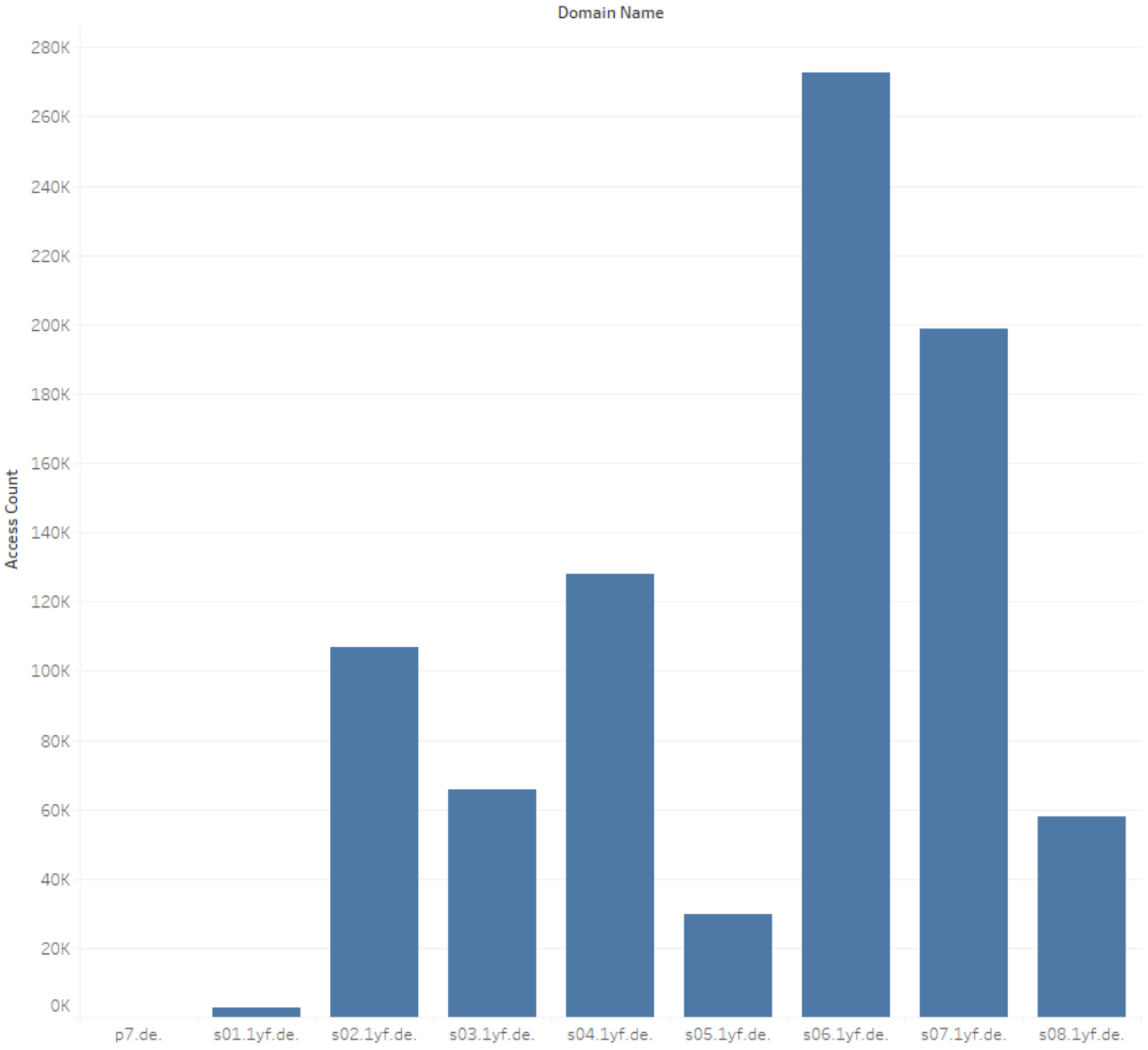

Figure 4.4: Access Count of TXT Records for the .de ccTLD

Single-to-Single patterns allow targets to send just one single query in order to receive the corresponding attack payload. This enables these patterns to remain undetected because such DNS queries can blend with normal DNS traffic. According to Nadler et al. (2017), these are the most resilient payload distribution channels that are difficult to detect.

From the individual top level domain dataset, it was observed that the dataset for the `.de` domain (the country code top-level domain (ccTLD) for the Federal Republic of Germany) was the biggest in size and therefore became an interesting case that needed to be analysed. Figure 4.4 shows the distribution of top access counts for domains in the `.de` ccTLD. The Figure shows only domains whose access count for TXT queries is more than 10. Nine domains accounted for the bulk of queries.

Analysis of this sample shows some domains that have an abnormally high count of queries. The average number of counts per domain in the `.de` ccTLD was relatively low (an average count of approximately 2 per domain). However, the `1yf.de` domains show excessive access count of more than 2000 for each domain.

A further analysis on the domain `1yf.de` reveals that this could be DNS tunnelling traffic. Using the

```
$ dig +trace s01.1yf.de
```

command it can be observed that `s01.1yf.de` relies upon the name servers shown in Figure 4.5.

```
1yf.de.                 86400   IN      NS      dns4.resolution.de.
1yf.de.                 86400   IN      NS      dns5.resolution.de.
1yf.de.                 86400   IN      NS      dns3.resolution.de.
1yf.de.                 86400   IN      NS      dns2.resolution.de.
```

Figure 4.5: Output of the $ dig +trace s01.1yf.de Command

Checking the DNSDB to find out other domains that could be using the same name servers reveals that there are several other domains that use the same name servers. The domains that use the same name servers are shown in the output in Figure 4.6.

```
ishmael@ubuntu:~$ python  dnsdb_query.py -n dns4.resolution.de | awk '{print $1}' | grep -v " SOA " | sort -u
1yf.de.
2yf.de.
45q.in.
49o.de.
53r.de.
63z.de.
8u6.de.
pptp.your-freedom.de.
pptp.your-freedom.net.
q1x.be.
q1y.be.
www.your-freedom.de.
www.your-freedom.net.
your-freedom.de.
your-freedom.net.
```

Figure 4.6: Domains that use dns.resolution.de as a Name Server

The records in Figure 4.6 all appear to be related to the `www.your-freedom.net` domain. The `www.your-freedom.net` domain offers VPN tunneling, firewall and proxy bypassing, anonymisation and anti-censorship solutions (Your-Freedom, 2017). In this case, the record types NULL and TXT appear to be obfuscated by encryption.

Compared with other DNS traffic patterns, Many-to-Many patterns generate the most traffic and the voluminous exchange of traffic often reveals the name server used for payload distribution. According to Kara et al. (2014), cybercriminals often use such name servers just to initiate the payload distribution. Initial queries are made to rogue DNS name servers. Prior research on the FeederBot malware highlighted that legitimate domains are used to initiate queries but close investigation made the researcher to conclude that this could be the first stage in the payload distribution as the data is sent in some sub-domain labels. Responses to such queries come with the domain name to be used for payload distribution and the address of the open resolver that will be used for payload. According to Kührer et al. (2015), open resolvers are often abused for hiding name servers that are used for malicious purposes.

4.5.2 Detecting the Payload Distribution Channels

Once the pattern of DNS queries and responses had been observed, they were then inspected in the module that analyses DNS zones. This was achieved through the use of access counts of TXT resource records for each domain. This information was gathered from the passive DNS dataset. From the dataset, it was observed that there were over 10 million domains with TXT resource record activities. The Access Count ranged from 1 to 272 737 (for the domain `s06.1yf.de`: possibly used for DNS tunneling)

4.5.2.1 Comparison With Other Datasets

Alexa Top 1000 Domains: To ensure that the model can be validated, the difference in resource record activity of regular Alexa Top 1000 domains[2] and the payload distribution channels was investigated to account for malicious behaviour. This process involved retrieving the access counts of Resource Records for both known malicious domains and the regular Alexa Top 1000 domains as these are, in the opinion of the researcher, a reasonable measure that can be used to understand the activity of Resource Records of any domain under consideration.

As expected, the domains for the Alexa Top 1000 received queries for various resource records and this could be attributed to the fact that these domains are utilised for different

[2]Of the Alexa Top 1000 domains extracted as at 12/08/2018, only 708 had DNS traffic during the period under investigation. The remaining 292 domains did not have any activity and the extracts were empty: some domains were not even existent during the period under consideration.Unfortunately the researcher could not obtain a historical Alexa Top 1000 domains list for the period under investigation

services (Figure 4.7). The malware domains also received queries for various resource records but in some instances there were significantly higher counts for certain resource records as shown in Figure 4.8. For these malware domains, the CNAME records were also investigated, as this record type can be used to redirect between domains. According to Kara et al. (2014), cybercriminals and botmasters use this resource record type to maintain their malicious payload distribution channel network.

SOA	9,541,378,576
A	285,639,889
NS	24,970,076
AAAA	19,029,319
TXT	1,634,698
MX	187,206
SPF	16,014
CNAME	1,290
HINFO	344
TYPE0	6
CAA	4
TYPE56436	2

Figure 4.7: Alexa Top 1000 Resource Record Distribution

NS	3,841,395
A	2,313,640
CNAME	995,715
SOA	393,299
MX	93,550
AAAA	28,094
RP	9,527
TXT	7,078
HINFO	4

Figure 4.8: Malware Domains Resource Record Distribution

Top 20 "Shady" TLDs: According to Larsen (2018), "shadiness" looks at the ratio of malicious second-level domains to the total number of domains registered. Other alternative terms used are "evil index", "badness", among others. The formula for calculating "shadiness", "evil index", "badness" of a top-level domain is:

$$Shadiness = \frac{MalwareDomains}{TotalNumberOfDomains}$$

An analysis of the TXT count for the Top 20 “Shady” TLDs was not conclusive. This may be because these malicious domains focus mainly on phishing and spams and they are not used for intensive payload distribution. The resource record count for the Top 20 ”Shady” TLDs is shown in Figure 4.9.

There were other lists that were used to compare resource activities but these all generally point to the same direction: known malware top-level domains generally do not have a higher TXT resource record access count.

