Identification of abnormal DNS traffic via Hurst Parameter

A. Gezer

***Abstract*— It is a necessity for effective network management to be aware of the activities taking place on computer networks. Network managers should always be alarmed about what is happening now, what might be, or what will be in the future for the sake of network. To gather information about a computer system or a network, attackers mostly exploit networking tools to gain some privileges and login systems. Penetration testers also use these tools to gather information about systems, but their main concern is to discover the vulnerabilities of the system, and to find out what kind of measures could be applied to make the system more resistant to these vulnerabilities. In this study, we propose an abnormal DNS traffic identification method via utilizing Hurst parameter estimation. To do so, we employ DNS information gathering tools in Kali Linux to generate abnormal DNS flows. Then, we estimate its self-similarity degree to compare the differences between normal DNS traffic flows and abnormal ones. Obtained results show that abnormal DNS traffic show higher self-similarity degrees. Another interesting finding is that abnormal DNS traffic shows different distribution characteristic.**

Index Terms— **Traffic analysis, DNS protocol, distribution fitting, abnormal traffic detection**

I. INTRODUCTION

NOWADAYS, we need computer network systems more than ever to fulfill our daily routines. The Internet is a global network which we couldn't endure its absence even for a few minutes. The usage of Internet is so widespread that we could perform most of our daily works such as banking, health care, navigation, food order, travel reservations, payment, mailing, telephone calling, and even education; almost all the things which used to need physical presence and effort in the past. Security is a significant issue for every type of computer system and network. We share our most secret credentials to the world through the Internet.

A. GEZER is with Department of Electrical and Computer Engineering, University of Alabama at Birmingham, Birmingham, Alabama USA (e-mail: aligezerr@hotmail.com)

Manuscript received June 21, 2018; accepted July 25, 2018.
DOI: 10.17694/bajece.435230

Although the networking systems and technologies are getting safer, there are always malicious people whose aim are to cause harm to the computer systems, steal significant information, or use our network resources without permission. Attackers always stay vigilant for discovering system vulnerabilities and whenever, wherever possible they will carry out their attacks. For the sake of not only systems but also users, network administrators should also stay vigilant to verify that the network resources and policy are not being violated.

To ensure security and get rid of violations that put the systems in danger, firewalls are mostly used [1]. Although firewalls perform packet inspection, filtering, and prevent any login to systems through the Internet without permission, they don't ensure network system safety for all possible violations and intrusions [2]. In order to further strengthen the network from intruders, intrusion detection and intrusion prevention system usage are on the rise. Intrusion detection systems require monitoring the events occurring in a computer system or network and analyzes them for signs of possible violations, security threats and intrusions [3].

In this study, we focus on the identification of abnormal DNS traffic flows as an intrusion detection system which might be an indicator of an upcoming cyber-attack in advance. DNS packets could be exploited for malicious purposes in many ways. For example, some trojans benefit from a domain name generation algorithm (DGA) to conceal their command and control (C&C) servers [4]. Group activities in DNS traffic might be a signature of a botnet presence. A botnet detection method is proposed by monitoring group activities in DNS traffic [5, 6]. In [7], a DNS protection method is proposed against DNS Spoofing and Poisoning attacks. Attackers could also utilize DNS protocol to obtain information about a local area network before carrying out their attacks vector [8]. They might exploit some DNS tools for gathering information about a targeted domain. In this study, we will simulate this behavior to generate abnormal DNS traffic through exploiting some penetration testing tools. Penetration testing is a simulation of an attack on computer systems to find out the vulnerabilities on the systems and evaluate the security of the system being tested [9]. The basic concept behind this is to find out the vulnerabilities and the weaknesses of systems. Generally, penetration tests include three main steps for pen testing [10]. Sequentially, information gathering, attacking, and reporting. The information gathering phase might consist of

DNS gathering tools, web crawlers, sniffing technologies, spoofing and so on. The idea behind the information gathering phase is to scan and identify system specifications and vulnerabilities and exploit them in attacking. The obtained information about the system is utilized to drive the attack generation phase. Software developers could use this information to eliminate the vulnerabilities and improve the security of their software [9].

