Behavior Analysis based DNS Tunneling Detection and Classification with Big Data Technologies

Bin Yu, Les Smith, Mark Threefoot and Femi Olumofin
CTO Office, Infoblox Inc., 3111 Coronado Dr, Santa Clara, California 95054, U.S.A.

Keywords: Behaviour Analysis, Time Series, Big Data Analytics, DNS Security, Data Exfiltration, Anomaly Detection, Classification.

Abstract: Domain Name System (DNS) is ubiquitous in any network. DNS tunnelling is a technique to transfer data, convey messages or conduct TCP activities over DNS protocol that is typically not blocked or watched by security enforcement such as firewalls. As a technique, it can be utilized in many malicious ways which can compromise the security of a network by the activities of data exfiltration, cyber-espionage, and command and control. On the other side, it can also be used by legitimate users. The traditional methods may not be able to distinguish between legitimate and malicious uses even if they can detect the DNS tunnelling activities. We propose a behaviour analysis based method that can not only detect the DNS tunnelling, but also classify the activities in order to catch and block the malicious tunnelling traffic. The proposed method can achieve the scale of real-time detection on fast and large DNS data with the use of big data technologies in offline training and online detection systems.

1 INTRODUCTION

Domain Name System (DNS) that mainly services a domain name resolution to IP addresses on UDP is a service ubiquitous in every network. Because DNS is not intended for data transfer, people can overlook it as a threat for malicious communications or for data exfiltration. Most networks, public or private, do not firewall DNS traffic which creates security vulnerability. Tunnelling data over DNS or TCP over DNS is a technique that can be used as a way to circumvent access and security policies in firewalled networks. A typical example is to illegally browse the web through public hotspot while free service is not provided. There are many free software tools available for people of interest to setup a DNS tunnelling system quickly. One of the most popular tools is Iodine (Iodine). The fact that information bypasses a network first line security mechanism makes DNS tunnelling very attractive also in contexts other than free web browsing. Such examples include command and control and data exfiltration in cyber-espionage attacks in which it is fundamental for an attacker to have an available but inconspicuous communication channel.

DNS tunnelling works by encapsulating data into DNS packets. Typically, the tunnel client encapsulates the data to be sent in a query for a specific domain name. The DNS resolver treats the tunnel traffic as a regular request by starting the lookup process for the requested domain name, possibly recursively consulting other DNS resolvers, as shown in Figure 1. At the end of this operation, the request is processed by the tunnel server. The server retrieves the encapsulated data and responds to DNS queries by enclosing tunnel data in the answer section of the DNS response message.

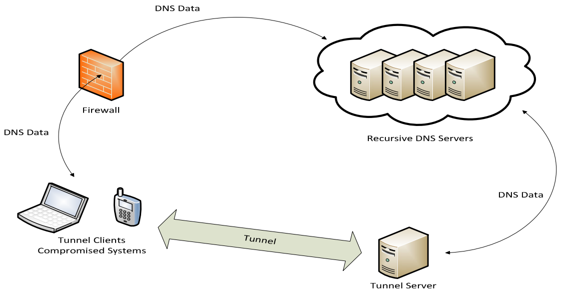


Figure 1: DNS tunnelling setup.

Although most DNS tunnelling techniques use TXT type queries in DNS that can maximize the payload in response packets, there are implementations that make use of DNS query types other than TXT such as A, AAAA, CNAME, NS,

Yu, B., Smith, L., Threefoot, M. and Olumofin, F.
Behavior Analysis based DNS Tunneling Detection and Classification with Big Data Technologies.
DOI: 10.5220/0005795002840290
In *Proceedings of the International Conference on Internet of Things and Big Data (IoTBD 2016)*, pages 284-290
ISBN: 978-989-758-183-0

MX and so on. Our research shows that unlike legitimate users that often use TXT, malicious users tend to use other query types that are more difficult in detection.

DNS tunnelling poses a significant threat and there are methods to detect it. DNS tunnels can be detected by analysing a single DNS payload based on its fundament that the tunnel is used to convey information. However, as a simple technique, DNS tunnels are often used by legitimate users to transfer short messages frequently. Single payload based methods have less latency in detection but cannot make an accurate classification between legitimate and malicious activities. This paper describes a novel approach based on behaviour analysis of DNS traffic. In order to get this approach practical for a real world deployment, it needs to overcome the scalability problems brought up by behaviour analysis or time series modelling. We will explain the big data technologies used in the proposed method in this paper.