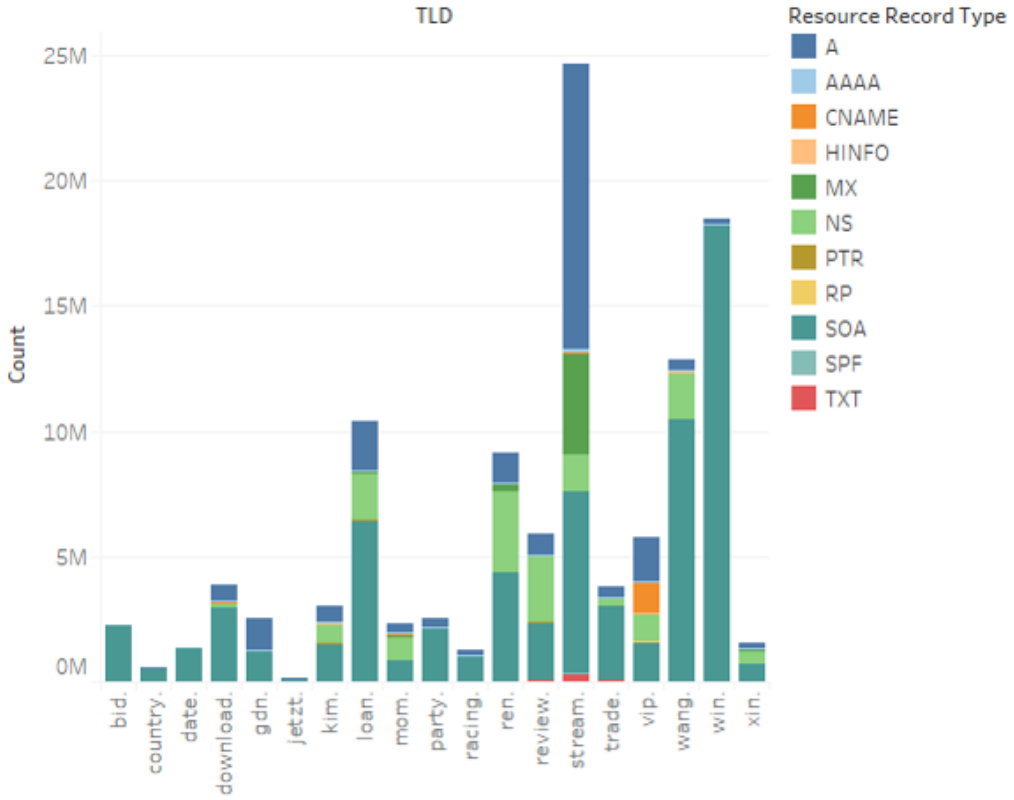

Figure 4.9: Resource Record Count in the Top 20 Shady TLD domains

4.6 Malware Families Under Consideration

The list below shows some of the malware families under consideration including the year that they were discovered. The list is based on previous research and includes the well-known malware families that use DNS resource records to conduct their activities.

- Morto discovered in 2011 (Mullaney, 2011).

- FeederBot discovered in 2011 (Dietrich et al., 2011).
- PlugX discovered in 2013 (Valisenko, 2013).
- FrameworkPOS discovered in 2014 (Cian et al., 2016).
- Wekby discovered in 2015 (Grunzweig et al., 2015).
- BernhardPOS discovered in 2015 (Cian et al., 2016).
- JAKU discovered in 2015 (Settle et al., 2015).
- MULTIGRAIN discovered in 2016 (Cian et al., 2016).
- DNSMessenger discovered in 2017 (Brumaghin and Grady, 2017).

The analysis of passive DNS data focused on the behaviour exhibited by these malware families as they are perfect candidates for transmission of malicious payload via DNS TXT records. Malware samples were obtained for some of these families and for almost all the researcher used previously validated malware analysis to model the DNS query and response behaviour of these samples. In the two cases where the analysis was performed (following what others had done), the results showed that the malware samples had hard-coded command and control servers that were to be contacted. The online analysis environment (`https://www.hybrid-analysis.com`) is sandboxed, and as such there was no harm done (most domains are dead and hijacked ones are now clean).

After a careful consideration of the malware families that use DNS for payload distribution, only two families were considered for this research purpose:

- Wekby discovered in 2015 (Grunzweig et al., 2015).
- DNSMessenger discovered in 2017 (Brumaghin and Grady, 2017).

The researcher focused on these two malware families (Wekby and DNSMessenger) because they were recent and unlike other malware families, they used not just the DNS to distribute payload, but they specifically used TXT records for their malicious activities. The rest of the families either exhibit the “Many-to-Many” pattern and would be easy to detect and be addressed, or there have not been any known/discovered variants that would warrant further investigation. Some malware families like Morto and FeederBot

have substantial academic research on them. DNSMessenger was discovered during the period under consideration.

The analysis of the two malware families revealed that they used certain "phrases" during the querying and responding to DNS requests. Due to the recurring pattern of having hard-coded command and control servers in all the malware families that were considered, the researcher decided to perform a manual analysis and manual detection process. The decision to do so was influenced by the resources and the level of automation that would have been required to achieve what can be achieved via a manual process. Automating the detection process would have increased the scope and complexity of the research considerably

4.6.1 Wekby

Wekby attackers used DNS tunneling which takes advantage of the TXT transport layer within the DNS protocol used by top and second level domain name system servers (Grunzweig et al., 2015). Wekby attackers that have already gained a foothold on targeted systems could use DNS Tunneling techniques to send commands and data. Though slow, this method is well suited for long-term APT campaigns. For example, in the case of Pisloader, attackers would use their own DNS server that they controlled to send and receive command and control commands to infected computers (Chirgwin, 2016). Embedded in the DNS TXT layer of the call and responses between infected client and Wekby's DNS server would be a mix of five instructions/commands described below. To obfuscate those commands, Wekby attackers use base32 encoding on the DNS TXT layer making it appear that the DNS TXT was simply garbage strings of DNS metadata. The following commands, and their descriptions are supported by the malware:

- `sifo`: Collect victim system information
- `drive`: List drives on victim machine
- `list`: List file information for provided directory
- `upload`: Upload a file to the victim machine
- `open`: Spawn a command shell

In addition, according to Grunzweig et al. (2015), the command and control server periodically sends a random 4-byte uppercase string that is used as the payload. The string is appended as part of the domain name. As an example, using the hard-coded command and control domain name `ns1.logitech-usa.com` in the malware samples considered, the domain name in the response would contain information or command in the fourth-level top domain (4LTD) label.

Manual searches for "commands" sent by Wekby did not yield any meaningful results, and therefore using the manual method of searching for phrases/commands resulted in a detection rate of zero. Automated checking for these commands could have helped but this required the ability to decode some domain name labels as these could have been obfuscated. The lack of detection was also checked against tools that track various malware families to ensure accuracy and validity of the findings and during the period under consideration, there was no activity from the Wekby malware family.

4.6.2 DNSMessenger

According to an investigation conducted by Brumaghin and Grady (2017), DNSMessenger uses malicious macros in Word documents to infect victims. The VBA script in the macro unpacks a self-contained PowerShell script and executes it. The script contains code to ensure persistence on the infected host by modifying registry keys and verifying PowerShell versions. DNSMessenger then sends DNS queries to one of the domains included in its source code. The queries retrieve the domain's DNS TXT record, small snippets of text that domain owners add to DNS entries containing base64-encoded PowerShell commands. This loads more DNSMessenger components in the victim's RAM without leaving any trace of malware code on the disk.

According to Brumaghin and Grady (2017), this memory-based code allows the attacker to interact with the victim's computer by relaying shell commands from the attacker. It can use other DNS queries to get the commands it needs from another list of domains. The attackers leave commands inside the TXT records of their domains, the trojan queries for it, gets the command, executes it via the Windows Command Line Processor, and sends the output back as another DNS query. In early March 2017, the domains used and registered by the trojan are all down, preventing researchers from identifying the commands the attackers relayed to the victims. Unless victims monitor their DNS traffic, the infection will not be discovered as DNSMessenger uses DNS queries to hide its activity.

The script uses specific subdomains which are combined with the domains and used for the initial DNS TXT record queries performed by the malware. The malware uses the contents of the TXT record in the response to these queries to determine what action to take next. For instance, the first subdomain is `www` and a query response with a TXT record containing `www` will instruct the script to proceed. Other actions that may be taken are `idle` and `stop`.

4.6.2.1 Analysis of the DNSMessenger Malware

The researcher obtained an earlier variant of the malware and performed a sandboxed malware analysis of DNSMessenger using the steps and methods used by the organisations and researchers who discovered the malware (Brumaghin and Grady, 2017; Brumaghin et al., 2017; Yates, 2017). The analysis by these entities was thorough and detailed. The researcher did not conduct a comprehensive analysis as this was out of scope, and the analysis was performed only to ensure that the DNS details about the malware were correct.

A search for `www` in TXT records revealed a total of six domains as shown in Figure 4.10. An investigation of the first four domains in Figure 4.10 did not point to anything interesting/amiss.

Rdata	Domain Name	Count
"www"	capshire.com.au.	29
	offitalia.it.	6
	tekserve.me.	2
	testdiana.guru.	2
	www.cihr.site.	1
	www.ckwl.pw.	3

Figure 4.10: Search for the "www" in TXT records

The other two domains (`www.cihr.site` and `www.ckwl.pw`) were part of the domains that were hardcoded in the source code of the malware. An analysis of the passive DNS data did not reveal any interesting activity associated with these two domains. The hard-coded domains appeared for less than 72 hours in the global-time series. This is a relatively short time compared to domains used for legitimate purposes, which may be in existence for many years.

The other commands `mail` and `stop` revealed what could be payload distribution but further investigation was not conclusive as shown in Figures 4.11 and 4.12.

Rdata	Domain Name	Count
"stop using spf"	clunk.nl.	2
"STOP443J"	trpmvx46sjt2fmz5ovmoymq.ldufh2a.d.ksx.la.	2
"stop"	_dmarc.svptest.ru.	2
	_adsp._domainkey.svptest.ru.	2
"stoplinux.org.ru"	mail.stoplinux.org.ru.	3
"STOPIH4J7TMT4WVWBL..	58aee.4.1008.lvo6hakhunwwseqkyxlvuwpxzwzh4uydvpcwqy..	2

Figure 4.11: Search for "stop" in TXT Records

Rdata	Domain Name	Count
"mail"	bisnet.com.mx.	38
	brandt-lysgaard.dk.	2
	mail.numearologgy.site.	466

Figure 4.12: Search for "mail" in TXT records

A manual search for the `idle` command yielded interesting results, which are discussed in Section 4.7.