We propose an abnormal DNS traffic identification method through the Hurst parameter estimation procedure. The Hurst parameter, which is between 0 and 1, is a numerical measure of self-similarity. It gives information about long-range dependency and probability density characteristic of statistical times series. We generate abnormal DNS flows via exploiting some DNS information gathering tools to discover IP address of domains, domain names, subdomain names, name servers, and mail servers in a targeted domain. To do so, we employed some DNS information gathering tools in Kali Linux. These tools generally use the methods like DNS inquiries to DNS server, zone transfer, brute force, and google searches about a targeted domain. We analyze how these tools work to obtain DNS information on a domain. This process would give us detailed information about particular behaviors of some DNS information gathering tools. We also propose a method to identify abnormal DNS flows via the estimation of self-similarity parameter, namely the Hurst parameter. To compare the differences between normal DNS traffic and abnormal ones, we employ wavelet-based Hurst parameter estimation method. Normal DNS traffic flows are generated via interacting with popular internet domain names for the aim of comparing the self- similarity level of abnormal and normal DNS traffic.

II. BACKGROUND

A. KALI LINUX

Developing tools for hacking and pen-testing purposes consumes a lot of times and is very demanding. Some software like Kali Linux makes it easy to carry out penetration tests. Kali Linux is a Debian based Linux distribution which is popular for penetration testing purposes. It was developed and maintained by Offensive Security [11]. It includes more than 600 tools for penetration testing such as information gathering, vulnerability identification, sniffing and spoofing, attacking, exploitation, forensic investigation tools, etc. It is an open source distribution, which means anyone can modify and enhance it via accessing its source code. Kali Linux can run on many different platforms; even on resource constrained devices such as a raspberry pi. It adheres to the Filesystem Hierarchy Standard. This allows users to easily locate supported files, libraries, and binaries [9].

B. SELF-SIMILARITY

A great variety of physical phenomena could be well represented via self-similar processes [12]. It has found a strong base in many disciplines such as biology, econometrics, natural images, fluctuations of the stock market and turbulence [13, 14]. Self-similar processes have also been considered for modeling network traffic. Previously, network traffic was characterized with Poisson distribution [15]. However, Poisson distribution could not reflect the self-similar nature of network traffic. After modeling Ethernet traffic via self-similar processes, performance-related computations for network traffic such as resource sharing, queue and routing management have been revised [16]. Nowadays, self-similarity artifacts have been considered for the modelling of many broadband network traffic types [16–19].

As self-similarity is so significant in many disciplines, correct estimation of it is a necessity. Powerful properties of wavelet analysis have been an inspiration source for Hurst parameter estimation. Time-scale dependent working nature of wavelet analysis makes it a valuable candidate [20, 21]. An efficient wavelet-based Hurst parameter estimator called Abry-Veitch DWB was proposed in 1998 [22]. Abry-Veitch DWB used Daubechies wavelets as kernel function due to their limited time support which makes it more efficient for handling of border effects. Daubechies wavelets lead more accurate results by better matching self-similar structure of long range dependent processes [23].

The Hurst parameter indicates whether a stochastic process is long range dependent or not. It is a numerical measure of self-similarity. A continuous time stochastic process {X(t), t ϵ R} is strictly self-similar with the Hurst parameter {H, 0<H<1} if

$$X(\text{at}) \overset{d}{=} a^H X(t). \quad (1)$$

X(at) is a new process scaled by factor a and $\overset{d}{=}$ means equal in finite dimensional distributions. When the Hurst value is 0.5, the process is randomly scattered. Let X_k is a discrete time stochastic process, which is defined at discrete time points k=1, 2 …n and $X_k^{(m)}$ is m aggregated time series which is calculated as

$$\left\{X_k^{(m)} = (X_{\text{km}-m+1} + .. + X_{\text{km}})/m, k \geq 1, m = 1,..\right\} \quad (2)$$

$$\text{Var}[X_k^{(m)}] = m^{2H-2}\text{Var}[X_k] \quad (3)$$

$$\rho_z^{(m)} = \rho_z, z \geq 0. \quad (4)$$

If variance and correlation of the time series follows (3) and (4), then the process is called wide sense self-similar or second order self-similar. $\rho_z^{(m)}$ shows autocorrelation coefficient of m aggregated series with respect to lag z.