2 DNS BASICS

DNS is a critical protocol and service used on the internet. The most common use of DNS is to map domain names to IP addresses. Users can enter a domain name in the web browser. DNS is used to perform a forward lookup to find one or more IP addresses for that domain name. The user's network stack can then send http traffic to the destination IP address. DNS is constantly being enhanced to provide new capabilities.

DNS has over 30 record types with many of the common ones being critical to core internet services. The A record type maps a domain name to an IPv4 address. 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. This 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 (Wong, 2006). The most commonly used types A and AAAA have a population of 85% (Yu, 2014). One of the special DNS types is TXT that is commonly used in tunnelling applications. Its response contains a larger payload of text messages.

DNS uses both UDP server port 53 and TCP server port 53 for communications. Typically, UDP is used, but TCP will be used for zone transfers or with payloads over 512 bytes. There is also the Extension Mechanisms for DNS (EDNS) (Vixie, 1999). If EDNS is supported by both hosts in a DNS communication, then UDP payloads greater than 512 bytes can be used. EDNS is a feature that can be leveraged to improve bandwidth for DNS tunnelling.

DNS is a hierarchical system in which each level can be provided by another server with different ownership. For the internet, there are 13 root DNS servers labelled A through M. These are represented by many more than 13 physical servers. With this hierarchical system, a given domain owner can define the authoritative servers for their domain. This means that they are in control of the ultimate destination host for DNS queries for their domain. In a typical enterprise, endpoints do not make DNS requests directly to the internet. They have internal DNS servers that provide DNS services to an endpoint. However, given that DNS will forward their requests until the authoritative name server is contacted, an attacker with access on an internal endpoint can leverage the enterprise's DNS infrastructure for DNS tunnelling to a domain that they control.

DNS performs caching. When DNS answers are provided a time to live (TTL) is included. The receiving intermediate server can use that value for the amount of time to cache the results. Then if an identical request comes in, the cached result can be provided instead of performing another lookup.

A domain name consists of multiple sections each is referred to be a label with the higher ones on the right side and lower ones on the left side. The whole string is called Fully Qualified Domain Name (FQDN). Removing labels one by one from left to right will yield a Nth level domain name, where N is number of labels for most cases. When N=1, it's called Top Level Domain name (TLD) and N=2, Second Level Domain name (SLD) and so on.

3 RELATED WORK

Some statistical approaches for detecting TCP-over-DNS tunnels have been proposed. Web Tap (Borders, 2004) detects anomalies by looking at HTTP request regularity, inter-request delay, bandwidth usage, and transaction size. In legitimate DNS queries, those attributes have a wide distribution. Crotti et al. approaches the problem from the IP layer by finding inconsistencies in interarrival time, order, and size of the packets (Crotti, 2007, Crotti, 2008, Dusi, 2008). An artificial neural network is used to detect tunnels with features: the domain name, how many packets are sent to a particular domain, the average length of packets to that domain, the average number of distinct characters in the sub-domain labels, and the distance between sub-domain labels (Hind, 2009). Born and Gustafson take the approach of detecting tunnels by analysing unigram, bigram and trigram character frequencies of domains in DNS queries and responses based on some existing tunnelling applications (Bom, 2010). It's possible to be bypassed by newly developed tunnelling tools. Ellens et al. detect tunnels with use of net flow size analysis (Ellens, 2013). While this method can detect tunnels, it doesn't distinguish legitimate and malicious uses of tunnel.

Most traditional methods are based on single payload detection. DNS traffic has very small footprint which makes single payload based detection difficult and even harder to distinguish between legitimate and malicious tunnels.

