Author(s): Nuojua, Viivi; David, Gil; Hämäläinen, Timo

Title: DNS Tunneling Detection Techniques – Classification, and Theoretical Comparison in Case of a Real APT Campaign

Year: 2017

Version:

Please cite the original version:

Nuojua, V., David, G., & Hämäläinen, T. (2017). DNS Tunneling Detection Techniques – Classification, and Theoretical Comparison in Case of a Real APT Campaign. In O. Galinina, S. Andreev, S. Balandin, & Y. Koucheryavy (Eds.), NEW2AN 2017, ruSMART 2017, NsCC 2017 : Internet of Things, Smart Spaces, and Next Generation Networks and Systems (pp. 280-291). Springer International Publishing. Lecture Notes in Computer Science, 10531. https://doi.org/10.1007/978-3-319-67380-6_26

DNS Tunneling Detection Techniques – Classification, and Theoretical Comparison in Case of a Real APT Campaign

Viivi Nuojua, Gil David and Timo Hämäläinen

University of Jyväskylä, P.O. Box 35, FI-40014 University of Jyväskylä, Finland
{ viivi.nuojua, gil.david, timo.t.hamalainen }@jyu.fi

Abstract. Domain Name System (DNS) plays an important role as a translation protocol in everyday use of the Internet. The purpose of DNS is to translate domain names into IP addresses and vice versa. However, its simple architecture can easily be misused for malicious activities. One huge security threat concerning DNS is tunneling, which helps attackers bypass the security systems unnoticed. A DNS tunnel can be used for three purposes: as a command and control channel, for data exfiltration or even for tunneling another protocol through it. In this paper, we surveyed different techniques for DNS tunneling detection. We classified those first based on the type of data and then within the categories based on the type of analysis. We conclude with a comparison between the various detection techniques. We introduce one real Advanced Persistent Threat campaign that utilizes DNS tunneling, and theoretically compare how well the surveyed detection techniques could detect it.

Keywords: DNS Tunneling Detection, Covert Channels Detection, APT.

1 Introduction

Domain Name System (DNS) is an important protocol used when accessing the Internet. It translates domain names (e.g. www.google.com) into IP addresses and vice versa, and therefore it acts as an interpreter between the user and the Internet.

DNS is a distributed hierarchical protocol. It means that groups of servers scattered around the world are responsible for certain domains. The hundreds of authoritative name servers around the world serve the 13 named root name servers (i.e. DNS root zone), from A to M. Every new request for an IP address goes first to a root name server, which then forwards the request to the next server in the hierarchy and so on. DNS servers also cache request results for later use to avoid unnecessary search. [1]

DNS is a simple and popular request and response protocol and hence can quite easily be misused. It is, indeed, widely used by the attackers for malicious activities. One huge security threat concerning DNS is tunneling. A DNS tunnel can be used for three purposes: “command and control” (C&C), data exfiltration and even for tunneling another protocol through it [1]. A typical example is to bypass a Wi-Fi access fee by

tunneling the network traffic through DNS protocol. Tunnels over DNS are very powerful because DNS requests are seldom blocked by firewalls. [2] DNS tunneling has been utilized in some Advanced Persistent Threat (APT) attacks, which are stealthier and more sophisticated continuous attacks with a specific target. Unfortunately, DNS is often the least supervised protocol in many organizations.

The rest of the paper is organized as follows. Section II shortly reviews one real APT campaign utilizing DNS tunneling. Section III describes a classification for DNS tunneling detection techniques. Section IV provides our theoretical comparison of DNS tunneling detection techniques in case of the reviewed APT campaign. Finally, Section V concludes the paper with some future work ideas.