Rdata	Domain Name	Count
"1rQwQDbu4Y5P3RZzIOqdFeQbFdBBdWuZzjRMKI7CA..	api.2a757723c72.18732.ns.newsfeeds-microsoft.press.	2
"3MHEETDA4AQ5JV4YNAKGLPKWNVGLOKGWKJNTZZ..	5936b.1.296.w5uo2xagfxgvzmnodpyuwfblopyinxnkgngnqmvm4h3aly3r..	2
"4bqc/R6fnxp2q/gOnm94keO2fRcIQRKtF1oXHWAIPxp..	api.2351747e2c.90306.bbs.broadcast-microsoft.tech.	2
"4LGM7ALETSLKNFIZEAIKC5VYFAJEPVTZGSRW5ATM..	60e29.1.280.cd6w7lopw7mt335ugec35ud4jbu4m7zbvyf2dsq2zoxjdf3n..	2
"4UMYHORQ324ZI5QSF7VPWWMDHIDLET2PXGXOTM..	eb484.1.296.tyhvcxbempag4nu6bpli4bdqk4xctkdkechvhyrtd73r3ychckf..	2
"5DPYBZ5ETGDRC65WE4C2MHKIGJQ6Q4XEIZFHTRKP..	22d40.1.296.4l3jeet7cse2svd7ozo3gs57rqya2u3kiud6dusfcrek5jjj2kajp..	2
"6eNsmNuCOCWbxRTI1ID7ni4ez640CUFhZ+/zUcAbGJ..	api.33557723c72.18732.ns.newsfeeds-microsoft.press.	2
"6HKdwAvp1gSDnX3tKaHXXeIshjuU2cWKzf8TmVTUe..	api.6881747e2c.90306.bbs.broadcast-microsoft.tech.	2
"6JF6IVNHICLLEZQQWHM3RIDLEE5TWX7RGFANQE5T..	3a3ae.4.1008.5bj6hyj2btuciwagjhwexeha24dh5zdbkns4bv6ismrkjgfbz..	2
"6S4YVRNW52HSBUA5ZBJCHXDY2OGVOFF7IDLEDC6..	4d401.1.296.xkoee6zgmgqf4mf6qw5y5a46v2g4t3yuue5havrfng62st3i..	2
"7EBKFRFM5257MJFAON4MIK4ZOXXYBCIDLE6G3DY6..	3a50e.1.288.crzcus6nxjpoworww4ajlqcun7s2ghg6omlmtmcp5bhshdus..	2
"7JU6AUJ6GT67STNRF5757YEVQK5WDXYHOZJDX72..	bf658.4.1008.qhpdzcjxz7euzwehrdnsdiwhu6oe6rcxqhv5lhp2qc4mmjlcj..	1
"7JVLUXCTOO57BQJ44UBLVGKEILMHOBAHLXGLH3G2..	4698d.1.296.v3neqaz5l3dmhedxw3elbflzuuzqzplw7wwwtmrqnnezdv7c..	2
"7MOUHIR3QJYWD7OHIDLE3QDJMP5L55HJTCJ42WV..	4b904.1.296.me7lp6pbgxot3no5qo2bbxwfnyhx6mxut6iifq5go6iaqja7d..	2
"25UQF55YU6TKAPSOBHO7CIPZ7JIHIDLE7AK3K7EPG..	96e30.1.296.lzavmxc2a4o274z2akr5665r33ubchuuixfgvibf2mlhvnqzn7..	2
"36IWHW62BHWE3HV6YQJHI5IU46F4G7W53KQZ7R..	546de.4.1016.3bnttbu5u6avqq43nrdqnpnx42lo4ghmnsfiawg4pljgygu2..	2
"93mqOAvAJzOBmhQKUZ3DfxXmhH6sGAy/AfzI42xxz..	api.1f4b38072a.13252.email.sharepoint-microsoft.co.	2
"AAAFAGIAAAEAAAAPPHFAMOIPGOFAAAAIDMEBEFN..	zga.stage.7545557.ns1.microx.club.	2
	zga.stage.7585921.jreupdate.javaupdate.co.	4
	zga.stage.15803732.jreupdate.javaupdate.co.	2
"AAKBBANAACBADDMFIJEFPMILEFAIIDKFOEPLPPPPA..	yba.stage.318801.email.sharepoint-microsoft.co.	6
	yba.stage.519047.email.sharepoint-microsoft.co.	8
	yba.stage.1065646.email.sharepoint-microsoft.co.	7
	yba.stage.1115132.jreupdate.javaupdate.co.	5
	yba.stage.1142749.jreupdate.javaupdate.co.	8
	yba.stage.2389207.email.sharepoint-microsoft.co.	5
	yba.stage.2584332.jreupdate.javaupdate.co.	2
	yba.stage.2757521.email.sharepoint-microsoft.co.	3
	yba.stage.3356426.email.sharepoint-microsoft.co.	2
	yba.stage.3704754.jreupdate.javaupdate.co.	3

Figure 4.13: Search for "idle" in TXT Records

Three domains were observed to contain the `idle` command/text:

1. `microx.club`,
2. `sharepoint-microsoft.co`,
3. `jreupdate.javaupdate.com`.

A DNS analysis of the `microx.club` domain did not reveal anything useful. The nameservers associated with the `microx.club` domain are also used by various other seemingly legitimate domains which are active. The `microx.club` domain and all these other domains use the `ultradns.com`, `ultradns.biz`, `ultradns.net`, `ultradns.uk`, `ultradns.org`, `ultradns.info` nameservers, among others.

According to Neustar (2017), UltraDNS is a cloud-based authoritative DNS service that securely delivers fast and accurate query responses to websites and other vital online

assets. It is used for faster, safer and accurate query resolution. No further investigation was carried out on the `microx.club` domain. The other two domains exhibited behaviour that warranted further investigation. The analysis and investigation of the two other domain is described in Section 4.7.

Based on the analysis by the researchers who discovered the DNSMessenger malware and the analysis by the author:

1. The activity on the hard-coded domains was short-lived (roughly two days) meaning that by then there might not have been enough time for the malware to spread. It can be assumed that the idea was to spread the infected Word document containing a macro to as many possible targets as possible using phishing attacks before the malware could start communicating with the command and control servers.
2. The malware appears to have been targeted, which is typical in Advanced Persistent Threats (APTs). Pointers are in the code of the malware which made it fileless malware operating in memory.
3. There is possible use of open resolvers as may have been shown by low activity on passive DNS data. There is also possible use of rogue DNS resolvers. Later variants of DNSMessenger changed the way the malware communicates using the TXT records (Brumaghin et al., 2017). The later variants do not have hardcoded domain names, and are dynamic in nature.

4.7 Botnet

Section 4.6.2 introduced the analysis of DNSMessenger and TXT records which have the text `idle` in them. The analysis revealed a possible botnet associated with the domain `sharepoint-microsoft.co`. The domain exhibited features that are usually associated with DGAs and had specific patterns that made it look suspicious and warrant further investigation. Further analysis was conducted as the behaviour of the domains showed that it was a possible command and control for a botnet. This was because the third-level and upper level domains had incremental names that appeared to be generated by algorithms. Further investigation of the `sharepoint-microsoft.co` domain revealed that it is used for malware delivery, command and control, among others.

Noteworthy observations about the domains `sharepoint-microsoft.co` and `jreupdate.javaupdate.com` are highlighted below:

1. The two domains impersonate major Internet, software companies and services Microsoft (`sharepoint-microsoft.co`), Oracle (`jreupdate.javaupdate.com`), among others.

2. The attacker used WHOISGUARD[3] for protecting whois information for the domains. WHOISGUARD substitutes a domain owner's private information with its own information so that this information cannot be readily accessed by spammers. This is one of the ways that are used by cybercriminals to avoid being identified.

   The above domains had long sub-domains like those used by Content Delivery Networks as highlighted in Figure 4.14.

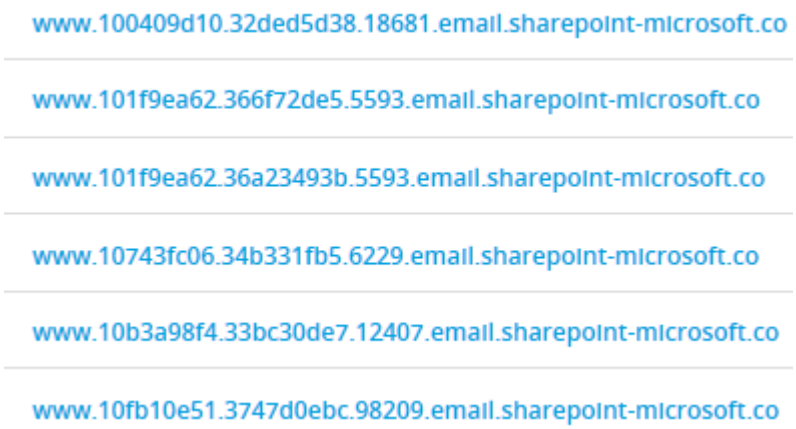

Figure 4.14: Content Delivery Network

The `sharepoint-microsoft.co` also had many multiple sub-domains associated with it as shown in Figure 4.15.

[3]http://www.whoisguard.com/

SUBDOMAINS

Show : 25 1-25 of 1,503 Sort : Hostname Ascending

Hostname
10650.email.sharepoint-microsoft.co
10862.email.sharepoint-microsoft.co
10869.email.sharepoint-microsoft.co
11014.email.sharepoint-microsoft.co
11016.email.sharepoint-microsoft.co
11078.email.sharepoint-microsoft.co

Figure 4.15: WHOIS Information

It appears that the botnet points malicious domains to IP addresses not in their control. For example, the `sharepoint-microsoft.co` domain pointed to a non-malicious IP owned by Microsoft (`sharepoint-microsoft.co` resolved to the IP address `104.43.195.251`).

Resolve	Location	Network	ASN	First	Last
52.54.12.82	US	52.54.0.0/15	14618	2018-02-05	2018-02-05
54.172.134.14	US	54.172.0.0/15	14618	2018-02-05	2018-02-05
35.153.23.210	US	Unknown		2018-02-05	2018-02-05
104.43.195.251	US	104.40.0.0/13	8075	2017-02-12	2017-08-01

Figure 4.16: IP Address Resolution

`104.43.195.251 IP address information`

```
Country         US
Autonomous system 8075 (Microsoft Corporation)
```

Below are some of the domains associated with the IP address `104.43.195.251`:

- `windows.com`,
- `dynamics.com`,
- `dynamics.com`,
- `www.msdn.com`,
- `www.windows.com`,
- `xbox.com`,
- `microsoft.ca`,
- `www.microsoft.ch`.

A DNS analysis of the domain `javaupdate.co` revealed a pattern that was similar to that of `sharepoint-microsoft.co`. The domain `javaupdate.co` resolved to a non-malicious IPs owned by Oracle Corporation.

Show : 25 ◂ 1-1 of 1 ▸ Sort : Last Seen Descending

Resolve	Location	Network	ASN	First	Last
137.254.120.50	US	137.254.0.0/17	792	2017-01-19	2017-08-24

Figure 4.17: IP Address Resolution

This pattern was instrumental in directing our research focus, and detecting that the `sharepoint-microsoft.co` and `javaupdate.co` domains were malicious domains. This information pointed to other possible malicious domains, but the leads were not followed due to the fact that this malware family has already been analysed in detail before in a study conducted by ClearSky (2017), which is described below.