4 DATA SOURCE

It's important to get real world data for building a high quality detection system. In this project, we utilize DNS data collected from Internet Systems Consortium/Security Information Exchange channel 202 (ISC). It is now spun off as part of Farsight Security (Farsight). This data is collected through its passive DNS technology from more than 100 contributors distributed worldwide. On average, it receives more than 1.8 billion DNS queries per day (It has increased to 17 billion per day when this paper is finished). We have collected more than nine month worth of data for this project. One caveat of using Farsight data is that it doesn't provide end user IP addresses in addition to the DNS server IP due to the sake of privacy protection. Therefore, the detected tunnels are identified by DNS server IP rather than the client IP that is behind the DNS server. Figure 2 shows the percentage of each DNS query type among the data collection. Among various DNS query types, type A and AAAA dominate by 85% while TXT is about 0.8% (Yu, 2014). For tunnelling detection, we will analyse query names for outbound payload and resource records for inbound payload. Except for TXT and ANY query types that contain text resource records, the inbound payloads for all the other query types are extracted from FQDNs.

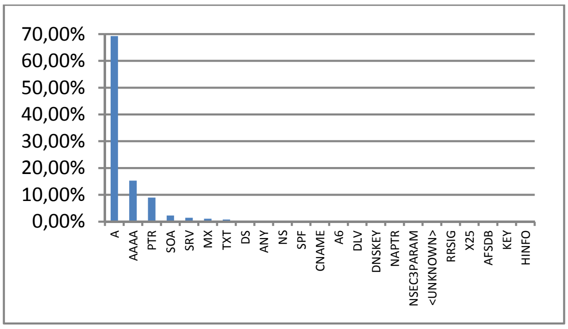


Figure 2: DNS query distribution by query type.

5 FEATURE EXTRACTION AND SELECTION

The legitimate DNS traffic typically has very small payload. That's the reason many approaches detect tunnels based on payload size (Farnham, 2013, Ellens, 2013). However, when space and bandwidth get cheaper, more and more legitimate users are using longer domain names. Since the main objective of the tunnelling technique is to convey information via the tunnel in a way as efficient as possible, the entropy metrics become good features. On the other side, human readability of domain names is a good indicator in tunnel detection (Bom, 2010).

5.1 Effective Payload

There are many types of DNS queries. A tunnel will use query name to carry outbound payloads. The inbound payloads are carried in many different ways depending on the DNS resource record type. For example, in TXT type, the payload is encoded in the text. For many other types, such as A, AAAA, or CNAME, the payload is carried in one or more FQDNs. Unlike legitimate DNS queries that have consistency in query and response, a malicious tunnel tends to change the payload from message to message. An effective payload is a string that is extracted from its original with common prefix, suffix and aligned middle segments removed so that the real signal can stand out.

5.2 Common Features for Inbound and Outbound

Several payload features common for both inbound and outbound traffics are extracted. Figure 3 provides the feature analysis results for inbound provides the feature analysis results for outbound traffic as example where red and green curves are for positive and negative samples, respectively. Their details are described in following sections.

5.2.1 Entropy

According to information theory (Shannon, 1948), entropy is a measurement to quantify the amount of information on a payload. The major objective of a tunnel is to convey as much information as possible over a limited payload size. For single payloads, the entropy features are calculated based on the character distribution of the effective payload.

Given the distribution $D(x)$ of a character set $\{x\}$ within a text string, its entropy is defined as

$$ent = \sum -D(x)\log D(x).$$

A tunnel is assumed to maximize its bandwidth by increasing the entropy of the data being tunnelled.

5.2.2 N-gram Features

In natural English words, the distributions of N-grams are not uniformed that can be used to distinguish them from non-natural English terms. This feature is defined as the value in the P^{th} percentile of the N-gram score distribution $f_N(x|S)$ from a text string S, or

$$nl = \int_{-\infty}^{L_N} f_N(x|S)dx.$$

P can be empirically set to be between 40 to 55. In order to generate N-gram scores, we built a lookup table of N-gram and their frequencies from a set of N-gram English words Google collected from large amount of historical publications (Google). Based on the experiments, we choose to use 2 and 3 grams to have features named *nl2* and *nl3*, respectively.

5.2.3 Lexical Features

In order to pass non text or binary data, a tunnel tends to use some coding methods such as base 64 that introduces many non-human readable characters that can be measured by the lexical features. For a given text string S, the lexical feature is defined as

$$naz = 1 - \frac{|A|}{|S|}, A = \{c \in [a-z], c \in S\}.$$

5.2.4 Payload Size

There are two features for payload size. One is the size of the effective payload *len* and the other is the ratio between effective and original payloads *reo*.