2 Example of a Real APT Campaign Utilizing DNS Tunneling

Network security company Palo Alto Networks discovered a DNS tunneling utilizing APT attack in May 2016. The group behind this APT campaign was Wekby (aka TG-0416, APT-18, Dynamite Panda [3]), which has been quite active during the past years. It has maliciously taken advantage of the recently released exploits and targeted various organizations in the field of manufacturing, technology and healthcare. Although Wekby seems to use somewhat primitive methods, such as stolen credentials or basic malwares, it is constantly improving its toolkit. [4]

Palo Alto Networks named the malware family used in the attack as "Pisloader". It was delivered via HTTP from the specific URL, and the used domains globalprint-us.com, ns1.logitech-usa.com and intranetwabcam.com were registered just before the attack. The executable file containing the Pisloader payload has similarities with the well-known Poison Ivy malware family. In addition, based on the metadata of the Pisloader sample and its malware commands, Pisloader is inferred to be a variant of the HTTPBrowser [5] malware family. In fact, HTTPBrowser was led by Wekby too. [4]

Pisloader utilizes DNS tunneling not only as a C&C channel but also for data exfiltration. The malware payload can be transferred from the endpoint to the C&C server via DNS TXT transport layer as a part of a DNS request. A DNS request for TXT record consists of a randomized header and a base32-encoded data part with padding removed. The C&C server responds with a DNS TXT record with similar encoding. Only 255 bytes of data can be transported in a DNS TXT request. It is slow but won't draw attention. Additionally, Pisloader uses various obfuscation techniques, such as string and payload obfuscation, and other anti-analysis tactics to deter the detection and analysis of the sample. [4]

3 Classification of DNS Tunneling Detection Techniques

Farnham & Atlasis [1] divide DNS tunneling detection techniques into two categories: traffic inspection and payload inspection. In this paper, we use the same classification into traffic and payload inspection, based on the type of data that is being analyzed. However, the grouping of techniques within the categories is implemented differently.

3.1 Traffic Inspection

According to Farnham & Atlasis [1] DNS tunneling can be detected by inspecting multiple requests or overall traffic, i.e. by traffic inspection. The following attributes are used: volume of DNS traffic per IP address, volume of DNS traffic per domain, number of hostnames per domain, geographic location of DNS server, domain history, volume of NXDomain responses, visualization, orphan DNS requests and general covert channel detection. Here, we group the detection techniques based on the type of analysis.

Statistical Analysis. A technique based on the statistical flow data analysis is introduced in [6], where flow is the sequence of packets exchanged between two hosts associated with a single service [7]. The dataset consists of unsampled flow records, each containing the following information about the flow: source IP address, destination IP address, source port, destination port, number of packets and bytes, start and end time of the flow, protocol, and the flags used (in case of TCP protocol).

They propose a module called Tunneling Attack Detector (TUNAD) for DNS tunneling detection. Its purpose is to detect anomalies in the statistics of the DNS packet sizes near real-time. TUNAD separates DNS packets to requests, responses, and unknown query types (uqr) when it is not known for sure whether those are requests or responses. For each of these packet types and each monitored circuit (receive or transmit) TUNAD calculates hourly packet size histograms from flow records. In addition, for each packet type TUNAD calculates the frequency of the packets deviated from the normal UDP/DNS packet size, that is, for responses and uqr packet types over 512 bytes and for requests over 300 bytes. The module is supposed to detect changes in the frequency of large packets. The authors managed to detect the Sinit virus in 2003 before it was reported in any external reports. No performance metrics were introduced.

Another flow-based statistical approach to detect DNS tunnels was presented in [2]. The novelty of the approach lies in combining flow information with statistical methods for anomaly detection. The four collected datasets consist of one dataset including normal DNS traffic and three datasets including tunneled traffic from three DNS tunnel misuse scenarios: C&C, data exfiltration and web browsing. These scenarios are tested by using four different DNS resolvers: a local resolver, a Norton open DNS resolver, a resolver in the network of the tunnel client and a resolver at the tunnel server. A local resolver behaves as normal DNS traffic, i.e. uses a new port number for each request. The latter three are non-local resolvers that reuse the same port number.

The analysis is based both on the raw data (unprocessed flows) and on time-binned data (preprocessed time series data). In case of normal DNS traffic, the time-of-the-day-dependency can be clearly seen. In case of tunneled traffic, the choice of resolver influences the behavior. The fact that the use of a local resolver makes tunneled traffic behave as normal traffic can easily be identified from the results. It was also observed that the number of packets sized approximately 200 bytes increase in case of a tunnel.

