International Journal of Recent Trends in Engineering & Research

ISSN(ONLINE) : 2455-1457

IMPACT FACTOR : 3.344

DNS Tunneling Detection

Yakov Bubnov[1]

[1]Department of Electronic Computing Machines,
Belarusian State University of Informatics and Radioelectronics, Minsk, Belarus

Abstract—This paper surveys the problems, related to network traffic analysis to detect anomalies, particularly DNS tunnels. Suspicious connections are revealed with DNS packet analysis. Processing of the collected data performed with algebraic topology techniques.

Keywords—DNS tunneling, data clustering, persistent homology.

I. INTRODUCTION

Security of the computer systems functioning is impossible without ensuring of the proper level of the network infrastructure protection.

The common practice of the intrusion prevention as well as blocking potentially unsafe egress connection implies utilization of the firewalls. Such strategy suppresses direct attacks on host service ports. Nevertheless, one of the numerous methods of penetration into the private networks is establishing the tunnels on the top of protocols on the application layer. Conventional approach preventing undesirable tunneling is in access restriction to incoming connections by blocking certain set of the application layer protocols. Such approach commonly uses pattern matching of a transmitter and receiver addresses. Thus, this is almost futile when dealing with tunnels.

In addition, traffic filtering with access control lists could not be treated as an effective solution. Since, the permitting of the distinct subset of the ports is not always possible. By way of example, the SSH is treated as a crucial protocol of the infrastructure configuration, meaning that generally it could not be entirely disabled. But in this case, network administrators are losing the total control over the ports, available for forwarding through the established SSH connections.

Summarizing, the mechanisms of the application layer tunneling are intelligent enough to circumvent traditional security intrusion prevention systems [1].

DNS is one of the application layer protocols ubiquitously ignored by the firewalls, primarily due to supposed safety. DNS requests to the remote servers are performed either directly with the client applications, or by the recursive requests from the local servers. This protocol specificity might be used to conceal the tunnels.

In the next sections, we describe how to detect the application level tunnels as network traffic anomalies. In the section II, we first discuss the principles of the DNS tunneling detection. In section III we explain the clustering of the DNS traffic images with persistent homology. Finally, section IV concludes this paper.

II. DNS TUNNELING INDICATIONS

The paper of the Farnhman and Atlasis [1] is assembling a plenty of the known approaches for DNS tunnels detecting. According to their research, recognition methods could be divided into two categories: DNS packet analysis and DNS traffic analysis. Packet analysis denotes the request and response payload examination. Traffic analysis denotes the packets study in time to collect statistics – such as count of the packets from a single host, submission frequency, etc.

The following techniques pertain to the DNS packet analysis:

1. Request and response packet size analysis. Krmíček in his paper [2] proposed to detect the tunnels counting of octets in the packet payload. Krmíček concluded that commonly, DNS tunneling tools utilize all available space of the DNS packet, unlike the regular DNS requests.
2. Domain names entropy analysis. Batler, Xu, Yao in paper [3] proposed to use the requesting domain names entropy. Normally, domain names are composed from the natural language words. Therefore, random or encoded domain names have larger entropy. This is the case when arbitrary messages are encapsulated into the domain names. The following formula represents the Shannon entropy:

$$H = \sum_{i=0}^{N} p_i \ln p_i \qquad (1)$$

where p_i – is the frequency of the i-th symbol occurrence in the domain name. However, the content delivery networks technology is an exception of the described approach, since it uses high entropy domain names.
3. Usage of the non-common types of DNS resource records. The end users rarely submit requests with such resource record types as TXT or SRV, while the DNS tunneling tools, such as DNScat, exploit plenty different resource record types.
4. Frequency of the digit occurrences in the domain names. This technique is likewise based on the analysis of the domain names and is proposed by Bilge and Kirda [4].