5.2.5 Gini Index

Similar to entropy feature, Gini index is another way to measure impurity of the data that is defined as

$$gni = 1 - \sum D^2(x).$$

However, unlike the entropy feature, Gini index is a feature whose value is bounded within a range between zero and one.

5.2.6 Classification Error

Another feature to measure the diversity of a data set is called classification error. Like the Gini index feature, the value of this feature is also bounded between zero and one. The definition is as follows.

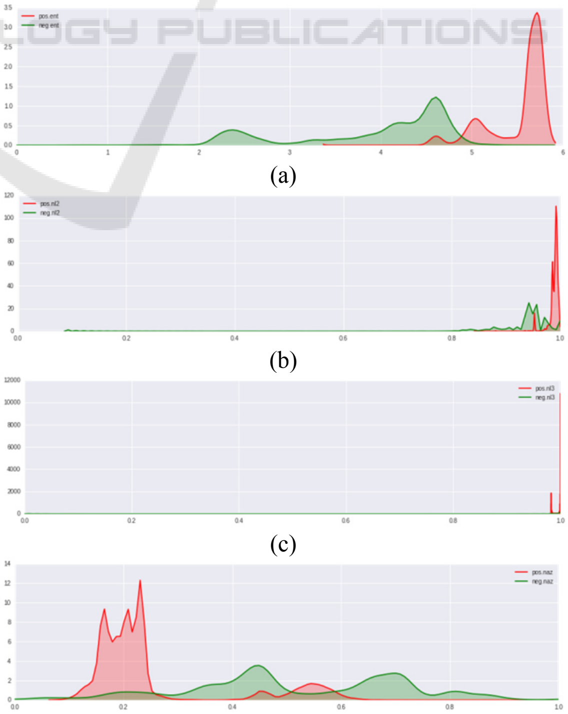
$$cer = 1 - \max\{D(x)\}.$$

5.2.7 Number of Labels

The last but not least feature is the number of domain labels in an FQDN payload named as *nlb* to differentiate legitimate and malicious payloads.

5.2.8 Encoding

The encoding feature *enc* is the output of a neural network that takes all of the above features as input. The classifiers are described in the next section.

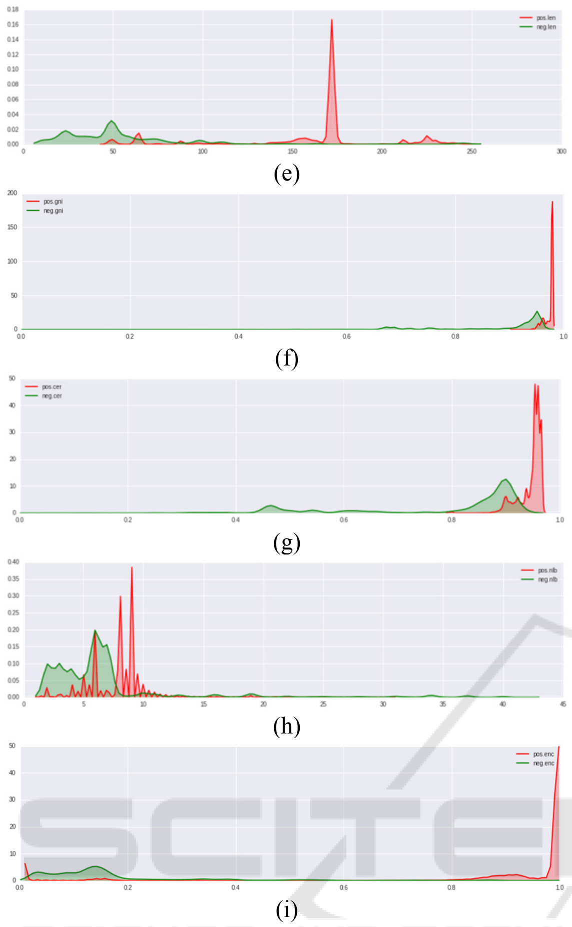


(a)

(b)

(c)

(d)



(e)

(f)

(g)

(h)

(i)

Figure 3: Feature analysis for outbound traffic. (a) Entropy. (b) Bigram. (c) Trigram. (d) Lexical feature. (e) Payload size. (f) Gini index. (g) Classification error. (h) Number of domain labels. (i) Encoding classification.