III. METHODOLOGY

A. BEHAVIORAL ANALYSIS OF DNS TOOLS IN KALI LINUX

Basically, a domain name system (DNS) resolves host names into IP addresses. It is among the core parts of TCP/IP protocol suite. Before downloading packets from a web site, it is

required to resolve the IP address of that web site. There is an ID part in every DNS query. A DNS server replies to a DNS query with the IP address of the URL by putting the same ID which was used in the query [24]. This process between the host and DNS server is open to many attacks because the packet exchange between the two ends occur in plain text. So, any packet sniffer which is located in a strategic position could capture packets and reveal the content easily. DNS information might be used for some malicious purposes such as gathering information about a domain before attacking.

In this section, we test some of DNS information gathering tools in Kali Linux. While carrying out this operation, we also capture the exchanged packets with a protocol analyzer. A protocol analyzer could capture all the packets that pass through the network and monitor the bottlenecks in the network, alert the user to irregular behavior in the network, provide useful information to the network administrator, and reveal packet content which also contains passwords [25].

Generally, packet sniffers are utilized by network managers to troubleshoot the network problems via analyzing passing traffic to find out any problematic traffic such as intrusions, malicious packets, or if there is any user who violates network policy. We utilize a protocol analyzer to capture the traffic and reveal the packet exchange process while executing DNS information gathering commands in Kali Linux. The behavior of the traffic can be compared to an established policy for deviations to further investigate the differences between normal DNS traffic and the traffic from an information gathering tool.

i. Dnsenum

Dnsenum is a tool used for information gathering about name servers, mail servers, and subdomains. The packet traffic is captured via a packet sniffer when we execute the Dnsenum command in Kali Linux and identify what are the exchanged packet contents during this operation. The packet exchange pattern is summarized in Table 1.

Table 1 The packet exchange pattern for Dnsenum

Requests and Replies
IP address of the desired domain
Name server name from the DNS server
IP address of name server
Mail exchange server of that particular domain
IP addresses of the mail servers
Again, IP address of the name server
IPv6 address of name server
Zone transfer is tried between the particular name server and resolver host
Scraping of given domain name through Google Search

ii. Dnsmap

Dnsmap tool is used generally for subdomain scans. A built-in word list is used for scanning subdomain names and IP addresses. According to the built-in word list, IP addresses of domain names are requested from the DNS server. Overall, more than 3000 DNS queries are raised to the DNS server for subdomain IP addresses. At the end of scanning, the number of subdomain names and IP addresses are given as a report on the Kali Linux screen. Most of the DNS queries take DNS error messages.

iii. Dnswalk

Dnswalk is used for zone transfer. According to the captured packets after execution of the Dnswalk command, the packet exchange process could be summarized as in Table 2.

Table 2 The packet exchange process for Dnswalk

Requests and Replies
Request to mark the start of zone transfer
Name server name from the DNS server
IP address of name server from DNS server
IPv6 address of name server from DNS server
Zone transfer is tried for a particular name server
Scraping of given domain name by Google search

iv. Dnsrecon

With using Dnsrecon in Kali Linux, one can reach a lot of information about subdomain names and IP addresses. The usage of the command is also customizable by changing parameters in the command line. With the goo parameter, it gets subdomain names using a Google search. Once the command is executed, it sets up a TCP connection between the host computer and Google servers. After getting subdomain names, DNS queries are raised to ask their IPv4 and IPv6 addresses to the DNS server and finally give them as a report on the Kali Linux terminal screen.