The authors use the following methods to detect anomalies in DNS traffic: 1) Threshold method, 2) Brodsky-Darkhovsky method [8] (BD) and 3) Distribution-based method. The anomalies under observation in each case are 1) peaks in the traffic time series, 2) changes in the average amount of traffic and 3) changes in the underlying

traffic distribution. By combining different choices of input data (raw or time-binned flows), anomaly detection methods and resolvers, altogether five flow-based detectors were implemented to detect DNS tunnels. The results show that all five detectors manage to detect various tunnel misuse scenarios with high detection rates.

Machine Learning -based Analysis. A passive DNS monitoring solution is introduced in [9]. It aims to detect anomalies in DNS traffic by using data mining, particularly a classical clustering algorithm named k-means [10]. The purpose is to detect the tunneling of IP traffic over DNS traffic.

The two datasets that were used were carefully chosen such that they differ as much as possible in geographical location, volume, duration and access networks. The k-means algorithm is used for data clustering. The clusters given by the algorithm turn out to successfully cover all different types of DNS traffic activities. The authors had no trouble in detecting DNS tunnels, when observing at the statistical profiles of different DNS record types. The number of TXT records, namely, clearly increase in case of a tunnel. No performance metrics were introduced.

Another machine learning -based approach for DNS tunneling detection is introduced in [11, 12]. The authors aim to avoid the monitoring of individual sockets and instead exploit all the information exchanged by the DNS server. Thus, it is possible to achieve real-timeness. The focus is on the exploration of simple statistical properties of DNS queries and responses, such as packet inter-arrival time and packet size statistics, with the help of supervised machine learning. The concentration isn't only on anomaly detection, but in the utilization of other statistical methods too.

The DNS databases consist of three tunnel databases based on the tunneled application and two clean databases (without tunnel), small and medium DNS, based on the size of a server. The applications tunneled through DNS protocol are a wget dump of an entire website, SSH protocol and a peer to peer (p2p) application. The DNS tunnel traffic is produced by a DNS tunneling tool called Dns2tcp [13].

In the first paper [11], the authors use the Bayes classifier [14] for classifying the DNS traffic into normal and tunneled traffic. In the second paper [12], they introduce, in addition, three other machine learning -based classifiers: k-nearest neighbor [15], neural networks [16] and support vector machine [17]. Classification error is avoided with the help of high order statistics, such as skewness and kurtosis, to detect the data positioning far from the assumed Gaussian distribution, i.e. the tunnel points. If just one of the aforementioned classifiers detects a tunnel, the tunnel is considered as detected. Classification is successful despite of the classification error of 15 % in case of p2p application. The computational effort of the approach proves to be negligible. As a result, a reliable and fast DNS tunneling detection is achieved.

In the most recent paper surveyed in here [18], the authors use machine learning methods for anomaly behavior analysis of DNS protocol. The purpose is to learn the normal and abnormal behavior of DNS and develop an anomaly metric to detect DNS tunneling and other exploitations of the DNS protocol. The analysis is targeted at different levels of the domain name hierarchy and at the temporal transitions of normal DNS traffic.

Supervised machine learning methods, such as bagging classification algorithm [19], are used for observing and n-gram approach [20] for modeling the state transitions of the DNS protocol. A DNS state machine can be built based on the observed and analyzed DNS messages. In the first stage of the training phase, data structures called dnsgrams are used for capturing important properties of consecutive DNS queries and replies. The dnsgrams of DNS protocol under attack differ clearly from the normal dnsgrams, because the protocol doesn't follow the protocol state machine anymore. In the second stage of the training phase, dnsgrams are grouped periodically to dnsflows that cover a time window of t seconds, and the statistical properties of those are obtained. Finally, dnsflows are classified as normal and abnormal by comparing the extracted query indexes of a particular flow with the corresponding values of normal DNS operations. According to the authors, their general approach is able to detect both known and unknown attacks against DNS protocol. They managed to detect the executed DNS tunneling attack in every case. For known DNS attacks, such as DNS tunneling, the detection rate was high (97 %) and the false positive rate low (0.014 %). However, their focus was not only in DNS tunneling attacks.