A review of the behavior and a search for previous studies led to a report by a company called ClearSky which documented an APT group called CopyKittens. ClearSky and

Trend Micro published a report in July 2017 exposing a vast espionage apparatus spanning the entire time the espionage group has been active (ClearSky, 2017). The report identified CopyKittens as a cyberespionage group that has been operating since at least 2013. The report also covers the period under investigation in this research, although not all domains mentioned in the report could be detected using the manual methods employed in this research.

4.7.1 Targets

According to ClearSky (2017), CopyKittens is an active cyber espionage actor whose primary focus appears to be foreign espionage on strategic targets. Targeted organisations include government institutions, academic institutions, defence companies, municipal authorities, sub-contractors of various ministries of defence, and large IT companies (ClearSky, 2017). Online news outlets and general websites were also breached and weaponised as a vehicle for watering-hole attacks.

The group uses Matryoshka v1, a self-developed Remote Access Trojan (ClearSky, 2017). The group also uses Cobalt Strike. According to CobaltStrike (2018), their product is a publicly available commercial software for "Adversary Simulations and Red Team Operations". Adversary Simulations and Red Team Operations are security assessments that replicate the tactics and techniques of an advanced adversary in a network. While penetration tests focus on unpatched vulnerabilities and misconfigurations, these assessments benefit security operations and incident response teams as they replicate the techniques used by advanced adversaries in a network. Cobalt Strike gives users a post-exploitation agent and covert channels to emulate a quiet, long-term embedded actor in the network. A notable characteristic of CopyKittens is the use of DNS for command and control communication, and for data exfiltration. This feature is available both in Cobalt Strike and in Matryoshka RAT (ClearSky, 2017).

Most of the infrastructure used by the group is in the United States of America and Great Britain (IP address and analysis of information obtained from other sources: geolocation, WHOIS, etc.).

	Resolve	Location	Network	ASN	First	Last
☐	69.172.201.153	US	69.172.201.0/24	19324	2016-04-22	2018-08-29
☐	176.74.176.187	GB	176.74.176.0/22	13768	2016-09-14	2018-02-01

Figure 4.18: Geolocation

4.7.2 Infrastructure Analysis

The malware uses the `sharepoint-microsoft.co` domain for malware delivery, command and control, among others. It was also revealed by ClearSky (2017) that the malware runs a Cobalt Strike stager that communicates with the domain:

`yyy.stage.xxx.email.sharepoint-microsoft.co`

where `xxx` is a stage number (as shown in the TXT DNS records that gave pointers) and the meaning of `yyy` cannot be determined at the moment.

4.7.3 Cobalt Strike

While not malicious in and of itself, Cobalt Strike is often used by cybercrime groups and state-sponsored threat groups, due to its post-exploitation and covert communication capabilities. Malicious communication generated by the tool is easy to detect, because a special header is sent in each HTTP GET transaction. The special header is "X-Malware," i.e. there is a literal indication that "this network communication is malicious." All that defenders need to do to detect infections is to look for this header in network traffic. According to ClearSky, CopyKittens often use CobaltStrike's DNS-based command and control capability.

Other capabilities include PowerShell scripts execution, file downloads, spawning other payloads, and peer-to-peer communication over the Server Message Block (SMB).

4.8 Analysis of the Other Domains

Other domain records with 255 characters (excluding those with DKIM, SPF and other filters used before) were analysed for any suspicious behaviour. Decoding the content of TXT records did not yield anything using readily available encoding/decoding tools.

Most of the DNS records with 255 characters in the TXT field were either associated with the:

1. CopyKitten APT mentioned above, or
2. the `1yf.de` domain associated with `https://your-freedom.net/`, a VPN Tunneling, Firewall and Proxy Bypassing, Anonymization and Anti-Censorship Solution, discussed in previous sections.

The second group of domains that have 255 characters in the TXT field were:

- `dns.njcate.org` - shares the nameserver with the `sharepoint-microsoft.co` domain. There were, however, multiple changes in name servers since the `dns.njcate.org` domain was registered.
- `ksx.la` - nameserver associated with `ksx.la` resolves to multiple domains, most of which are associated with suspicious malware content.
- `bn.tl` - has been used to serve malware in the past.

The rest of the second level domains follow the same pattern as `sharepoint-microsoft.co`, `jreupdate.javaupdate.com`. These domains are:

`tg3.11v.in`, `tg3.09m.in`, `tg16.0x7.in`, `tg58.09j.in`, `tg58.09m.in`.

Taking the first domain `tg3.11v.in` and searching for whois information, reveals the following:

```
Domain Name:11V.IN
Registrant Street1:Noapara.
Registrant City:Kolkata
Registrant State/Province:West Bengal
Registrant Postal Code:700125
Registrant Country:IN
Registrant Phone:+91.9433300300
Registrant Email:papai.sofo@gmail.com}
Admin Phone:+91.9433300300
```

Reverse Whois results for `papai.sofo@gmail.com` shows that there are 61 domains that matched this search query. However, a search for +919433300300 (the phone number that's associated with these domains) reveals that +919433300300 was also used to register 19 other domains. This could be an indicator of what these domains could be used for.

Of the 19 domains registrations using this number, seven have expired and all have a second level label of `bokbok`. Checking the reversewhois tool for that number returns a number of domain names, including (but not limited to):

- `bokbok.biz`
- `bokbok.info`
- `bokbok.co.in`
- `bokbok.live`
- `bokbok.chat`
- `bokbok.guru`
- `hammervpn.com`
- `antidpi.com`
- `tunnelguru.net`
- `smartdnsproxy.pro`
- `dnsproxy.pro`
- `vpnoverssh.com`
- `slowdns.com`
- `myvpncompany.com`
- `myvpn.company`
- `troidvpn.com`
- `tunnelguru.com`
- `web-tunnel.com`

- `bokbok.com`

In spite of the disappointing lack of detail in that whois data, the whois information for the websites above does at least give a new email to potentially follow: `domainstores2@gmail.com`. Based on the above domain names, the full set of domains can be comfortably tagged as likely being DNS tunnel-related too. The above scenario highlights the complexity in hunting for threats and tracing them to the source. Multiple tools, leads and methods were used, and these help give a high level of confidence in the finding/conclusion.

4.9 Conclusion

This chapter discussed the analysis of the passive DNS dataset and various enrichment datasets in order to investigate the payload distribution channels that use the TXT resource record. Specific malware families, namely Wekby and DNSMessenger were considered for this analysis, and these malware families exhibit features that can help in the understanding and detection of similar malware families.

Using manual procedures and searching for specific commands in the TXT records, it was discovered that these payload distribution channels can be detected using a manual process.

While investigating one of the commands, the researcher stumbled upon what appeared to be a botnet that uses TXT records, and further investigation indicated that this was indeed an advanced botnet used by one of the APT groups, CopyKittens.

From the analysis (excluding the `.com` TLD, the `.arpa` TLD, those with SPF, DKIM among others), it can be concluded that for DNS records that have 255 characters in the TXT field:

- They are either involved in malware payload distribution or,
- They are being used for DNS Tunnelling services (which can be abused to distribute malicious payload).

The next chapter will discuss the findings, make recommendations, identify gaps and give a conclusion to the research undertaking.

Chapter 5

Results and Discussion

5.1 Introduction

The main objective of this research was to analyse multi-purpose payload distribution channels using the DNS. The goal was approached by analysing historical DNS data in order to characterise and understand the behaviour and infrastructure used by the criminals who use DNS to distribute malicious payload. This was achieved through the analysis of the DNS activities related to TXT resource records during a six-month period in 2017. This chapter presents the results and the findings of the analysis that was conducted in the previous chapter based on the outlined methodology.

Inspired by the changes in the way the DNS TXT resource records are used to distribute malicious payload and various mechanisms that are used to evade detection, this work presents an analysis of malware samples that are known to primarily use DNS as part of their malicious infrastructure.

5.2 Background

Detection of malicious payload distribution can be challenging considering that there are genuine cases where the DNS is used to distribute legitimate traffic or payload. The findings presented in this section demonstrate that prior work conducted by others aimed at the detection of payload distribution channels is still effective but generally faces the challenge of being passive, never real-time and generally depends on old data sets. Some of

the prior works depended on datasets for a short period (usually two weeks to a month) that were more than two years old. This research, while it did not address all these challenges, analysed domain queries for a predefined (six month) window which was recent at the time when the research concept was conceived.

While previous research has studied payload distribution channels, these previous studies have primarily focused on DNS tunnels and have given less attention to the analysis of malware families that distribute malicious payloads using these DNS tunnels.

This research analysed methods and mechanisms for detecting these DNS tunnels that are used to distribute malicious payloads, and also investigated some payload distribution malware families so as to understand the trends and the shifts in this class of malware. Historical records of DNS were analysed for a six-month period since shorter time windows are inefficient for analysing and investigating malware families with low throughput payload distribution and exfiltration. This was done in comparison to the Top 1000 Alexa domains so as to compare the behaviour of malicious domains to those domains that are mostly used for legitimate purposes.

The manual analysis observed and found the presence of DNS tunnelling activities some of which were used for legitimate purposes such as the Your Freedom project used for bypassing censorship controls in certain regions and countries. While these tunnels provide different capabilities that are useful, criminals also use the same mechanism to conduct their activities and this ability to provide covert channels that can be used for malicious activities threaten the security of many organisations. The threats posed by DNS tunnelling tools and capabilities can be mitigated by payload and DNS traffic analysis. This analysis aids in the detection and finding ways of mitigating the threats posed by payload distribution malware families such as DNSMessenger.

The research was able to manually detect DNS tunnels using signatures that are based on known DNS payload distribution malware families attributes. This was achieved through a combination of DNS query and response analysis (traffic analysis), which is based on the characteristics of aggregate DNS traffic, and through the analysis of the payload contained in the TXT records of the DNS queries and responses. While the above technique was successful in demonstrating the ability to detect and analyse payload distribution mechanisms, there is still room for improvement as the malware families that were investigated have evolved and become dynamic. Later variants of DNSMessenger discovered no longer have hard-coded command and control servers. The new variants have command and control servers that are dynamically generated and performing manual searches can

be time consuming if the format of the domains is not static. To improve the analysis, it is important to incorporate automated analysis as this will minimise human intervention and reduce the time needed to perform an investigation.