With the -tld parameter, it performs DNS queries according to the typed domain name in command line. Built-in word list is used for second order domain names. All generated DNS queries are requested from the DNS server. Mostly, the returned answers aren't associated with the desired domain name.

v. Fierce

Subdomain names and IP addresses of subdomains could be fetched with the execution of the Fierce command. Fierce usage could be customized with some extra parameters. With the usage of the DNS parameter, at first the name of name server query is raised to the DNS server. Then, IP address of the name server is requested from DNS server. Another query is raised for the IPv6 address of the name server. After obtaining the IP information of the name server of the interested domain, a TCP connection is established between the name server and the host for zone transfer. Mostly, the name server rejects zone transfer. A Brute Force method is attempted to obtain subdomain names and their IP addresses from the name server of the desired domain.

vi. Dmitry

Lots of information could be gathered such as subdomains, e-mail addresses, and port scan with this tool. After executing Dmitry in Kali Linux, we capture the packet traffic with a packet sniffer application. First, the IP address of the desired domain is requested from the DNS server. After getting the IP address, the Ripe.net IP address is requested from the DNS

server. Then, a TCP connection is established between Ripe.net and the host computer to investigate the IP address of the typed domain name. Ripe.net gives IP address ranges for that domain and the information about the organization who owned that IP address. Next, the Whois-servers.net IP address is raised to the DNS server. A TCP connection has been established with the Whois server. Mail addresses, phone numbers, and organization names are obtained from the Whois server. Then the uptime.netcraft.com IP address is requested from the DNS server and a TCP connection is established for searching particular domain addresses. Then a Google-based search is performed after obtaining the IP address from DNS server. Google returns a lot of subdomain names. After getting subdomain names from servers, DNS queries are raised to the DNS server to learn their IP addresses. At the end, a port scan is performed.

B. ABNORMAL DNS TRAFFIC IDENTIFICATION VIA HURST PARAMETER

We employ the aforementioned DNS tools in Kali Linux to gather information about sub-domains, name servers, and mail servers on two domains: www.erciyes.edu.tr and www.mit.edu.tr. Abnormal DNS flow traffics are generated via exploiting DNS information gathering tools while benign traffic captures are generated through network interaction with some popular web sites.

Table 3 Captured DNS Traffic

Datasets	DNS packet counts
Dataset 1 (Normal)	7268
Dataset 2 (Normal)	33522
Dataset 3 (Abnormal)	30344
Dataset 4 (Abnormal)	77519

The content of captured DNS traffic is given in Table 3. Dataset 1 is obtained during the interaction with some well-known web sites in Turkey such as university domains, online newspaper, and some public enterprise websites. Dataset 2 is generated via visiting the Top 215 global URLs in the Alexa top 500 global web sites list. Captured DNS packets during these interactions are counted as normal DNS datasets. Dataset 3 and Dataset 4 are obtained via exploiting mentioned DNS information gathering tools on www.erciyes.edu.tr and www.mit.edu.tr domains, respectively.

In the Hurst estimation procedure, Wavelet-based Hurst estimation method is employed to get the self-similarity degree of packet length and inter-arrival time of DNS traffic. Hurst parameters of the given flows are easily calculated through the relationship between variance of wavelet coefficients and corresponding scale [24] as given in the equation (5)

$$\log_2(\mathrm{var}(d_j[n])) = (2H-1)j + \text{consant.} \quad (5)$$

where $d_j[n]$ represents the wavelet coefficients at j scale, and H represents Hurst parameter. The slope between $\log_2(\mathrm{var}(d_j[n]))$ and j are in relation with 2H-1. Therefore, Hurst parameter could be calculated utilizing this regression line. Each point in Figure 1 and Figure 2 is computed through the variances of wavelet coefficients at the corresponding scale for DNS packet interarrival times and packet length obtained when DNS information gathering tools are employed for www.mit.edu.tr domain.

