Detecting Advanced Persistent Threats using Fractal Dimension based Machine Learning Classification

Sana Siddiqui [1]
siddiqu5@myumanitoba.ca

Muhammad Salman Khan[2]
muhammadsalman.khan@umanitoba.ca

Ken Ferens[3]
ken.ferens@umanitoba.ca

Witold Kinsner[4]
witold.kinsner@umanitoba.ca

[1,2,3,4] Department of Electrical & Computer Engineering, University of Manitoba, Canada

ABSTRACT

Advanced Persistent Threats (APTs) are a new breed of internet based smart threats, which can go undetected with the existing state-of-the-art internet traffic monitoring and protection systems. With the evolution of internet and cloud computing, a new generation of smart APT attacks has also evolved and signature based threat detection systems are proving to be futile and insufficient. One of the essential strategies in detecting APTs is to continuously monitor and analyze various features of a TCP/IP connection, such as the number of transferred packets, the total count of the bytes exchanged, the duration of the TCP/IP connections, and details of the number of packet flows. The current threat detection approaches make extensive use of machine learning algorithms that utilize statistical and behavioral knowledge of the traffic. However, the performance of these algorithms is far from satisfactory in terms of reducing false negatives and false positives simultaneously. Mostly, current algorithms focus on reducing false positives, only. This paper presents a fractal based anomaly classification mechanism, with the goal of reducing both false positives and false negatives, simultaneously. A comparison of the proposed fractal based method with a traditional Euclidean based machine learning algorithm (k-NN) shows that the proposed method significantly outperforms the traditional approach by reducing false positive and false negative rates, simultaneously, while improving the overall classification rates.

General Terms

Cyber security

Keywords

Advanced Persistent Threats (APT); Remote Trojans; Machine learning; Classification; Cyber threats; Complexity; Multifractal

1. INTRODUCTION

The name APT comes from three major attack processes; (I) *Advanced* refers to the usage of innovative and sophisticated procedures to exploit vulnerabilities in a system, (II) *Persistent* refers to a long lasting attack, in which externally controlled command and control (C&C) communications persistently syphon data from the target. As opposed to a simple malware attack in which the intruder avoids detection by minimizing the access time, an APT utilizes lengthy, continuous, and advanced evasion techniques to execute an attack, sometimes over a period of years! (III) *Threat* implies the damages and losses that an attack can cause to the target organization or a human being [**5**] [**8**].

An important technique to detect the presence of an APT, which is employed by most of the state-of-the-art tools, is the analysis of packet data signatures or behaviors based on these signatures or the underlying protocol [**20**]. However, due to the surreptitious nature of the APT, the patterns keep changing, and, therefore, it is difficult to detect them, unless a match is found with the previously known APT repository.

In order to timely detect such attacks, one possible approach would be to identify a distinguishing and unique feature that is associated with APTs, and apply this feature in the training of a machine learning algorithm. The authors have researched and studied various features of APTs i.e. status codes of HTTP traffic [**9**], social engineering techniques which again utilize HTTP mostly [**4**] and firewall rule based fingerprinting [**12**], However, we argue that features based on either HTTP traffic codes or firewall rules are weak in detecting sophisticated APTs which tend to exploit these simple features to obfuscate their presence and still maintain an undetected link with their remote command and control servers. Therefore, a possible distinguishing feature may be a fractal dimension representation of the network layer level command and control communication protocol of APTs. Therefore, the authors have implemented a fractal based machine learning algorithm to detect the presence of APTs using TCP based network connections attributes and compared it with a standard machine learning algorithm.

Using standard instance based training examples in a machine learning algorithm, a data set containing APT and non-APT communications was analyzed and correct classification rates of

IWSPA'16, March 11 2016, New Orleans, LA, USA
© 2016 ACM. ISBN 978-1-4503-4077-9/16/03…$15.00
DOI: http://dx.doi.org/10.1145/2875475.2875484

above 90% were obtained. However, these techniques make use of the statistical information to detect anomalies, and, therefore, they are unable to reduce false positive and false negative rates, simultaneously. This is due to the single/mono scale based Euclidean based error minimization techniques widely used in statistical estimation methods that forms the basis of most of the supervised Machine Learning (ML) algorithms. Single scale analysis does not extract the hidden complexities of an object and is, therefore, limited in detecting deep complex features. Moreover, APT attacks have a tendency to change their patterns, and, in such cases, their detection using an ML approach will tend to fail as they expect the test and training data distributions to be similar. Therefore, Euclidean distance based algorithms, which use mono-scale analysis, are not sufficient enough to extract features; rather, a method that takes multifractal analysis into account is needed. This paper discusses the fractal dimension based algorithm and corresponding results for classification of APTs, in internet traffic data series. The results indicate a substantial reduction in false positive and false negative rates, simultaneously, while improving the overall classification rates.