5.3 Answering the Research Questions

The aim of the research, as mentioned in Chapter 1, was to widen the research into the use of DNS for malicious payload distribution by filling in the existing research gap through analysing DNS traffic generated by domains used to distribute multiple payloads. In answering the research questions above, the researcher characterised DNS messages associated with malicious networks and investigated various ways that malicious networks use DNS to distribute attack payloads.

The findings are presented below and attempt to answer the research question that is the basis of this research:

5.3.1 Research Problem

Research Question: Can domains used for distributing multiple malicious payloads using covert channels be characterised and modelled based on the data transferred using the DNS?

Answer: From the analysis of the passive DNS data and information associated with the malicious domains, it can be established that the domains that are used for distributing payload can be detected and identified using various features that are synonymous with malicious activities. These were identified in previous works and were applicable to this research. TXT resource records were used in this research to characterise each distribution channel and establish whether a channel was distributing payload based on the DNS activities related to TXT resource records.

The analysis was based on two steps, namely the DNS zone analysis which was followed by the payload distribution analysis. Using the malware samples that were identified to have been active during the period under consideration, namely Wekby and DNSMessenger, it was found that these malware families still use the basic method of sending using simple commands using the TXT resource records. These commands could be sent to initiate other commands and at times to transmit information from one end to the other. A

high proportion of TXT resource records in DNS network traffic was found to be an indicator of the presence of DNS tunnels that are used for distributing payload (malicious or non-malicious).

Based on the analysis conducted, behaviour associated with Advanced Persistent Threats (APTs) was identified. The TXT records for domains associated with the analysed malware families (Wekby and DNSMessenger) indicated relatively low levels of activity and had fewer DNS queries compared to other malicious domains. This may be due to the fact that these malware families appeared to be directed at specific targets, and therefore having many DNS queries would increase the likelihood of detection. The DNS messages associated with domains linked to DNSMessenger indicated that the malware sent queries and responses with only a few characters, unlike the CopyKittens malware that utilised all the 255 characters available in the TXT records.

5.3.2 Research Sub-Questions

1. **How can the abuse of the DNS be quantified and all the involved parties and infrastructure be identified?**

   Answer: The passive DNS dataset provides a starting point for identifying the abuse of the DNS. From the analysis, there were many indicators of anomalies ranging from simple indicators such as many DNS queries and responses during a specific time, followed by a period of relative inactivity, to other indicators such as FQDNs that resemble the pattern used by DGAs. This information coupled with other publicly available information such as WHOIS data, domain registration data, among others, can help identify the infrastructure and possibly some parties that are involved. WHOIS data contains information associated with an IP address, such as the Autonomous System (AS) number, Border Gateway Protocol (BGP) prefix, country code, and registrar information and allocation date. Using this information, the infrastructure involved was identified, although attributing it to specific organisations/individuals is a bit complicated due to the fact that botnet masters tend to hijack other infrastructure in order to conduct their attacks. Care should, however, be taken to attribute the source of these abuses as the origin of a payload distribution could also be compromised and could be used as a mask in order to avoid being identified by law enforcement agencies.

   An analysis of the passive DNS data set data provided clues about potential malicious activity, based on the length of domain names, the period the domain was

active and the count of DNS queries and responses. This information is vital in establishing the foundation for further analysis.

The abuse of the DNS can be quantified through the query and response pattern and this gives an indication of the parties involved. Using supplementary information from secondary sources, the infrastructure used by malicious actors can be identified

2. **Can encrypted payload distribution channels that are used for various purposes be detected and monitored through analysis of DNS traffic?**

   Answer: From the TXT records that were analysed, possible encryption was observed especially for the domains that are associated with legitimate DNS Tunnelling (discussed in Sections 2.4 and 4.5.1). However, using the publicly available decryption and decoding tools the researcher was unable to get any meaningful information from the entries of TXT records. While the content of the TXT records could not be determined, in some of the instances, certain patterns like having 255 characters in a TXT record were enough indicators to warrant further analysis and investigation because it was a deviation for normal usage of the TXT Resource Record. In most instances investigated, only a few TXT records carrying legitimate information had up to 255 characters.

   Based on the analysis conducted it can be deduced that TXT records with 255 characters are a strong indicator of a possible payload distribution channel (malicious or non-malicious), and therefore DNS records with this characteristic need further investigation for them to be deemed non-malicious.

   While in the case of DNSMessenger the commands were not encrypted and were in “plain-text”, there is no need to attempt to decrypt the contents of the TXT records as malicious actors can go to greater lengths to ensure that the contents are encrypted and at times the content may even be text that does not appear to have any meaningful content. In the case of Wekby, the commands like ”sifo” in TXT records appears to be meaningless, but these are core to the functioning of the payload distribution channel.

   Even though the channel is encrypted and there is no easy way to decrypt and get meaningful information from the contents of the TXT record, the existence/presence of payload distribution can be detected through the analysis of the pattern and

trends of DNS queries and responses contained in the passive DNS data.

5.3.3 Multi-Purpose Payload Distribution

The expectation was that certain channels would be used for various purposes/different payloads, but the analysis showed that these channels are normally used for targeted attacks and as such, are used for a single purpose to avoid detection. There was no instance where a malware domain was used for phishing and then also for payload distribution. Neither was there an instance where a malware domain was associated with various malware families. Using a channel for various purposes would increase the chances of detection.

5.4 Further Discussion

5.4.1 Passive DNS Data

It was essential to get real-world data for building a high-quality detection system. In this project, the researcher utilised passive DNS data collected by Farsight Security. This data is collected through its passive DNS technology from various contributors distributed worldwide. The researcher used six months of data for this project. One caveat of using Farsight data is that it does not provide end-user IP addresses in addition to the DNS server IP due to the necessity for privacy protection. Therefore, the detected payload distribution channels are identified by DNS server IPs rather than the client IPs.

5.4.2 Analysis

The analysis will be discussed as two separate categories, traffic analysis and payload analysis. For DNS traffic analysis, the count, frequency and other request attributes were considered. For payload analysis, the DNS payload for one or more requests and responses was analysed for indicators of payload distribution.

5.4.2.1 Traffic Analysis

The use of DNS as a malicious payload distribution channel was explored through the analysis of passive DNS traffic. The malicious payload distribution channels are characterised based on the behaviour of the DNS request and response messages. All the messages with TXT resource records were extracted, and those of a given domain name aggregated. Then, the request and response pattern analysis was applied on the domains, which were then labelled as one of the four exchange pattern types: Many-to-Many, Many-to-Single, Single-to-Many and Single-to-Single relations. The naming indicates how many subdomains the client sends and whether the server replies with one or several different TXT records.

The evaluation shows that the approach succeeds in determining different payload distribution channel patterns for malware families. The Many-to-Many pattern seems to be the best choice for malware families that want to reach as many using many clients that are sending the malware/payload. Nevertheless, the Single-to-Single pattern proved to be the most popular amongst malware samples under consideration. This pattern produced the least amount of traffic, compared to the Many-to-Many pattern. The low traffic volumes associated with Single-to-Single patterns help the malware to stay undetected, a characteristic feature for Advanced Persistent Threats (discussed in Section 5.4.3).

The aim was to build DNS zone profiles using the access counts of the resource records in order to detect payload distribution channels. The access counts indicate the number of requests for each domain.

5.4.2.2 Payload Analysis

The messages from the Traffic Analysis were further analysed by the payload distribution module. This analysis and investigation relied mostly on human intervention as various features were investigated to determine malicious activities that may be associated with particular domains. Using the DNS information obtained from malware samples, the history of identified domains, WHOIS information, associated name servers and the contents of the TXT resource records, the researcher was able to determine the infrastructure involved in the malicious network. The identified domains that are involved in the distribution of payload were also compared with both known legitimate and malicious domains. As a result, it was noted that in the case of malware domains the number of DNS queries for TXT records is unusually high. In contrast, the DNS queries received by the Alexa Top 1000 domains are generally for different resource records.

5.4.3 Advanced Persistent Threats

From the analysis conducted, it can be concluded that cybercriminals and botnet masters typically used DNS TXT records for targeted attacks that are commonly referred to as Advanced Persistent Threats (APTs). APTs aggressively pursue and compromise specific targets and often use social engineering techniques to gain an entry point into the target network, and then attempt to move laterally throughout the network to discover weaknesses in the network, distribute payload and exfiltrate sensitive information. This was highlighted by the DNSMessenger malware and CopyKittens (discussed in the previous chapter).

In the DNSMessenger malware sample analysed, the malware payload was not necessarily transferred from the endpoint to the command and control server via the DNS TXT transport layer as a part of a DNS request. Instead, the DNS TXT records were used to relay commands and further instructions. Since only 255 bytes of data can be transported in a DNS TXT request, this method is slow, and typically does not draw attention. As seen in the malware samples discussed in previous chapters, the malware used by these APT groups use various obfuscation techniques, such as string and payload obfuscation, and other anti-analysis tactics to deter the detection and analysis of the communication.

Although most DNS tunnelling techniques and payload distribution channels use TXT type queries in DNS that can maximise the payload in response packets, there are implementations that make use of DNS query types other than TXT such as A, AAAA, CNAME, NS, MX and so on. This research undertaking shows that unlike legitimate users that often use TXT, malicious users can also use other query types that are more difficult to detect.

Legitimate DNS traffic typically has a minimal payload. As a result, many approaches can detect payload distribution channels based on the size of the payload.

5.5 Limitations of this Research

There are limitations with the system. One of the limitations is that the system was unable to identify malware and distribution channels that mimic the DNS activities related to TXT resource records that belong to legitimate payload channels. Another limitation

is that the system relies on malware datasets and dynamic analysis in a sandboxed environment may not be able to detect all the behaviour of some malware families that have the ability to detect that they are in a sandboxed/virtual environment.

Another limitation is that the system cannot offer real-time solutions as the system is an offline one that relies on data being analysed only at the end of a given time period. The last limitation has to do with the shortcomings of the passive DNS systems themselves. The passive DNS systems collect DNS traffic through multiple sensors located at various locations. Some payload distribution channels and malware families may not use caching resolvers in their respective networks and instead, send their DNS queries to open resolvers directly. This would mean that the DNS traffic would bypass the sensors and therefore cannot be captured for analysis.

5.6 Conclusion

This chapter discussed the findings, answered the research questions and identified the limitations of the current research.

The next chapter will give a summary of the whole research, and provide a conclusion to the subject under consideration.

Chapter 6

Conclusion

6.1 Introduction

The DNS protocol has over 30 record types with many of the common ones being critical to core Internet services. The most commonly used resource records are types A, AAAA, CNAME, NS, MX and the TXT. The A record type maps a domain name to an IPv4 address while the AAAA record is used to map a domain to an IPv6 address. The CNAME record type is used to map a domain name to the canonical name. The MX record type is used to define mail servers for a domain. The NS record type is used to define authoritative name servers for a domain. The PTR or pointer record is commonly used to map an IP address to its domain name in what is commonly referred to as reverse lookup. The TXT record type is used to return text data. This record type has been leveraged for specific purposes such as Sender Policy Framework (SPF) for anti-spam.