5.3 Additional Inbound Features

For inbound messages that come from DNS response, the TTL *ttl* of resource records and the response delay *delay* are used as features based on the rational that most of the legitimate DNS queries tend to have longer TTL for reducing number of queries by use of cache. On the other side, tunnelling DNS messages involve extra process such as encoding and decoding, encryption and decryption, proxy and so on. That implies longer response time than normal DNS traffic.

5.4 Time Series Features

The time series data is defined by tunnel ID. Since a tunnel is defined by the requester IP address on one end and the SLD on the other end, the tunnel ID is composed of query IP address and SLD. The requester IP address can be a resolver or DNS server IP address and the internal client IP address combined depending on the information availability. The data points are inserted into an observation cache (Yu, 2014) that has a TTL pre-set to remove old data points from the series. It also has a capacity pre-set for each series to remove old data points when the number of points hit the capacity though they haven't passed the TTL criterion. This is to guarantee the data freshness and reserve the storage space so that it can be recycled. Applying the payload features on to each of the messages within the time series, a feature set that is denoted as a 2-dimensional matrix

$$F = \{f_{i,k}\},$$

can be derived, where $f_{i,k}$ is the k^{th} feature on the i^{th} message for outbound or inbound payloads. The time series based behavior features are the basic statistics of individual features on the series that can be denoted as

$$g_k = \underset{i}{\text{stat}}(f_{i,k})$$

where the $stat$ is the collection $[count, sum, min, max, avg]$ to represent the distribution of individual features across the time series. In addition, the entropy values on effective inbound and outbound payloads are calculated, respectively, because of the fact that the payload of legitimate traffic doesn't change as much as the malicious ones over the time series.

6 CLASSIFICATION

There are two tiers of classification. In the first tier, the classification is targeted on identifying encoded payload while the second tier is for tunnel detection.

6.1 Encoding Classification

Two neural network classifiers are designed and trained to provide a score indicating if a payload is full of encoded text for inbound and outbound payloads, respectively. Each of the classifiers is trained on millions of samples with truth labelled by security experts and tested on independent sets of samples, respectively. The classifiers have a single hidden layer with four neurons and each uses a logistic activation function defined as follows.

$$\frac{1}{1 + e^{-(\sum_{k=1}^{n} f_k w_k + w_0)}}$$

where $f_k$are inputs, $w_k$are weights, and w_0 is the bias for each neuron. To measure the accuracy of the classifier training, the ROC curves are generated on the independent test datasets. Figure 4 is the ROC curve for outbound classifier.

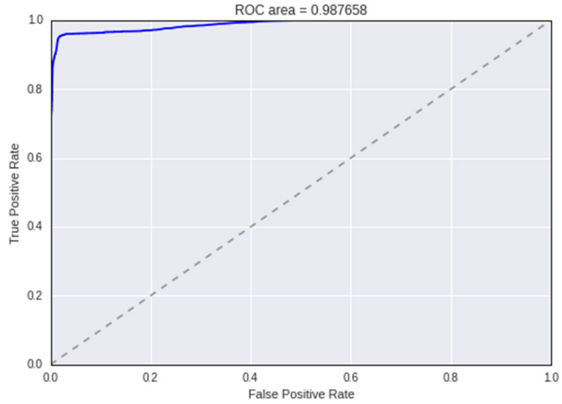


Figure 4: ROC curve for outbound classifier.

6.2 DNS Tunnelling Behaviour Classification

Among various advanced persistent threats, DNS tunnelling is one of the most active and harmful attacks that utilize DNS traffics, therefore, its detection is included in the proposed online detection system. The comprehensive detection workflow is given in Figure 5 where the details of the benign detection and fast flux detection modules are discussed in (Yu, 2014).