3.2 Payload Inspection

In addition to traffic inspection, a DNS tunnel can be detected by inspecting a single request. This so called payload inspection uses the following attributes: size of request and response, entropy of hostnames, statistical analysis, uncommon record types, policy violation and specific signatures. [1] In this section, we introduce a different grouping based on the type of analysis. All the detection techniques introduced under the title "Statistical Analysis" can also be classified as machine learning -based analysis.

Statistical Analysis. The form of statistical analysis, character frequency analysis, is widely used for the detection of DNS tunneling. It was first introduced in the context of DNS tunneling detection in [21, 22]. In the first paper [21], the authors introduce the ability to detect DNS tunnels by analyzing the unigram (single character), bigram (two characters) and trigram (three characters) character ranks and frequencies of domain names. The approach is based on the assumptions that domain names behave similarly as natural languages, i.e. follow Zipf's law [23], whereas tunneled traffic clearly shows different, random-like, behavior.

First, the unigram, bigram and trigram character ranks and frequencies of domains are compared with the corresponding ranks and frequencies of English language. The top 1 000 000 most popular domains in 2009 (Alexa Top Sites) are used. The Top-Level Domains (TLD) and Lowest Level Domains (LLD), i.e. the furthest right and left parts of the domain names are removed. For example, in the domain name "www.example.com" TLD is "com" and LLD is "www". After removing the TLDs and LLDs, the unigram character ranks and frequencies of popular domains are compared with the unigram character ranks and frequencies of new domain names with randomly generated characters. Next, the unigram and bigram character ranks and frequencies of 1 000 000 most popular domains and 100 most popular domains are compared. On the other

hand, a comparison between the unigram character ranks and frequencies of 100 randomly generated domain names and 1 000 000 popular domains is made. The purpose of these comparisons is to ensure that there won't be any problems with the character ranks and frequencies being skewed toward the outliers in the data, i.e. a smaller amount of data is sufficient to separate normal and tunneled traffic. In the detection of DNS tunneling, subdomains are the most critical ones to be analyzed. Consequently, the unigram character ranks and frequencies of popular domains are compared with subdomains and name server subdomains, with TLD and the following label removed from the domain names. The used subdomains and name server subdomains are captured by a custom website crawler visiting the aforementioned most popular sites. Finally, in order to verify the legitimacy of the introduced character frequency analysis, the unigram ranks and frequencies of subdomains are compared with the unigram ranks and frequencies of 100 Iodine [24] / Dns2tcp / TCP-over-DNS [25] packets. Thus, altogether three popular DNS tunneling tools are used.

The results support the assumptions made by the authors. The unigram character ranks and frequencies of English language and popular domain names are very much alike. The bigram and trigram character frequencies, in turn, are not that similar in ranks between English language and popular domain names, but still follow quite a similar pattern. On the other hand, the unigram character ranks and frequencies of popular domains and randomly generated domains have almost nothing in common. Since the authors presume that the randomly generated domains can represent the tunneled traffic, it seems to be quite easy to tell whether there's a tunnel or not. A comparison between the unigram and bigram character ranks and frequencies of 1 000 000 and 100 most popular domains shows that especially the unigram character ranks and frequencies are very similar. However, in case of bigrams the situation isn't bad at all either. On the contrary, there's almost no correlation between the unigram character ranks and frequencies of 100 randomly generated domain names and 1 000 000 popular domains. Consequently, a smaller amount of data is sufficient to make a clear separation between normal and tunneled traffic. The unigram character ranks and frequencies of subdomains turn out to be much alike and name server subdomains quite alike compared to popular domains. Only the ranks of the characters "n" and "s" seem to stand out in case of name server domains. Finally, subdomains have surprisingly similar unigram character ranks to the tunneling tools. Nevertheless, by inspecting the change in frequency between ranks it is easy to see the difference between normal and tunneled traffic.

As an extension to the authors' previous work, a tool called NgViz is introduced in [22]. It automates the visualization and analysis of DNS traffic. In order to detect anomalies in the n-gram frequencies, NgViz compares the input files with the fingerprint of legitimate traffic. A hash map creates a new entry for each unique n-gram by storing the number of times it is seen in the input file. The authors recommend to remove the duplicate subdomains and analyze the traffic only in case of a new subdomain.