For many other types, such as A, AAAA, or CNAME, the payload is carried in one or more label in the FQDN. The TXT record type is usually used in tunnelling applications. A typical DNS tunnel or payload distribution channel will use a DNS query to carry outbound payloads. In TXT record type, the payload can be embedded in the text. Unlike legitimate DNS queries that have consistency in query and response messages, a malicious payload distribution channel tends to change the payload from message to message.

The goal of this research was to analyse multi-purpose payload distribution channels using the DNS. The goal was approached by analysing historical DNS data in order to characterise and understand the behaviour and infrastructure used by the criminals who

use the DNS to distribute malicious payload. This was achieved through the analysis of the activities of TXT resource records during a six-month period (January 2017 to June 2017).

6.2 Dataset

The primary dataset was the passive DNS database obtained from FarSight Security. The passive DNS data carries rich traces of the activities of the Internet, and this historical DNS data is a powerful resource that can be used to detect and identify malicious domains that are used as a key platform for distributing malicious payloads and other forms of cyber-attacks.

6.3 System for Analysing Payload Distribution Channels

The research analysed DNS queries and responses in the passive DNS dataset, with the aim of detecting and analysing multi-purpose payload distribution channels. The system was composed of a module for DNS query and response analysis (Section 6.3.1), and another for detecting and analysing the malicious payload distribution channels (Section 6.3.2).

The system is basically composed of a DNS query and response message analysis sub-module where the pattern of each communication channel is analysed. After the DNS query and response message analysis, the DNS query and response messages were then sent to a zone responsible for analysis of the presence of a potential payload distribution channel. This zone extracted all the concerned DNS Resource Record (RR) activities for particular domains from the passive DNS data. The sub-module is responsible for determining the intensity of the payload distribution using the DNS (the intensity of a payload distribution channel measures the level of activity in a particular channel). Further to that, there was a need to do some filtering so that legitimate domains that may resemble multipurpose payload distribution channels may be filtered out.

6.3.1 Analysing Query and Response Patterns

To identify payload distribution activities in the DNS, certain parameters were analysed as they are often used for establishing DNS tunnels and channels. The goal of the pattern analysis module was to differentiate various behaviours (Single-to-Many, Many-to-Many, Single-to-Single, and Many-to-Single) of channels distributing payload. This was achieved through analysing the behaviour during the exchange of DNS queries and response messages. The analysis was based on the fact these channels are forced to exchange only small amounts of information within each DNS message due to the limitation in the size of the DNS response packet (512 bytes).

6.3.2 Payload Distribution Detection

After the DNS query and response patterns had been recognised, the contents of the TXT records were analysed using the module responsible for DNS zone analysis to characterise and understand the payload carried by these messages. Name servers play an important role in the DNS protocol and as such, malicious networks require access to these servers in order to be able to distribute their payload. When a name server has been configured to be an authoritative domain name server that manages payload distribution, cybercriminals can configure the zone file of that particular domain to contain all payloads for the information/commands to be delivered through the DNS.

During the analysis, malware datasets were used to obtain and understand how some malware families operate, and their behaviour was validated using the historical passive DNS data. The research also investigated differences between regular legitimate domains and malicious domains. For regular domains, the Alexa Top 1000 sites list was utilised as these domains are generally considered to be non-malicious websites used for various legitimate purposes. Malware domains used for malicious purposes were extracted from various sources. The access count for resource records for each domain (malware domain or regular domain) was retrieved to get a measurement that could be used to gain a deeper understanding of individual resource record activity of a particular domain.

6.4 Summary of Findings

The summary below discusses findings presented in Chapter 5 and answers the research question and the sub-questions that are the basis of this research.

From the analysis of the passive DNS data and information associated with the malicious domains, it was established that the domains that are used for distributing payload can be detected and identified using various features that are synonymous with malicious activities. TXT resource records were used in this research to characterise each distribution channel and establish whether a channel was distributing payload.

The abuse of the DNS can be quantified through the query and response pattern and this gives an indication of the parties involved. Using supplementary information from secondary sources, the infrastructure used by malicious actors can be identified.

From the TXT records that were analysed, possible encryption was observed especially for the domains that are associated with legitimate DNS Tunnelling. However, using the publicly available decryption and decoding tools the researcher was unable to get any meaningful information from the entries of TXT records. While the content of the TXT records could not be determined, in some of the instances, certain patterns like having 255 characters in a TXT record were sufficient indication to warrant further analysis and investigation because it was a deviation from normal usage of the TXT Resource Record.

The aim of the research undertaking was to detect and characterise payload distribution channels that are used for multiple purposes. Contrary to the expectation of the researcher, it was discovered that these payload distribution channels are usually used solely by malware families for one specific purpose and that, based on the analysis conducted, these channels are not used for other purposes.

6.5 Future Work

The analysis identified several significant challenges that need to be explored further in future work. First, concerning the passive DNS dataset, the researcher observed that these logs are not readily accessible and publicly available making it difficult to obtain access due to bureaucratic obstacles and the commercial interests of those who have such historical data. Such information is also not shared publicly due to privacy concerns and other legal issues related to the protection of personal data legislation enacted in various jurisdictions.

Manually crawling through the collected passive DNS dataset to analyse the payload distribution channels proved to be a challenge due to the large number of domains that were of interest. As such, automated and dynamic analysis should be explored in future so

that the detection rate and the dynamic nature of these malware families can be taken into consideration. The dynamic and automated analysis will hopefully detect even malicious services that have inbuilt mechanisms to avoid detection. The dynamic analysis could improve the process of identifying services that have specific targets as highlighted by APTs highlighted in previous chapters as this approach is not based on known signatures but is based on a rule set.

The research undertaking uses mainly the TXT records for identifying payload distribution channels and in some cases, domains may use different DNS resource record types at different times to perform malicious activities. Below are some of the items that need to be actioned to increase the scope and fine tune the analysis.

- Expand the scope: the analysis focused mainly on TLDs that are known to be hot-spots for malicious activities. Further analysis is also required to distinguish between legitimate domains that have been hijacked from those that are purely used for malicious purposes.

  Resources permitting, further research could expand the scope to involve the `.com` top level domain. There are constraints, however, with the nature of access given to the DNSDB as extracting the whole block of the `.com` domain has proven to be a futile exercise due to the volume of DNS traffic for the `.com` domain.

- Increase the coverage of analysis to include other Resource Record types so that other malicious channels can also be given consideration. While the aim was to detect payload distribution channels via TXT records, this has been found to be a very inefficient method that many cyber criminals are not willing to adopt. Using other resource records would increase the coverage and improve the detection rates as some malware families are increasingly adopting other DNS resource records in their operations.

- Refine the DNS zone analysis to involve classifiers and dynamic analysis to improve the detection rate. The current prototype system requires a high degree of user intervention.

- Based on the model proposed, the usage of open DNS resolvers and rogue resolvers may need to be investigated as they may shed light on some activities that are not detected and sensed by the Passive DNS sensors.

- The use of machine learning techniques could further improve classification of domains associated with malicious activities, particularly those that are associated with malicious payload distribution.

6.6 Conclusion

In conclusion, malicious payload distribution channels can be analysed through the use of historical passive DNS data. This historical database of Internet activities sheds light into malicious domains that have been used to distribute payload. By analysing this database and using other supplementary sources of information, the research was able to investigate the DNS tunnels and channels used by various malware families to distribute malicious content. Challenges and limitations were identified and possible future work and recommendations were given to address the shortcomings of this research.

References

Aiello, M., Merlo, A., and Papaleo, G. (2013). Performance assessment and analysis of DNS tunneling tools. *Logic Journal of the IGPL*, 21(4):592–602.

Aiello, M., Mongelli, M., and Papaleo, G. (2015). DNS tunneling detection through statistical fingerprints of protocol messages and machine learning. *International Journal of Communication Systems*, 28(14):1987–2002.

Aitchison, R. (2011). *Pro DNS and BIND 10*. Apress.

Alenazi, A., Traore, I., Ganame, K., and Woungang, I. (2017). Holistic Model for HTTP Botnet Detection Based on DNS Traffic Analysis. In *International Conference on Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*, pages 1–18. Springer.

Alieyan, K., ALmomani, A., Manasrah, A., and Kadhum, M. M. (2017). A survey of botnet detection based on DNS. *Neural Computing and Applications*, 28(7):1541–1558.

Almomani, A. (2018). Fast-flux hunter: A system for filtering online fast-flux botnet. *Neural Computing and Applications*, 29(7):483–493.

Anagnostopoulos, M., Kambourakis, G., Kopanos, P., Louloudakis, G., and Gritzalis, S. (2013). DNS amplification attack revisited. *Computers & Security*, 39:475–485.

Andrews, M. (1998). Negative Caching of DNS Queries (DNS NCACHE). RFC 2308, RFC Editor. Available online at `https://www.rfc-editor.org/rfc/rfc2308.txt`.

Antonakakis, M., Perdisci, R., Dagon, D., Lee, W., and Feamster, N. (2010). Building a Dynamic Reputation System for DNS. In *USENIX Security Symposium*, pages 273–290.

Bertino, E. and Islam, N. (2017). Botnets and internet of things security. *Computer*, 50(2):76–79.

Bilge, L., Kirda, E., Kruegel, C., and Balduzzi, M. (2011). EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis. *Network and Distributed System Security*. Available online at `http://www.cs.ucsb.edu/~chris/research/doc/ndss11_exposure.pdf`. Retrieved 10 May 2018.

Bilge, L., Sen, S., Balzarotti, D., Kirda, E., and Kruegel, C. (2014). EXPOSURE: a passive DNS analysis service to detect and report malicious domains. *ACM Transactions on Information and System Security (TISSEC)*, 16(4):14.

Binsalleeh, H., Kara, A. M., Youssef, A., and Debbabi, M. (2014). Characterization of covert channels in DNS. In *2014 6th International Conference on New Technologies, Mobility and Security (NTMS)*, pages 1–5. IEEE.

Brumaghin, E. (2016). Want Tofsee My Pictures? A Botnet Gets Aggressive. Available online at `https://blog.talosintelligence.com/2016/09/tofsee-spam.html`. Retrieved 4 December 2018.

Brumaghin, E. and Grady, C. (2017). Covert Channels and Poor Decisions: The Tale of DNSMessenger. Available online at `https://blog.talosintelligence.com/2017/03/dnsmessenger.html`. Retrieved 20 January 2018.