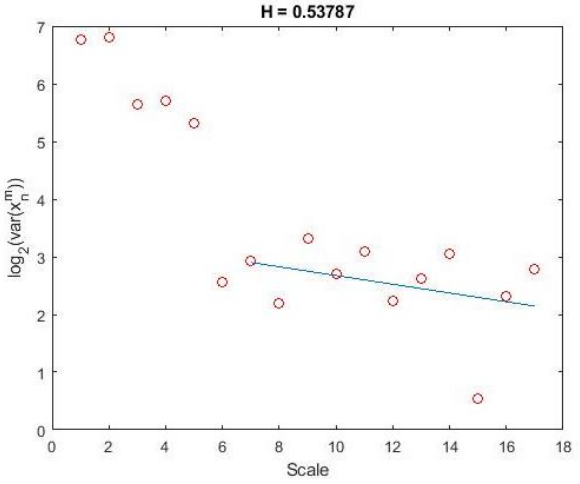

Fig. 1 Wavelet-based self-similarity estimation of inter-arrival times of DNS packets (Utilization of DNS information gathering tools for www.mit.edu.tr domain)

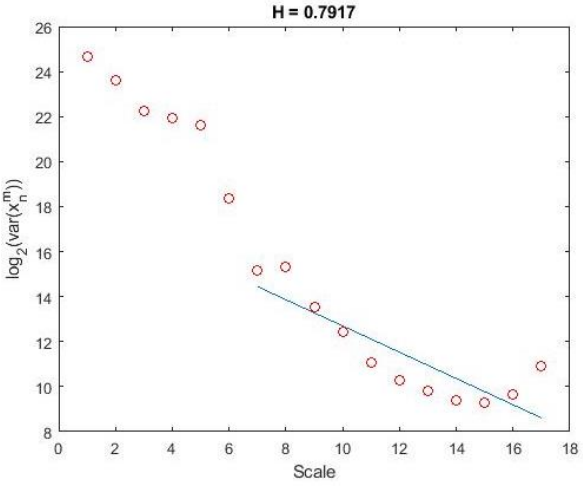

Fig. 2 Wavelet-based self-similarity estimation of DNS packet length (Utilization of DNS information gathering tools for www.mit.edu.tr domain)

Similar figures are obtained during the variance computations of wavelet coefficients at each scale for all Datasets in Table 3. Due to the resemblances in figures, only the associated figures with Dataset 2 and Dataset 4 are shared in the manuscript. Figure 3 and Figure 4 illustrate the variance of wavelet coefficients at corresponding scale for DNS packet interarrival time and packet length computed during the visit of Top the 215 URLs in the Alexa top 500 global web site list.

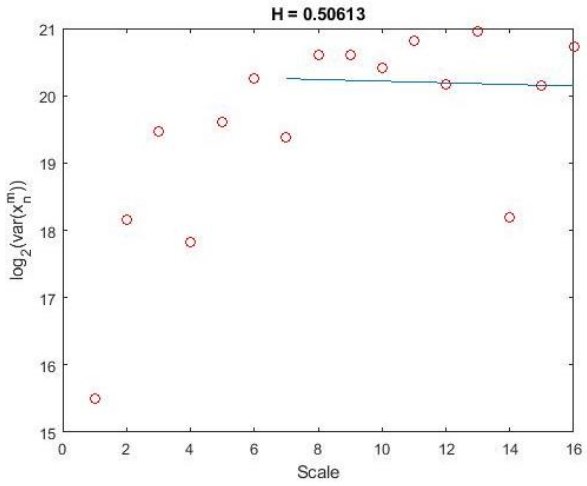

Fig. 3 Wavelet-based self-similarity estimation of inter-arrival times of DNS packets (Top 215 URLs in Alexa top 500 global web sites)

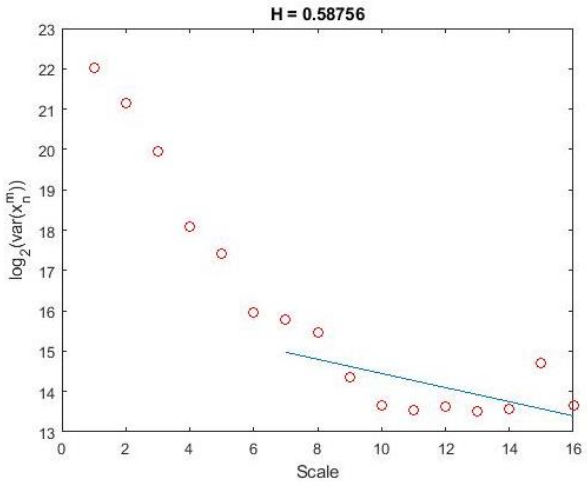

Fig. 4 Wavelet-based self-similarity estimation of DNS packet length (Top 215 URLs in Alexa top 500 global web sites)

The Estimated Hurst values for each data set are given in Table 4 for DNS packet inter-arrival times and DNS packet lengths.