First, the rank of each n-gram is compared with the corresponding n-gram rank in the fingerprint. After that, the similarities between the input and fingerprint files are explored by comparing the drops in frequency from one rank to the next. Significant differences may indicate to a tunnel in the DNS traffic. The detection of tunnels masked by a large amount of legitimate traffic is possible by splitting the traffic into separate

files based on IP addresses, domains or specified time intervals. The results are sorted into highest and lowest matches for the fingerprint, making it easier to detect the anomaly parts of the traffic. The problem of the approach is the skewness caused by a domain consisting of similarly labeled subdomains. Therefore, the frequencies of the n-grams in that label grow so much that the domain is falsely considered as a tunnel.

The list of 1 000 000 most popular domains in 2009 (Alexa Top Sites) is utilized to form the fingerprint file of legitimate traffic. As in their previous paper [21] a custom website crawler is used to simulate a typical internet traffic by visiting thousands of popular websites, i.e. acting like a typical internet user. In the first experiment, the popular DNS tunneling tools (Iodine, TCP-over-DNS and Dns2tcp) show a poor match and the simulated traffic split every 100 unique subdomains a good match for the popular domains. In the second experiment, the 500 unique subdomains using bigrams show a great match and tunneling tools a very poor match for the popular domains. Because of this and the fact that bigrams provide significantly more data points and that way decrease the matching of tunneled and legitimate domain names, the results are even clearer. The performance evaluation is left for future research.

Character frequency analysis was used for detecting DNS tunnels in [26] too. The key idea of their approach is the score mechanism, which is able to separate normal and tunnel domain names based on bigram character frequency in real-time. Here the score of a domain is defined as the expected value of bigram character frequency. As in [21, 22], it is assumed that the normal domain names follow Zipf's law and the character frequencies of DNS tunnel domain names follow a random distribution.

Once again, the normal domain name dataset consists of the 1 000 000 most popular domains (Alexa Top Sites, 2012), and the DNS tunneling domain names are generated by DNS tunneling tool called DNScat [27]. The data are divided into two parts: training and classification. The TLDs are removed from the domain names, since those don't bring that much information about the domain. The scores are calculated for both normal and tunneled domain names based on the character frequencies of the bigrams. Then, a suitable threshold is calculated according to the training data such that the classifier produces least false positives. It seems that DNS tunnel domain names have low scores that are centered within a small range. Instead, the scores of normal domain names are widely distributed. In the evaluation part, the scores calculated for the classification data are compared to the previously determined threshold. Based on that, the domain is classified either as normal or tunnel. The performance evaluation shows that the approach gets notably high accuracy of 98.74 % and low false positive rate 1.24 %.

Signature-based Analysis. The use of DNS as a malicious payload distribution channel, e.g. for tunneling, was explored in [28, 29]. In the first paper [28], the malicious payload distribution channels are characterized based on the exchange behavior of the DNS request and response messages. First, all the messages with TXT resource record activities are extracted and those of a given domain name aggregated. Then, the request and response pattern analysis is applied on the domains, which are labeled as one of the four exchange pattern types: 1) Many-to-Many, 2) Many-to-Single, 3) Single-to-Many and 4) 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. Many-to-Many pattern seems to be the best choice for large datasets. Nevertheless, Single-to-Single pattern proves to be the most popular amongst malware samples. It produces the least amount of traffic, which helps the malware stay undetected. No performance metrics were introduced.

In the second paper [29], a DNS zone analysis -based approach is introduced. A near real-time detection is achieved by monitoring DNS requests and responses in passive DNS traffic. The purpose is to build DNS zone profiles out of the access counts of the resource records in order to detect payload distribution channels. The access counts indicate the number of requests for each record. The DNS request and response messages with TXT resource record activities are extracted and those of a given domain name aggregated. Finally, the messages are analyzed by the payload distribution module. A one-year malware database including malware samples that have TXT resource record activities is used for evaluation.

As a result, it is noted that in case of malware domains the number of DNS queries for TXT records is unusually high. The DNS queries received by the 500 most popular domains (Alexa Top Sites, 2014) are, in turn, for different resource records. The system is able to detect payload distribution channels regardless of the payload format and malware family. The authors were actually the first ones to detect some hidden domains used by the famous Morto worm. Additionally, they emphasize that their system is supposed to detect DNS tunneling based on other resource record type than TXT too, since the detection is based on the access counts. No performance metrics were introduced.