Brumaghin, E., Grady, C., and Maynor, D. (2017). Spoofed SEC Emails Distribute Evolved DNSMessenger . Available online at `https://blog.talosintelligence.com/2017/10/dnsmessenger-sec-campaign.html`. Retrieved 20 August 2018.

Butler, P., Xu, K., and Yao, D. D. (2011). Quantitatively analyzing stealthy communication channels. In *International Conference on Applied Cryptography and Network Security*, pages 238–254. Springer.

Celik, Z. B., McDaniel, P. D., Izmailov, R., Papernot, N., and Swami, A. (2016). Building Better Detection with Privileged Information. *Computing Research Repository*. Available online at `http://arxiv.org/abs/1603.09638`. Retrieved 29 September 2017.

Chahal, P. S. and Khurana, S. S. (2016). TempR: Application of stricture dependent intelligent classifier for fast flux domain detection. *International Journal of Computer Network and Information Security*, 8(10):37.

Cheshire, S. and Krochmal, M. (2013). DNS-Based Service Discovery. RFC 6763, RFC Editor. Available online at `http://www.rfc-editor.org/rfc/rfc6763.txt`.

Chirgwin, R. (2016). Palo Alto IDs another C&C over DNS attack. Available online at `https://www.theregister.co.uk/2016/05/26/palo_alto_ids_another_ccoverdns_attack/`. Retrieved 17 November 2018.

Chung, T., van Rijswijk-Deij, R., Choffnes, D., Levin, D., Maggs, B. M., Mislove, A., and Wilson, C. (2017). Understanding the role of registrars in DNSSEC deployment. In *Proceedings of the 2017 Internet Measurement Conference*, pages 369–383. ACM.

Cian, L., Dimiter, A., and Claudiu, T. (2016). MULTIGRAIN Point of Sale Attackers Make an Unhealthy Addition to the Pantry. Available online at `https://www.fireeye.com/blog/threat-research/2016/04/multigrain_pointo.html`. Retrieved 12 June 2018.

CIRCL (2017). CIRCL Passive DNS. Available online at `https://circl.lu/services/passive-dns/`. Retrieved 1 November 2017.

ClearSky (2017). Operation Wilted Tulip. Available online at `https://www.clearskysec.com/wp-content/uploads/2017/07/Operation_Wilted_Tulip.pdf`. Retrieved 20 August 2018.

CobaltStrike (2018). Adversary Simulations and Red Team Operations. Available online at `https://www.cobaltstrike.com/`. Retrieved 20 August 2018.

da Luz, P. M. (2013). *Botnet detection using passive DNS*. PhD thesis, Department of Computing Science, Radboud University, Nijmegen.

Damas, J., Graff, M., and Vixie, P. (2013). Extension mechanisms for DNS (EDNS (0)). RFC 6891, RFC Editor. Available online at `https://www.rfc-editor.org/rfc/rfc6891.txt`.

Dharna, P. S. (2017). Analysis and Detection of Botnets and Encrypted Tunnels. *International Journal of Computer Science Trends and Technology (IJCST)*, 5(2):211–217.

Dietrich, C. J., Rossow, C., Freiling, F. C., Bos, H., Van Steen, M., and Pohlmann, N. (2011). On Botnets that use DNS for Command and Control. In *2011 Seventh European Conference on Computer Network Defense (EC2ND)*, pages 9–16. IEEE.

Dooley, M. and Rooney, T. (2017). *DNS Security Management*. John Wiley & Sons.

Engelstad, P., Feng, B., van Do, T., et al. (2017). Detection of DNS tunneling in mobile networks using machine learning. In *International Conference on Information Science and Applications*, pages 221–230. Springer.

Farsight (2018). Frequently Asked Questions. *Farsight Security*. Available online at `https://www.farsightsecurity.com/faq/`. Retrieved 25 June 2018.

Frosch, T. (2012). Mining DNS-related Data for Suspicious Features. Master's thesis, Ruhr Universitat, Bochum.

Frosch, T., Kührer, M., and Holz, T. (2013). Predentifier: Detecting botnet C & C domains from passive DNS data. *Advances in IT Early Warning. Fraunhofer Verlag*, pages 1–14.

Fu, Y., Yu, L., Hambolu, O., Ozcelik, I., Husain, B., Sun, J., Sapra, K., Du, D., Beasley, C. T., and Brooks, R. R. (2017). Stealthy Domain Generation Algorithms. *IEEE Transactions on Information Forensics and Security*, 12(6):1430–1443.

Fukuda, K., Heidemann, J., and Qadeer, A. (2017). Detecting Malicious Activity With DNS Backscatter Over Time. *IEEE/ACM Transactions on Networking*, 25(5):3203–3218.

Gordon, J. (2013). Systems and methods for identifying a network. US Patent 8.353.007. Available online at `https://www.google.com/patents/US8353007`.

Grunzweig, J., Scott, M., and Lee, B. (2015). New Wekby Attacks Use DNS Requests As Command and Control Mechanism. Available online at `https://researchcenter.paloaltonetworks.com/2016/05/unit42-new-wekby-attacks-use-dns-requests-as-command-and-control-mechanism`. Retrieved 20 June 2018.

Hachem, N., Mustapha, Y. B., Granadillo, G. G., and Debar, H. (2011). Botnets: lifecycle and taxonomy. In *2011 Conference on Network and Information Systems Security (SAR-SSI)*, pages 1–8. IEEE.

Hands, N. M., Yang, B., and Hansen, R. A. (2015). A study on botnets utilizing DNS. In *Proceedings of the 4th Annual ACM Conference on Research in Information Technology*, pages 23–28. ACM.

Haneef, A. (2016). *On the Scalable Generation of Cyber Threat Intelligence from Passive DNS Streams*. PhD thesis, Concordia University.

Hock, F. and Kortiš, P. (2016). Design, implementation and monitoring of the firewall system for a DNS server protection. In *2016 International Conference on Emerging eLearning Technologies and Applications (ICETA)*, pages 91–96. IEEE.

Hussain, M. A., Jin, H., Hussien, Z. A., Abduljabbar, Z. A., Abbdal, S. H., and Ibrahim, A. (2016). DNS Protection against Spoofing and Poisoning Attacks. In *2016 3rd International Conference on Information Science and Control Engineering (ICISCE)*, pages 1308–1312. IEEE.

Jalalzai, M., Shahid, W., and Iqbal, M. (2015). DNS security challenges and best practices to deploy secure DNS with digital signatures. In *2015 12th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pages 280–285. IEEE.

Jin, Y., Ichise, H., and Iida, K. (2015). Design of detecting botnet communication by monitoring direct outbound DNS queries. In *2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing (CSCloud)*, pages 37–41. IEEE.

Kalafut, A. J., Shue, C. A., and Gupta, M. (2011). Touring DNS open houses for trends and configurations. *IEEE/ACM Transactions on Networking (TON)*, 19(6):1666–1675.

Kara, A. M., Binsalleeh, H., Mannan, M., Youssef, A., and Debbabi, M. (2012). Detection of Malicious Payload Distribution Channels in DNS. Master's thesis, Concordia University.

Kara, A. M., Binsalleeh, H., Mannan, M., Youssef, A., and Debbabi, M. (2014). Detection of malicious payload distribution channels in DNS. In *2014 IEEE International Conference on Communications (ICC)*, pages 853–858. IEEE.

Khalil, I., Guan, B., Nabeel, M., and Yu, T. (2017). Killing Two Birds with One Stone: Malicious Domain Detection with High Accuracy and Coverage. *Cryptography and Security*. Available online at `https://arxiv.org/abs/1711.00300`. Retrieved 10 May 2018.

Khalil, I., Yu, T., and Guan, B. (2016). Discovering malicious domains through passive DNS data graph analysis. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*, pages 663–674. ACM.

Kikuchi, H. and Arimizu, T. (2014). On the Vulnerability of Ghost Domain Names. In *2014 Eighth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS)*, pages 584–587. IEEE.

King, Z. M., Henshel, D. S., Flora, L., Cains, M. G., Hoffman, B., and Sample, C. (2018). Characterizing and Measuring Maliciousness for Cybersecurity Risk Assessment. *Frontiers in Psychology*, 9:39.

Klein, A., Shulman, H., and Waidner, M. (2017). Internet-wide study of DNS cache injections. In *INFOCOM 2017-IEEE Conference on Computer Communications, IEEE*, pages 1–9. IEEE.

Koc, Y., Jamakovic, A., and Gijsen, B. (2012). A global reference model of the Domain Name System. *International Journal of Critical Infrastructure Protection*, 5(3):108–117.

Krämer, L., Krupp, J., Makita, D., Nishizoe, T., Koide, T., Yoshioka, K., and Rossow, C. (2015). Amppot: Monitoring and defending against amplification ddos attacks. In *International Workshop on Recent Advances in Intrusion Detection*, pages 615–636. Springer.

Kührer, M., Hupperich, T., Bushart, J., Rossow, C., and Holz, T. (2015). Going wild: Large-scale classification of open DNS resolvers. In *Proceedings of the 2015 Internet Measurement Conference*, pages 355–368. ACM.

Kumar, V., Kumar, S., and Gupta, A. K. (2016). Real-time Detection of Botnet Behavior in Cloud Using Domain Generation Algorithm. In *Proceedings of the International Conference on Advances in Information Communication Technology & Computing*, page 69. ACM.

Kuyama, M., Kakizaki, Y., and Sasaki, R. (2016). Method for detecting a malicious domain by using whois and dns features. In *The Third International Conference on Digital Security and Forensics (DigitalSec2016)*, page 74.

Kwon, J., Lee, J., Lee, H., and Perrig, A. (2016). PsyBoG: A scalable botnet detection method for large-scale DNS traffic. *Computer Networks*, 97:48–73.

Larsen, C. (2018). The "Top 20": Shady Top-Level Domains. Available online at `https://www.symantec.com/blogs/feature-stories/top-20-shady-top-level-domains`. Retrieved 20 August 2018.

Lauinger, T., Onarlioglu, K., Chaabane, A., Robertson, W., and Kirda, E. (2016). Whois lost in translation:(mis) understanding domain name expiration and re-registration. In *Proceedings of the 2016 Internet Measurement Conference*, pages 247–253. ACM.

Lee, B. (2015). Connecting the Dots in Cyber Threat Campaigns, Part 2: Passive DNS. Available online at `https://researchcenter.paloaltonetworks.com/2015/11/connecting-the-dots-in-cyber-threat-campaigns-part-2-passive-dns/`. Retrieved 29 September 2017.

Li, X., Wang, J., and Zhang, X. (2017). Botnet Detection Technology Based on DNS. *Future Internet*, 9(4):55.