Table 4 Hurst estimation values of each dataset via Wavelet-Based methods

Datasets	DNS packet inter-arrival time	DNS Packet Size
Dataset 1 (Normal)	0.4828	0.4906
Dataset 2 (Normal)	0.5061	0.5876
Dataset 3 (Abnormal)	0.5515	0.8334
Dataset 4 (Abnormal)	0.5379	0.7917

It is observed that abnormal DNS datasets give greater Hurst values for both DNS packet interarrival times and packet lengths. Obtained results demonstrate that Hurst parameters could be an indicator of abnormal DNS traffic, and it might presage an upcoming cyber-attack in advance.

C. DISTRIBUTION ANALYSIS

In this section, our aim is to obtain a fitted distribution function for the modeling of packet sizes of normal and abnormal DNS packet traffic. Distribution functions could be an efficient way of characterizing network traffic due to their non-deterministic behavior. If abnormalities in DNS traffic could be represented with distribution functions, this characteristic could be an indicator of possible cyber-attacks. In the distribution fitting procedure, we test 73 different distribution functions. As selection criteria for goodness of fitting, we employ Kolmogorov Smirnow results and determine the fitted distribution types for packet sizes of normal and abnormal DNS traffic.

Pareto Distribution gives the best result for Dataset 2 which is obtained through the visiting of 215 URLs in the Alexa Top 500 Global web sites. The probability density function of Pareto Distributions is as follows,

$$F_x(\mathrm{x}) = \begin{cases} 1 - (\frac{x_m}{x})^{\propto} & x \geq x_m, \\ 0 & x \leq x_m. \end{cases} \quad (6)$$

Figure 5 shows the cumulative distribution and fitted distribution for Dataset 2. According to the test result, the Pareto distribution parameters α and β are 1.2048 and 61 respectively.

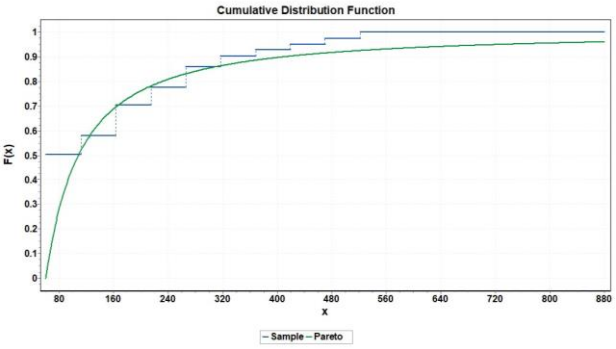

Fig. 5 Packet size distribution of normal DNS traffic (Top 215 URLs in Alexa top 500 global web sites)

Phased Bi-Exponential gives the best result for Dataset 4 which is obtained utilizing DNS information gathering tools for the targeted domain. The probability density function of Phased Bi-Exponential is as follows,

$$F_x(x) = \begin{cases} \lambda_1 e^{-\lambda_1(x-\gamma_1)} & \gamma_1 \leq x \leq \gamma_1, \\ \lambda_2 e^{-\lambda_2(x-\gamma_2)-\lambda_1(\gamma_2-\gamma_1)} & \gamma_2 \leq x \leq +\infty. \end{cases} \quad (7)$$

Figure 6 shows the cumulative distribution and fitted distribution for Dataset 4. According to the test results, the Phased Bi-Exponential distribution parameters Λ_1, γ_1, λ_2, and γ_2 are 0.02468, 66, 4.9635E-5, and 476 respectively.