3.3 Hybrid Inspection

The last DNS tunneling detection technique presented in this paper combines both traffic and payload inspection [30]. The purpose is to detect tunnels and other anomalies via statistical analysis in huge volume of DNS traffic near real-time. The approach doesn't focus on pattern matching, but the major interest lies in the differences between each DNS request/response.

The dataset consists of basic flow records extended with DNS fields, and collected from the monitoring probes. The analysis is targeted at the fields of DNS messages, such as TXT records. The tunneling data are produced by two DNS tunneling tools: Iodine and Dns2tcp. The module consists of two detection modes, basic and advanced. In the basic mode, all the observed network addresses are analyzed based on the statistics of flow information. The suspicious addresses are led to the advanced mode, and considered as a tunnel if a certain predetermined threshold is exceeded. Whitelisting of addresses and domain names is supported in order to avoid falsely classifying legitimate traffic as tunneled traffic.

The authors aim for a minimal resource consumption by processing smaller time windows at a time. In addition, the memory consumption is restrained by storing and analyzing only the domain names with suspicious addresses. To make the processing of huge volume of data possible, the approach utilizes efficient extended data structures. The detection module was put on a test by inputting data including both legitimate and

tunneled traffic. It managed to detect all the tunnels established by the used tunneling tools. No performance metrics were introduced.

Table 1. introduces the aforementioned DNS tunneling detection techniques divided into two categories based on the type of data and then grouped based on the type of analysis. From the table, it can clearly be seen that some techniques cover more than one group. Especially, the hybrid detection technique [30] covers both traffic and payload inspection and both statistical and signature-based analysis. All the surveyed machine learning -based detection techniques can also be grouped into statistical analysis. Despite of the hybrid detection technique, the two techniques grouped into signature-based analysis, don't cover any other type of analysis.

4 Theoretical Comparison of DNS Tunneling Detection Techniques in Case of a Real APT Campaign

In this section, we theoretically compare how well the surveyed DNS tunneling detection techniques could detect the Pisloader malware attack introduced in section II.

The two, only statistical analysis -based traffic inspection techniques would succeed quite differently in the detection of Pisloader. In the first surveyed paper [6], DNS tunneling attack detector, TUNAD, regards frequent DNS requests with over 300 bytes of data as tunneled traffic. However, the Pisloader payload is transported in a DNS TXT request, which is limited to contain only 255 bytes of data. In addition, TUNAD doesn't seem to pay attention to TXT records. Thus, it wouldn't be able to detect the Pisloader attack. Pisloader utilizes DNS tunneling for C&C and data exfiltration, which are both taken into account in [2]. Unlike in case of TUNAD, the authors observed that an increase in the number of packets sized over 200 bytes indicates a tunnel. Because of that, their versatile use of different setups, analyses and methods, and high detection rates, they would manage to detect the Pisloader malware attack, in theory.

The four separate papers covering not only statistical but also machine learning -based traffic inspection techniques are all a bit uncertain in detecting the Pisloader malware. The statistical profiles of different DNS record types are explored in [9], and unlike TUNAD, the authors pay attention to TXT records. They noticed that the number of TXT records increase when a DNS tunnel is misused. However, they did not provide any threshold for packet sizes indicating a tunnel, nor performance metrics. Thus, Pisloader might stay undetected. The authors of [11, 12] focus on the simple statistical properties of DNS requests and responses, such as packet size statistics. According to them, large packet sizes refer to tunneling, but the byte numbers are different in case of each application tunneled through DNS. Thus, it is a bit difficult to evaluate whether they would be able to detect Pisloader or not. The properties of consecutive DNS requests and responses are examined in [18]. It is not clear if the authors actually utilize DNS TXT records as part of their detection method. Their focus was not only in the detection of DNS tunneling attacks but multiple different attacks. According to the authors, they managed to detect a certain executed DNS tunneling attack with high detection rate and low false positive rate. However, it is a bit difficult to evaluate if Pisloader would be detected too.

Table 1. Classification of DNS Tunneling Detection Techniques.