Liska, A. and Stowe, G. (2016). *DNS Security: Defending the Domain Name System*. Syngress.

Liu, B., Lu, C., Li, Z., Liu, Y., Duan, H., Hao, S., and Zhang, Z. (2018). A Reexamination of Internationalized Domain Names: the Good, the Bad and the Ugly. In *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 654–665. IEEE.

Liu, C. (2015). Strengthen your network security with Passive DNS. Available online at `https://www.infoworld.com/article/2994016/network-security/strengthen-your-network-security-with-passive-dns.html`. Retrieved 29 September 2017.

Liu, C. (2016). Actively boosting network security with passive DNS. *Network Security*, 2016(5):18–20.

Liu, D., Hao, S., and Wang, H. (2016). All your DNS records point to us: Understanding the security threats of dangling DNS records. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1414–1425. ACM.

Liu, J., Li, S., Zhang, Y., Xiao, J., Chang, P., and Peng, C. (2017). Detecting DNS Tunnel through Binary-Classification Based on Behavior Features. In *2017 IEEE Trustcom/BigDataSE/ICESS*, pages 339–346. IEEE.

Lu, K., Li, Z., Zhang, Z., and Shi, J. (2016). DNS recursive server health evaluation model. In *2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pages 1–4. IEEE.

Manadhata, P. K., Yadav, S., Rao, P., and Horne, W. (2014). Detecting malicious domains via graph inference. In *European Symposium on Research in Computer Security*, pages 1–18. Springer.

Marchal, S., François, J., Wagner, C., and Engel, T. (2012). Semantic exploration of DNS. In *International Conference on Research in Networking*, pages 370–384. Springer.

Marrison, C. (2014). DNS as an attack vector–and how businesses can keep it secure. *Network Security*, 2014(6):17–20.

Marrison, C. (2015). Understanding the threats to DNS and how to secure it. *Network Security*, 2015(10):8–10.

Martinez-Bea, S., Castillo-Perez, S., and Garcia-Alfaro, J. (2013). Real-time malicious fast-flux detection using DNS and bot related features. In *2013 Eleventh Annual International Conference on Privacy, Security and Trust (PST)*, pages 369–372. IEEE.

Mc Carthy, S. M., Sinha, A., Tambe, M., and Manadhata, P. (2016). Data Exfiltration Detection and Prevention: Virtually Distributed POMDPs for Practically Safer Networks. In *International Conference on Decision and Game Theory for Security*, pages 39–61. Springer.

Merlo, A., Papaleo, G., Veneziano, S., and Aiello, M. (2011). A comparative performance evaluation of DNS tunneling tools. In *Computational Intelligence in Security for Information Systems*, pages 84–91. Springer.

Mohan, J., Puranik, S., and Chandrasekaran, K. (2015). Reducing DNS cache poisoning attacks. In *2015 International Conference on Advanced Computing and Communication Systems*, pages 1–6. IEEE.

Mullaney, C. (2011). Morto worm sets a (DNS) record. Available online at `https://www.symantec.com/connect/blogs/morto-worm-sets-dns-record`. Retrieved 20 January 2018.

Munkhbaatar, B., Mimura, M., and Tanaka, H. (2017). Dark Domain Name Attack: A New Threat to Domain Name System. In *International Conference on Information Systems Security*, pages 405–414. Springer.

Nadler, A., Aminov, A., and Shabtai, A. (2017). Detection of Malicious and Low Throughput Data Exfiltration over the DNS Protocol. *Cryptography and Security*. Available online at `https://arxiv.org/abs/1709.08395`. Retrieved 29 September 2017.

Neustar (2017). Cloud-based Authoritative DNS for Faster, Safer and Accurate Query Resolution . Available online at `https://www.security.neustar/resources/product-literature/ultradns-solution-sheet`. Retrieved 20 August 2018.

Noroozian, A., Korczyński, M., Gañan, C. H., Makita, D., Yoshioka, K., and van Eeten, M. (2016). Who gets the boot? Analyzing victimization by DDOS-as-a-service. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 368–389. Springer.

Nuojua, V., David, G., and Hämäläinen, T. (2017). DNS Tunneling Detection Techniques–Classification, and Theoretical Comparison in Case of a Real APT Campaign. In

Internet of Things, Smart Spaces, and Next Generation Networks and Systems, pages 280–291. Springer.

Pa, Y. M. P., Yoshioka, K., and Matsumoto, T. (2015). Detecting malicious domains and authoritative name servers based on their distinct mappings to IP addresses. *Journal of Information Processing*, 23(5):623–632.

Pomorova, O., Savenko, O., Lysenko, S., Kryshchuk, A., and Bobrovnikova, K. (2016). Anti-evasion technique for the botnets detection based on the passive DNS monitoring and active DNS probing. In *International Conference on Computer Networks*, pages 83–95. Springer.

Raman, D., De Sutter, B., Coppens, B., Volckaert, S., De Bosschere, K., Danhieux, P., and Van Buggenhout, E. (2012). DNS tunneling for network penetration. In *International Conference on Information Security and Cryptology*, pages 65–77. Springer.

Sammour, M., Hussin, B., and Othman, F. I. (2017). Comparative Analysis for Detecting DNS Tunneling Using Machine Learning Techniques. *International Journal of Applied Engineering Research*, 12(22):12762–12766.

Scheffler, S., Smith, S., Gilad, Y., and Goldberg, S. (2018). The Unintended Consequences of Email Spam Prevention. In *International Conference on Passive and Active Network Measurement*, pages 158–169. Springer.

Scott, K. (2014). Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1. RFC 7208, RFC Editor. Available online at `http://www.rfc-editor.org/rfc/rfc7208.txt`.

Settle, A., Dey, B., Griffin, N., and Toro, A. (2015). An analysis of Botnet Campaign:Jaku. Available online at `https://www.forcepoint.com/sites/default/files/resources/files/report_jaku_analysis_of_botnet_campaign_en_0.pdf`. Retrieved 20 June 2018.

Shafieian, S., Smith, D., and Zulkernine, M. (2017). Detecting DNS Tunneling Using Ensemble Learning. In *International Conference on Network and System Security*, pages 112–127. Springer.

Shah, B. (2017). Cisco Umbrella: A Cloud-Based Secure Internet Gateway (SIG) On and Off Network. *International Journal*, 8(2).

Sharifnya, R. and Abadi, M. (2015). DFBotKiller: Domain-flux botnet detection based on the history of group activities and failures in DNS traffic. *Digital Investigation*, 12:15–26.

Sharma, P., Kumar, S., and Sharma, N. (2016). BotMAD: Botnet malicious activity detector based on DNS traffic analysis. In *2016 2nd International Conference on Next Generation Computing Technologies (NGCT)*, pages 824–830. IEEE.

Sheridan, S. and Keane, A. (2015). Detection of DNS Based Covert Channels. In *European Conference on Cyber Warfare and Security*, page 267. Academic Conferences International Limited.

Sheridan, S. and Keane, A. (2017). Improving the Stealthiness of DNS-Based Covert Communication. In *ECCWS 2017 16th European Conference on Cyber Warfare and Security*, page 433. Academic Conferences and Publishing Limited.

Soltanaghaei, E. and Kharrazi, M. (2015). Detection of fast-flux botnets through DNS traffic analysis. *Scientia Iranica. Transaction D, Computer Science & Engineering, Electrical*, 22(6):2389.

Soltani, S., Seno, S. A. H., Nezhadkamali, M., and Budiarto, R. (2014). A survey on real world botnets and detection mechanisms. *International Journal of Information and Network Security*, 3(2):116.

Sood, A. K. and Zeadally, S. (2016). A taxonomy of domain-generation algorithms. *IEEE Security & Privacy*, 14(4):46–53.

Trevisan, M., Drago, I., Mellia, M., and Munafò, M. M. (2017). Automatic detection of DNS manipulations. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 4010–4015. IEEE.

Truong, D.-T. and Cheng, G. (2016). Detecting domain-flux botnet based on DNS traffic features in managed network. *Security and Communication Networks*, 9(14):2338–2347.

Valisenko, R. (2013). An Analysis of PlugX Malware. Available online at `https://www.lastline.com/labsblog/an-analysis-of-plugx-malware/`. Retrieved 20 June 2018.

van Beijnum, I. (2006). *The DNS*, pages 99–116. Springer.

van Rijswijk-Deij, R. (2017). *Improving DNS security: A Measurement-based Approach*. PhD thesis, University of Twente.

van Rijswijk-Deij, R., Sperotto, A., and Pras, A. (2014). DNSSEC and its potential for DDoS attacks: a comprehensive measurement study. In *Proceedings of the 2014 Conference on Internet Measurement Conference*, pages 449–460. ACM.

Wang, Z. (2014). POSTER: on the capability of DNS cache poisoning attacks. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pages 1523–1525. ACM.

Wang, Z. (2016). Combating malicious DNS tunnel. *Cryptography and Security*. Available online at `https://arxiv.org/abs/1605.01401`. Retrieved 10 May 2018.

Wu, H., Dang, X., Zhang, L., and Wang, L. (2015). Kalman filter based DNS cache poisoning attack detection. In *2015 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1594–1600. IEEE.

Xu, K., Butler, P., Saha, S., and Yao, D. (2013). Dns for massive-scale command and control. *IEEE Transactions on Dependable and Secure Computing*, 10(3):143–153.

Yates, A. (2017). More info on Evolved DNSMessenger . Available online at `https://wraithhacker.com/2017/10/11/more-info-on-evolved-dnsmessenger/`. Retrieved 20 August 2018.

Your-Freedom (2017). VPN tunneling, anonymisation and anti-censorship. Available online at `https://your-freedom.net/`. Retrieved 20 August 2018.

Yu, B., Smith, L., and Threefoot, M. (2014). Semi-supervised time series modeling for real-time flux domain detection on passive DNS traffic. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition*, pages 258–271. Springer.

Yu, B., Smith, L., Threefoot, M., and Olumofin, F. G. (2016). Behavior Analysis based DNS Tunneling Detection and Classification with Big Data Technologies. In *IoTBD*, pages 284–290.

Zhang, Z.-K., Cho, M. C. Y., Wang, C.-W., Hsu, C.-W., Chen, C.-K., and Shieh, S. (2014). IoT security: Ongoing Challenges and Research Opportunities. In *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications (SOCA)*, pages 230–234. IEEE.

Zhauniarovich, Y., Khalil, I., Yu, T., and Dacier, M. (2018). A Survey on Malicious Domains Detection through DNS Data Analysis. *ACM Computing Survey*. Available online at `https://arxiv.org/pdf/1805.08426.pdf`. Retrieved 10 May 2018.