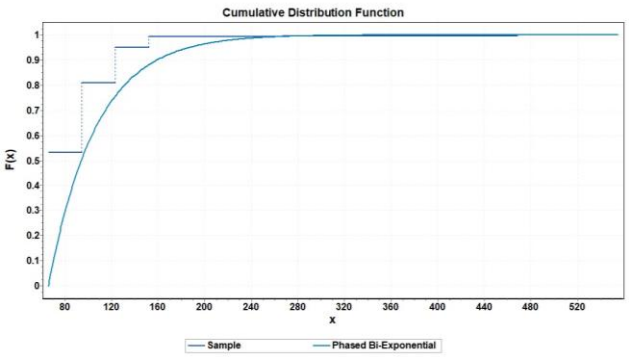

Fig. 6 Packet size distribution of abnormal DNS traffic (Utilization of DNS information gathering tools for www.mit.edu.tr domain)

Obtained results demonstrate that distribution characteristic is a significant feature to identify abnormal DNS traffic.

IV. CONCLUSIONS

Identification of abnormal DNS traffic might presage some cyber-attacks that could take place in the future. We demonstrate that Hurst-parameter estimation might be an indicator of abnormal DNS traffic identification method. To do so, we generate abnormal DNS traffic via utilizing pen testing tools in Kali Linux. We also generate normal DNS packets via interacting with some popular domains in the Alexa top 500 global web sites. Obtained results show that abnormal DNS traffic gives greater Hurst values compared to normal DNS traffic. Therefore, it is understood that Hurst parameter could be an indicator of abnormal DNS traffic. According to Kolmogorov Smirnov test statistics, Pareto distribution could be used to model normal DNS traffic which is obtained during the interaction of the first 215 URLs in the Alexa top 500 global web sites. However, Phased Bi-Exponential gives the best fitting results for abnormal DNS traffic, which is obtained by utilizing some DNS information gathering tools on targeted domains.

ACKNOWLEDGMENT

The authors would like to thank the editors and anonymous reviewers for their valuable suggestions and comments. I would like to thank Dr. Murat M. Tanik for his support, encouragement and guidance.

I also would like to thank Michael Mistretta for helping in the review of the study.

I would like to thank, TUBITAK-Directorate of Science Fellowships and Grant Programmes (BIDEB) to support my postdoctoral research project coded with 1059B191600549 application number.

REFERENCES

[1] H. Chen, J.H. Cho, and S. Hu, "Quantifying the Security Effectiveness of Firewalls and DMZs", In Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security, ACM, 2018.

[2] A. Patel, M. Taghavi, K. Bakhtiyari, and J. Celestino JúNior. "An intrusion detection and prevention system in cloud computing: A systematic review", Journal of network and computer applications, vol. 36, no. 1 , 2013, pp- 25-41.

[3] U.A. Sandhu, S. Haider, S. Naseer, and O. U. Ateeb, "A survey of intrusion detection & Prevention Techniques", 2011 International Conference on Information Communication and Managenent IPCSIT, vol. 16, Singapore, 2011, pp. 66-67.

[4] M. Wielorgorshka, and D. O'Brien, DNS Traffic Analysis for Botnet Detection.

[5] C. Hyunsang, H. Lee, H. Lee, and H. Kim. "Botnet detection by monitoring group activities in DNS traffic", In Computer and Information Technology, 2007. CIT 2007. 7th IEEE International Conference on, 2007, pp. 715-720.

[6] C. Hyunsang, and H. Lee. "Identifying botnets by capturing group activities in DNS traffic", Computer Networks, vol. 56, no. 1, 2012, pp. 20-33.

[7] M.A. Hussain, H. Jin, Z.A. Hussien, Z.A. Abduljabbar, S.H. Abbdal, A. İbrahim, "DNS Protection Against Spoofing and Poisoning Attacks", 3rd International Conference on Information Science and Control Engineering (ICISCE), Beijing China, 2016, pp. 1308-1312.

[8] M. Anagnostopoulos, G. Kambourakis, P. Kopanos, G. Louloudakis, and S. Gritzalis. "DNS amplification attack revisited." Computers & Security 39, 2013, pp. 475-485.

[9] D. Matthew, Z. Carlos, and H. Thaier, "Penetration Testing: Concepts, Attack Methods and Defense Strategies, Systems", Applications and Technology Conference (LISAT), 2016 IEEE Long Island, NY USA, 2016.