The following methods relate to the DNS traffic analysis techniques:
1. The DNS traffic volume from a single IP address. The UDP packet payload for the DNS protocol is limited to 512 octets. This force hosts to use an unreasonable amount of packets required to establish a stable tunnel. Besides, continuous polling of the DNS server is an approach of keeping open the DNS tunnel and requires a large amount of packets as well.
2. The DNS traffic volume for certain domains. As Butles pointed out in [3], the DNS tunneling tools commonly are reusing the single domain, so all traffic is forwarding entirely through it. However, the tunnel could be established for multiple domains to reduce the intensity of the requests transmission per domain.
3. The DNS server geographic location. The vast majority of the corporate networks are configuring the local DNS servers. Therefore, requests to the remote DNS servers could be taken into consideration as suspicious and abnormal behavior.
4. Time of the DNS resource records creation. As Brownlee and Wessels noticed [5] that recently created A and NS resource records could be used as an evidence of the established DNS tunnel. Whereas exactly these types of resource records more likely used by the DNS tunneling tools.

III. DNS TUNNELING DETECTION

Approaches discussed in the previous chapter cannot be considered as exhaustive and sufficient. Instead, they might be treated as components of the complex in-depth analysis. Statistical analysis is one of the research directions of the application layer traffic classification, as it is pointed in [5].
In the paper [6] Villamarin-Salomon and Brustoloni proposed to employ Chebyshev inequality and Mahalanobis distance to classify the DNS traffic in the campus networks. Another example is a paper of Wang and Tseng [7], where authors suggested network classification traffic on the basis of covariation of the tunnel attributes vectors.
However, the traditional statistical algorithms are sensitive to the preselected thresholds. Alternatively, topology data analysis algorithms do not experience such limitations [8], particularly, the persistent homology method gives promising results.

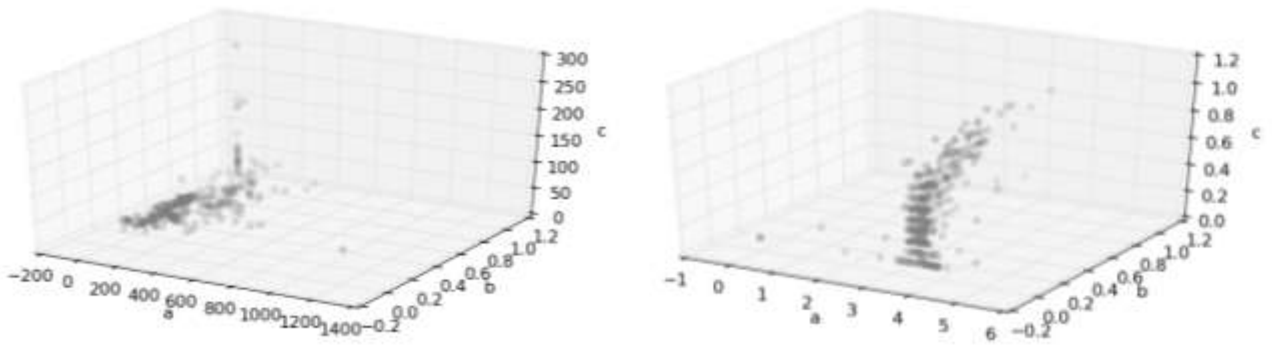
The persistent homology-based method of the data clustering lands on the consolidation of the multidimensional input images (or simplices) into the simplicial complex. The input images are interpreted as zero-dimensional simplices. Zero-dimensional simplex shapes that are glued together generate two-dimensional simplices and so on. Constructed such wise topology is reviewed to reveal the persistence of the topological invariants, or voids. The voids accordingly allow making decision on the separability of the image space.

Further we will review a clustering algorithm with a persistent homology method in action. Experiments have been performed to reveal the DNS tunnels.

Each image presented in the plots below consist of the DNS tunnel attributes described previously:

1. Fig. 1, a contains an average size of the DNS packet payload on percentage of requests without response and percentage of responses without request.
2. Fig. 1, b gives entropy of the requested host names on percentage of the all requests from the single host and on percentage of the total count of different types of resource records.

The classical approach in the adaptation of the persistent homology for data clustering grounds on the representation of images as a set of simplicies $\sigma^n \subseteq K(P)$ in a simplicial complex. In other words, the cluster images are treated as vertices of multidimensional polyhedron $P \subset R^N$ [9].



a) *b)*

Figure 1: DNS traffic collected data.