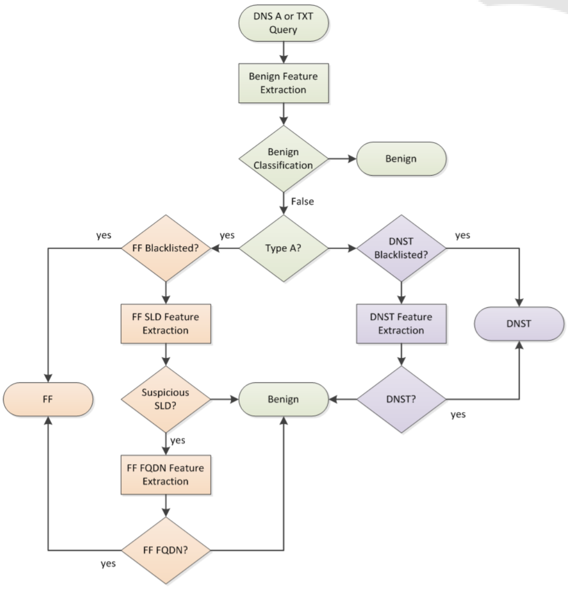


Figure 5: DNS malware online detection workflow.

Data used in this paper is collected from Farsight (Farsight) that receives passive DNS from a large number of contributors worldwide, mainly in the US. With some simple filtering logics such as DNS type, payload size, series length, and whitelisting, a set of candidates is extracted and reviewed by security experts for truth labelling. About 2000 samples are selected for training and testing a tree classifier that is carefully tuned to minimize the false positive rate.

7 REAL-TIME DETECTION SYSTEM

The classifiers that were trained in offline system will be deployed in an online real-time detection system (Yu, 2014) that is designated to deal with fast and large streaming data. In an enterprise deployment, the throughput can be up to 1-3 million DNS queries per second. The throughput can reach billion per second in a cloud based deployment. Therefore, the horizontal scalability is one of the most important factors in design.

As shown in Figure 6, the incoming stream is processed in real-time with Storm or Spark framework and inserted into the observation cache along with the extracted features that are indexed by requester IP address and SLD. The observation cache has an in-memory layer and an on-disk layer of which the use is dependent on the data size. The detection can be triggered by event or scheduled by interval to be cost effective.

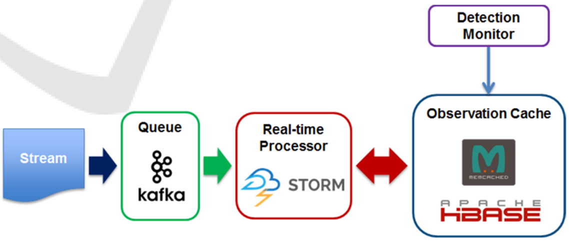


Figure 6: Real-time detection system architecture.

8 RESULTS AND CONCLUSION

Nearly nine month DNS data collected from Farsight from 2012 to 2013 at a rate of 1.8B/day is used in the evaluation process. In total, 126K tunnels are detected where a tunnel is defined as from one unique source IP address to one unique destination domain name. Table 1 shows the summery of the tunnels detected. About 70% detected tunnels are classified into legitimate with review and cross reference check that include many anti-virus companies or CDN vendors. A tunnel with significant payloads in query or response only is categorized into outbound or inbound tunnel and otherwise two-way tunnel. As illustrated in Figure 7, majority falls into one-way tunnel and malicious tunnels have a higher outbound to inbound ratio than legitimate ones due to their data exfiltration nature.

Table 1: Detected tunnels.

	Malicious	Legitimate	All
Two-way	356	869	1225
Outbound	35478	65820	101298
Inbound	2845	20504	23349
Total	38678	87193	125871

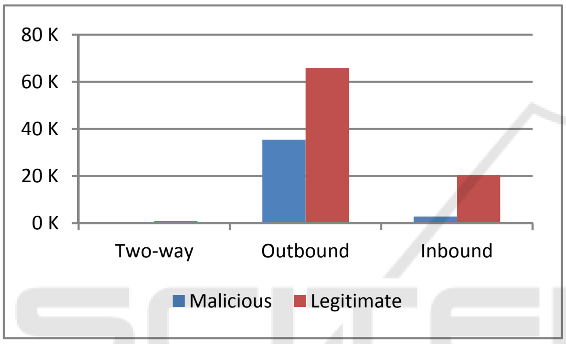


Figure 7: Tunnel distribution.