[10] W.G.J. Halfound, S.R. Choudrary, and A. Orson, "Penetration Testing with Improved Input Vector Identification" , Software Testing Verification and Validation, 2ICST'09, Denver Co, USA, 2009, pp. 346-355.

[11] Kali Linux by Offensive Security, https://www.kali.org/, accessed September 2017.

[12] S. Giardano, S. Miduri, M. Pagano, F. Russo, S. Tartarelli, "A wavelet-based approach to the estimation of Hurst parameter for self-similar data", International Conference on Digital Signal Processing, DSP 97 2, 1997, pp. 479–482.

[13] M. Barnsley, "Fractals Everywhere", Academic Press, San Dieog, 1998.

[14] J. Beran, "Statistics for Long Memory Processes", Chapman & Hall, New York, 1994.

[15] V. Paxson, S. Floyd, "Wide area traffic: the failure of Poisson modeling", IEEE/ACM Transactions on Networking, vol. 3, no. 3, 1995, pp. 226–244.

[16] W.E. Leland, M.S. Taqqu, W. Willinger, D.V. Wilson, "On the self similar nature of Ethernet traffic (extended version)", IEEE/ACM Transactions on Networking, vol. 2, no. 1, 1994, pp. 1–15.

[17] J. Beran, R. Sherman, M.S. Taqqu, W. Willinger, "Long-range dependence in variable-bit-rate video traffic", IEEE Transactions Communications, vol. 43, no. 234, 1995, pp. 1566–1579.

[18] M.E. Crovella, A. Bestavros, "Self similarity in world wide web traffic: evidence and possible causes", IEEE/ACM Transactions on Networking, vol. 5, no. 6, 1997, pp. 835–846.

[19] D.P. Heyman, T.V. Lakshman, "What are the implications of long-range dependence for VBR-video traffic engineering?", IEEE/ACM Transactions on Networking, vol. 4, no. 3, 1996, pp. 301–317.

[20] E. Masry, "The wavelet transform of stochastic processes with stationary increments and its application to fractional Brownian motion", IEEE Trans. Inform. Theory, Vol. 39, no. 1, 1993, pp. 260-264.

[21] G. Wornell, "A Karhunen Loe've like expansion for 1/f processes via wavelets", IEEE Trans. Inform. Theory, Vol. 36, No. 4, pp. 859-861, 1990.

[22] P. Abry, D. Veitch, "Wavelet Analysis of Long-Range-Dependent Traffic", IEEE Transactions on Information Theory, Vol. 44, No.1, pp. 2-15, 1998.

[23] H. J. Jeongy, D. McNicklez, K. Pawlikowski, "Fast Self-Similar Teletraffic Generation Based on FGN and Wavelets", IEEE International Conference on Networks, Brisbane, Australia, 1999, pp. 75-82.

[24] R. Bassil, R. Hobeica, W. Itani, C. Ghali, A. Kayssi, and A. Chehab, "Security Analysis and Solution for Thwarting Cache Poisoning Attacks in the Domain Name System", Proceedings of the 19th IEEE International Conference on Telecommunications (ICT'12), Lebanon, 2012, pp. 1-6.

[25] A. Pallavi, P. Hemlata, Network Traffic Analysis Using Packet Sniffer, International Journal of Engineering Research and and Applications, Vol. 2, No. 3, 2012, pp. 854-85

BIOGRAPHIES

Ali GEZER was born in Kayseri City, Turkey, in 1976. He received the B.S. degree in Electronic and Computer Education from Marmara University in 1999 and M.S. degree in computer engineering from Erciyes University in 2004, and the Ph.D. degree in electronic engineering from Erciyes University, Kayseri, TURKEY, in 2011.

He is an Assistant Professor with the Electronic and Communication Technology in Erciyes University. Nowadays, He is doing post-doctoral research study at University of Alabama at Birmingham. His research interests include internet traffic, self-similarity, traffic modeling, signal processing techniques, communication technologies, IoT botnet.