Detection approach	Traffic Inspection		Payload Inspection		
	Statistical Analysis	Machine Learning -based Analysis	Statistical Analysis	Signature-based Analysis	Machine Learning -based Analysis
[6]	X				
[2]	X				
[9]	X	X			
[11]	X	X			
[12]	X	X			
[18]	X	X			
[21]			X		X
[22]			X		X
[26]			X		X
[28]				X	
[29]				X	
[30]	X		X	X	

Character frequency analysis -based payload inspection techniques (covering both statistical and machine learning -based analysis) are quite tricky too when evaluating if the Pisloader attack would be detected, in theory. The authors of [21, 22] state that usually in case of a tunnel domain names seem randomly generated. The domain names used by Pisloader are not random-like, which is why Pisloader would stay undetected. Same goes for the detection technique presented in [26], although they better their detection possibilities by dividing data into training and classification parts, and by providing good performance metrics results.

A DNS tunneling detection technique that would possibly detect the Pisloader malware is introduced in [28, 29]. The authors' signature-based analysis focuses on monitoring DNS requests and responses. They noticed that in case of malware domains the number of DNS requests for TXT records is unusually high. By extracting DNS messages with TXT resource record activities, they managed to detect different payload distribution channel patterns regardless of the payload format and malware family. In fact, they were the first ones to detect some hidden domains used as malicious payload distribution channels by the famous Morto worm.

The hybrid inspection technique presented in [30] also pays attention to the differences between each DNS request/response, especially to the fields of DNS messages, such as TXT records. However, their detection module analyzes only the domain names that are classified as suspicious in advance. The domain names used by Pisloader might not be thought suspicious. Thus, Pisloader might stay undetected.

It seems that, in theory, the Pisloader malware attack would best be detected by the secondly surveyed, only statistical analysis -based traffic inspection technique [2]. In addition, the signature-based payload inspection techniques [28, 29] could theoretically be able to detect Pisloader. However, it is a bit difficult to evaluate the impact of the various obfuscation techniques and other anti-analysis tactics used by Pisloader.

5 Conclusions and Future Work

In this paper, we reviewed a real DNS tunneling utilizing APT campaign, called Pisloader. We surveyed several different DNS tunneling detection techniques and classified those, first, based on the type of data. Within those categories, we grouped the techniques based on the type of analysis. One of the problems when comparing the detection techniques was that all of those don't provide performance metrics and use different datasets. However, we made a theoretical comparison of how well the surveyed DNS tunneling detection techniques could detect the Pisloader malware attack. It came up quite clearly that the techniques with certain type of analysis approach could succeed better than others. Altogether, the surveyed techniques seem to cover the extensive range of different kind of DNS tunneling detection very well.

Next, our purpose is to actually implement, and compare in practice, the DNS tunneling detection techniques introduced in this paper, on the same dataset. That way, the actual comparison between their performance is made possible. Finally, we are going to present our own approach and compare it to the reviewed techniques.

References

1. Farnham, G., Atlasis, A.: Detecting DNS tunneling. Inst. InfoSec Read. Room. 1–32 (2013).
2. Ellens, W., Żuraniewski, P., Sperotto, A., Schotanus, H., Mandjes, M., Meeuwissen, E.: Flow-Based Detection of DNS Tunnels. In: Doyen, G., Waldburger, M., Čeleda, P., Sperotto, A., and Stiller, B. (eds.) Emerging Management Mechanisms for the Future Internet. pp. 124–135. Springer Berlin Heidelberg (2013).
3. Evasive Maneuvers by the Wekby group with custom ROP-packing and DNS covert channels, https://www.anomali.com/blog/evasive-maneuvers-the-wekby-group-attempts-to-evade-analysis-via-custom-rop.
4. New Wekby Attacks Use DNS Requests As Command and Control Mechanism, http://researchcenter.paloaltonetworks.com/2016/05/unit42-new-wekby-attacks-use-dns-requests-as-command-and-control-mechanism/, (2016).
5. Chinese Cyber Espionage APT Group Leveraging Recently Leaked Hacking Team Exploits To Target A Financial Services Firm, https://www.zscaler.com/blogs/research/chinese-cyber-espionage-apt-group-leveraging-recently-leaked-hacking-team-exploits-target-financial-services-firm.
6. Karasaridis, A., Meier-Hellstern, K., Hoeflin, D.: NIS04-2: Detection of DNS Anomalies using Flow Data Analysis. In: IEEE Global Telecommunications Conference, 2006. GLOBECOM '06. pp. 1–6 (2006).
7. III, J.A.C.: Flow-based detection of network intrusions, http://www.google.com/patents/US7185368, (2007).
8. Brodsky, E., Darkhovsky, B.S.: Nonparametric Methods in Change Point Problems. Springer Science & Business Media (2013).
9. Marchal, S., François, J., Wagner, C., State, R., Dulaunoy, A., Engel, T., Festor, O.: DNSSM: A large scale passive DNS security monitoring framework. In: Network Operations and Management Symposium (NOMS), 2012 IEEE. pp. 988–993. IEEE (2012).