We also analysed the tunnel transaction activity distribution over DNS query type and the result is plotted in Figure 8, where a tunnel transition is simply a DNS message. It shows malicious activities tend to use type A while legitimate ones have higher transaction rate on type TXT. This is understandable that malicious tunnels want to hide their activities from using TXT type that is designated for large payload transactions and may be exanimated by many traditional DNS tunnelling detection methods.

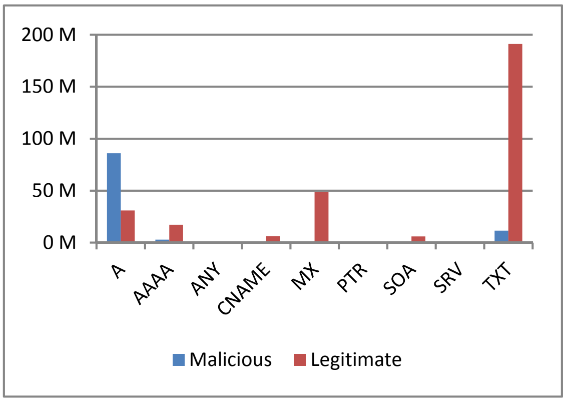


Figure 8: Tunnel transaction activities by DNS query type.

New data shows more and more malicious tunnels are using small payloads. This makes detection even harder and will be a future research work.

REFERENCES

Iodine, http://code.kryo.se/iodine/.

Farnham, G., Atlasis, A., 2013. Detecting DNS tunneling, SANS Institute InfoSec Reading Room.

Wong, M., 2006. Sender policy framework (SPF) for authorizing use of domains in e-mail, version 1, Retrieved from http://tools.ietf.org/html/rfc4408.

Yu, B., Smith, L., Threefoot, M., 2014. Semi-supervised time series modeling for real-time flux domain detection on passive DNS traffic, in *the 10th International Conference on Data Mining and Machine Learning*, St. Petersburg, Russia, pp. 258-271.

Vixie, P., 1999. Extension mechanisms for DNS (EDNS0), Retrieved from http://www.ietf.org/rfc/rfc2 671.txt.

Borders, K., Prakash, A., 2004. Web Tap: detecting covert web traffic, in *Proceedings of the 11th ACM conference on Computer and Communications Security*, New York, pp. 110-120.

Crotti, M., Dusi, M., Gringoli, F., Salgarelli, L., 2007. Detecting HTTP tunnels with statistical mechanisms, in *IEEE International Conference on Communications*, pp. 6162-6168.

Crotti, M., Dusi, M., Gringoli, F., Salgarelli, L., 2008. Detection of encrypted tunnels across network boundaries, in *Proceedings of the 43rd IEEE International Conference on Communications*, Beijing China, pp. 19-23.

Dusi, M., Gringoli, F., Salgarelli, L., 2008. A preliminary look at the privacy of SSH tunnels, in *Proceedings of the 17th IEEE International Conference on Computer Communications and Networks*, St. Thomas, U.S. Virgin Islands.

Hind, J., 2009. Catching DNS tunnels with AI, in *Proceedings of DefCon 17*, Las Vegas, Nevada.

Born, K., Gustafson, D., 2010. Detecting DNS tunnels using character frequency analysis, in *Proceedings of the 9th Annual Security Conference*, Las Vegas, NV.

Ellens, W., Zuraniewski, P., Sperotto, A., Schotanus, H., Mandjes, M., Meeuwissen, E., 2013. Flow-based detection of DNS tunnels, in *Emerging Management Mechanisms for the Future Internet, Lecture Notes in Computer Science*, Volume 7943, pp 124-135.

ISC, ISC Security Information Exchange: http://www.isc.org/

Farsight, Farsight Security, Inc.: https://www.farsightsecurity.com/

Shannon, C., 1948. A Mathematical Theory of Communication, *Bell System Technical Journal*, Vol. 27, pp. 379–423, 623–656.

Google, http://storage.googleapis.com/books/ngrams/books/datasetsv2.html.