Homology group *H_n(P)* of the polyhedron *P* is a factor group *Z_n(P) / B_n(P)* of *n*-dimensional cycles over the *n*-dimensional boundaries. It defines a manifold of topology invariants (such as holes and voids). Thus the homology group provides a condition of image space partitioning, a way to distinguish the individual clusters.

The image space is constructed by gluing equal-dimensional simplices together. The simplices are glued based on the distance proximity, while simplex borders are evaluated by sequential discarding of its vertices ν_j [10]:

$$\partial\left(\sigma^{\kappa}\right)=\sum_{i=0}^{p}\left(-1\right)^{i}\left[\nu_0,\ldots,\bar{\nu}_j,\ldots,\nu_{\kappa}\right] \tag{2}$$

For instance, discarding one of the two-dimensional triangle vertex results in two incident vertices, that is a one-dimensional line segment. Nevertheless, stationary topology space does not let us judge of the topology evolution and the general structure of the space [11]. Accordingly, a filtration of the simplicial complex *K* is applied [8]:

$$K_0 \subset K_1 \subset K_2 \subset \ldots \subset K \tag{3}$$

$$K_k = \{\cup\sigma \subseteq K_{k-1} | d(\sigma_i, \sigma_j) \leq \varepsilon, i \neq j\} \tag{4}$$

where $K_0, K_1, ..., K_k$ are simplicial complexes. While gluing together two n-dimensional simplices $\sigma^n{}_i, \sigma_j{}^n$ with the edge $e^n{}_{ij}$ performed if and only if the distance between them does not exceed defined value of ε – that is, the basis of cluster analysis. For each element of the filtration sequence a homology functor is applied:

$$0 \xrightarrow{\partial} H(K_0) \xrightarrow{\partial} H(K_1) \xrightarrow{\partial} H(K_2) \xrightarrow{\partial} \ldots \xrightarrow{\partial} H(K) \tag{5}$$

where, a map $f_{i,j} : H(K_i) \xrightarrow{\partial} H(K_j)$ is injective based on the injectivity of filtration sets $K_i \subset K_j$. That is, the manifolds of intervals where homology groups are persistent define the barcode diagrams [8].
The homology persistence intervals calculation lands on concept of birth and death simplices of the filtration $K^i = \{\sigma^i \vee 0 \leq j \leq i\}$. The birth simplices are called those n-dimensional simplices $\sigma_i{}^n$ of the i-th filtration step, which define a point of a new homology class creation. While the death simplex otherwise specifying the homology class absorption into the border group. Both the birth and death simplexes shape the lower and upper respective boundaries of the homology persistence interval.

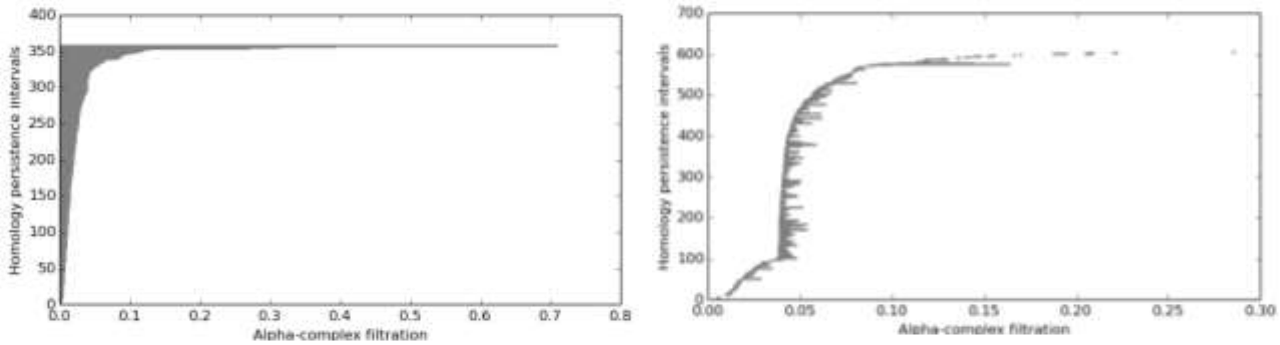
The boundary matrix is used to calculate the intervals of the homology group persistence. The intersection of the j-th column and i-th row of the boundary matrix is set to 1 if $(n\text{-}1)$-dimensional simplex $\sigma_i{}^{n-1}$ is the boundary of the n-dimensional simplex $\sigma_i{}^n$ and 0 otherwise. The intersection of the elements in the boundary matrix is defined for the each n-th step of the simplicial complex filtration.
The reduction operation is defined to calculate the homology persistence interval. The reduction operation denotes a sum of the columns with the same index value of the last non-zero element. These columns should satisfy the following criteria:

$$j_{i+i_0} = j_i + j_{i0}, \; i < i_0 \tag{6}$$

where the i_0 is an index of the processing column. Reduced in a described way matrix will contain the information about birth and death of the homology classes.

We collected source data for the test in the network threat revealing experiment in the local network. Filtration of the simplicial complexes is performed using the Alpha complexes [9]. It is worth noticing that Alpha complexes are based on the Delaunay triangulation.

The result space is constructed of the DNS traffic images. The fig. 2, a. demonstrates the persistent homology groups of the zero-dimensional topology space. Thus plot is characterizing the connected components of the simplicial complex. As it could be seen from the results, some topological invariants are persisted during the long period of the complex filtration, and therefore they could be treated as an image space split into the distinct DNS traffic clusters. The barcode diagram of the one-dimensional simplices is represented in the fig. 2, b. It is also notable for this particular task, that with the increase of the dimension of the homology groups, the persistence intervals are decreasing. Hence, new image clusters are not emerged.



a) *b)*

Figure 2: Results of DNS traffic clustering by persistence homology algorithm.

IV. CONCLUSION

In this paper we have presented a method of the DNS tunneling detection based on the clustering of the DNS traffic images. We have adapted the persistent homology for the data clustering. The most widespread methods of tunneling revelation include statistical analysis instruments. Compared to such methods, presented one have an advantage in revealing the clusters evolution. Future work includes the clustering quality evaluation. We plan to develop the parallel algorithm of the persistent homology computation.

REFERENCES

1. Farnham G., Atlasis A. "Detecting DNS Tunneling", Orlando. SANS, paper, p. 12 , 2013.
2. Krmíček V., "Inspecting DNS Flow Traffic for Purposes of Botnet Detection", Austria. GEANT, paper, p. 5, 2011.
3. Butler P., Xu K., Yao D., "Quantitatively Analyzing Stealthy Communication Channels", Blacksburg. Virginia Tech, paper, p. 14, 2011.
4. Bilge L., Kirda E., Kruegel C., Balduzzi M., "Exposure: finding Malicious Domains Using Passive DNS Analysis", Boston. Northeastern University, paper, p. 5, 2011.
5. Brownlee N., Wessels D., Zdrnia B,. "Passive Monitoring of DNS Anomalies", Auckland. University of Auckland, paper, p. 5, 2007.
6. Villamarin-Salomon R., Brustoloni J. "Identifying Botnets Using Anomaly Detection Techniques Applied to DNS Traffic", Pittsburgh. University of Pitsburgh, paper, p. 2, 2008.
7. Wang Z., Tseng S., "Anomaly detection of domain name system (DNS) query traffic at top level domain servers", Ebene. Academic Journals, paper, p. 2, 2011.
8. Edelsbrunner H., Letscher D., Zomorodian A. "Topological Persistence and Simplification", Urbana. University of Illinois, paper, p. 3, 2002.
9. Edelsbrunner H., Letscher D., Zomorodian A. "Topological Persistence and Simplification", Urbana. University of Illinois, paper, p. 2, 2002.
10. Johnson J., "Topological graph clustering with thin position", Stillwater. Cornell University Library, paper, p. 2, 2012.
11. Hatcher A., "Algebraic topology", Stillwater. Cornell University Library, p. 108, 2001.
12. Carlson, G., "Topology and data", Bulletin of the American mathematical society, vol. 46, n. 2, California. Stanford University Library, p. 256, 2009.