10. Hartigan, J.A., Wong, M.A.: Algorithm AS 136: A K-Means Clustering Algorithm. J. R. Stat. Soc. Ser. C Appl. Stat. 28, 100–108 (1979).
11. Aiello, M., Mongelli, M., Papaleo, G.: Basic classifiers for DNS tunneling detection. In: 2013 IEEE Symposium on Computers and Communications (ISCC). pp. 000880–000885 (2013).
12. Aiello, M., Mongelli, M., Papaleo, G.: DNS tunneling detection through statistical fingerprints of protocol messages and machine learning. Int. J. Commun. Syst. 28, 1987–2002 (2015).
13. HSC - Tools - Dns2tcp, http://www.hsc.fr/ressources/outils/dns2tcp/.
14. Moore, A.W., Zuev, D.: Internet Traffic Classification Using Bayesian Analysis Techniques. In: Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems. pp. 50–60. ACM, New York, NY, USA (2005).
15. Cover, T., Hart, P.: Nearest neighbor pattern classification. IEEE Trans. Inf. Theory. 13, 21–27 (1967).
16. Ripley, B.D.: Pattern Recognition and Neural Networks. Cambridge University Press (2007).
17. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. 20, 273–297.
18. Satam, P., Alipour, H., Al-Nashif, Y., Hariri, S.: Anomaly Behavior Analysis of DNS Protocol. J. Internet Serv. Inf. Secur. JISIS. 5, 85–97 (2015).
19. Breiman, L.: Bagging predictors. Mach. Learn. 24, 123–140.
20. Bramer, M.: Principles of data mining. Springer (2007).
21. Born, K., Gustafson, D.: Detecting DNS Tunnels Using Character Frequency Analysis. ArXiv10044358 Cs. (2010).
22. Born, K., Gustafson, D.: NgViz: Detecting DNS Tunnels Through N-gram Visualization and Quantitative Analysis. In: Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research. p. 47:1–47:4. ACM, New York, NY, USA (2010).
23. Zipf, G.K.: Selected studies of the principle of relative frequencies of language. Camb. Mass. Harv. Unive. (1932).
24. kryo.se: iodine (IP-over-DNS, IPv4 over DNS tunnel), http://code.kryo.se/iodine/.
25. TCP-over-DNS tunnel software HOWTO, http://analogbit.com/2008/07/27/tcp-over-dns-tunnel-software-howto/.
26. Qi, C., Chen, X., Xu, C., Shi, J., Liu, P.: A bigram based real time DNS tunnel detection approach. Procedia Comput. Sci. 17, 852–860 (2013).
27. DNScat, http://tadek.pietraszek.org/projects/DNScat/.
28. Binsalleeh, H., Kara, A.M., Youssef, A., Debbabi, M.: Characterization of Covert Channels in DNS. In: 2014 6th International Conference on New Technologies, Mobility and Security (NTMS). pp. 1–5 (2014).
29. Kara, A.M., Binsalleeh, H., Mannan, M., Youssef, A., Debbabi, M.: Detection of malicious payload distribution channels in DNS. In: 2014 IEEE International Conference on Communications (ICC). pp. 853–858 (2014).
30. Cejka, T., Rosa, Z., Kubatova, H.: Stream-wise detection of surreptitious traffic over DNS. In: 2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD). pp. 300–304 (2014).