2. RELATED LITERATURE

Researchers at the University of Twente give a list of necessary features, including aggregated high-speed internet traffic packets, bytes and flow counts, required for the accurate characterization of an anomalous data time series. Their analysis revealed that different metrics are related to various classes of attacks. Therefore, in order to have a global view of the network traffic, all types of metrics are needed to be observed and analyzed simultaneously [**1**]. Another interesting methodology of anomaly classification based on combining data mining and machine learning techniques is described in [**11**]. The scheme makes use of the frequent item-set mining to find common features. After the first step, a decision tree is built to classify benign and malignant sets. The approach is characterized by low false positive rate and is simple in operation. The promising results have led to the successful development and deployment of the anomaly detection and classification system. Authors in [**3**] have presented results for the very accurate attack techniques, almost 89%, and defence technique to reduce these attacks. The attack technique exploits the concept of Gaussian based clustering to identify traffic patterns. After the first step of classification, the Hidden Markov Model (HMM) is used to enhance accuracy by leveraging the link structure of the website. This paper thoroughly discusses the impact of web based caching and cookies on traffic characteristics as well. An alternate method of internet traffic anomaly detection exploited by Berezinski et al. is the entropy measure of the flows. Valuable results in term of anomaly classification have been shown by Tsallis and Renyi entropies using the features of addresses, ports and flows duration [**19**]. In paper [**10**], authors discussed the entropy based approached for anomaly detection using two different feature distributions. One of them is based on flow headers i.e., IP addresses, ports, flow-sizes and other features include behavior of the flows. The time series of the entropy values is used to analyze correlation between feature values and the anomaly. A detailed survey paper about different machine learning based internet traffic classifier is presented in [**23**].

Nevertheless, the use of fractal dimension for supervised and unsupervised learning has recently found a lot of interest among researchers. The paper by D. Barbara and P. Chen [**6**] demonstrated the use of box counting dimension to cluster datasets that scale in terms of size and dimension of the data and have an irregular shape. The stated approach places individual points in the cluster for which the change in fractal dimension is the smallest, exploiting the self-similarity property of the clusters. Another paper by Y. Zavala and et al. [**31**] used the concept of multiplicative binomial cascades to obtain a feature vector for classification. The technique is based on the theory of multifractals and classification rates of over 90% are achieved. Moreover, in [**22**] a combined wavelet and fractal based algorithm for network traffic time series analysis is presented. Discrete stationary wavelet transform is first applied on the time-series representing aggregate statistics of the network and the fractal dimension is calculated for the decomposed signal using variable window length. The reported results indicate potential for further exploration of this field. Further study of [**21**] revealed the use of Hurst parameter, another indicator of self-similarity, to detect anomalies in LAN traffic. The methodology involves the estimation of Hurst parameters for network traffic metrics and their comparison with normal values. The results showed higher accuracy as compared to the variance based time series estimation. Thus, it can be stated that the field of classifying anomalous internet traffic using algorithms based on fractal dimensions appears to be promising in terms of achieving lower values of false positives and false negatives, simultaneously.

3. DATA SET

3.1 Combining Packet Capture (PCAP) Files

The data set used in this study is a combination of the packet capture files obtained from two main sources. First of all, the APTs were collected from Contagio malware database [**15**] contributed by Mila Parkour. The normal and non-malicious data is obtained from PREDICT internet data set repository [**18**] under the category of "DARPA Scalable Network Monitoring (SNM) Program Traffic". The files used from PREDICT data set were filtered to extract normal packet flows. APT data set was combined with these normal flows to generate a combined data set mimicking the mechanism of an APT attack i.e. low and slow. APT attacks are devised so that they do not get prominent in the traffic and remain hidden within normal traffic by spreading the spatial features over larger time scale. Moreover, they as well remain very slow in speed.

The authors have taken guidance on how to synthesize an APT flow with normal packets from a report by researchers at Carnegie Mellon University [**7**]. Also, the authors have investigated the detection of APT attacks in an internet stream where APT establishes command and control with its source [**5**] [**8**]. Our initial experiment in this work is done on TCP based command and control communication, whereas, in general, it could be UDP based as well. Current state of the art technologies utilize pattern matching and signature detection mechanisms to detect APT, therefore, zero-day attacks cannot be detected by these techniques [**16**]. Therefore, the authors have utilized knowledge available in [**7**] and an in-lab experimentation is done to generate APT behavior conforming to the APT features as reported in the literature. Also, the Contagio blog [**15**] and McAfee white paper [**14**] provide various examples of how an APT ensures low and slow activity to evade firewall and intrusion detection technologies e.g. small number of packets in long duration.

An abstract view of the combined capture file is given in Fig 1. Boxes with purple color represents attack traces (19 malicious traces are used from [**15**]) and light blue color boxes are traces of normal traffic. This file was created in such a way that the total number of attack packets is considerably smaller than the normal traffic data. The basic reason for doing this is to mimic the characteristics of an APT which are slower and have smaller share of network traffic (ratio of clean data vs. malware data is very high) and thus makes the detection tougher.

Moreover, since the duration of APTs is often spread over days and months, sophisticated techniques are used to detect their obfuscation. One of these techniques is the use of variation of the same APT like Gh0st RAT which has a number of variants and therefore the dataset comprises of its two variants to ascertain the viability of the proposed algorithm. Moreover, the analysis of APT packet capture file showed that these do not only use port 80/8080 (HTTP/HTTPS) but also use other ports such as 443, 110, 21 and etc. Hence, a diverse set of APTs have been included in the final capture file that was further used to extract features for accurate and reliable classification of anomalous internet traffic.

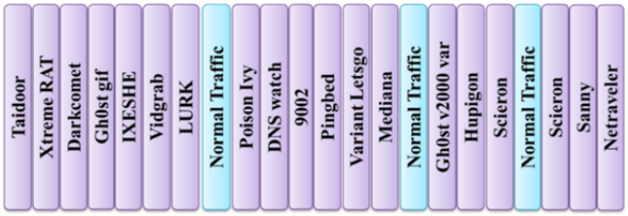

Fig 1: Abstract view of the capture files having mixed anomalous and normal internet traffic

The main reason for selecting a normal data set from the PREDICT DARPA SNM [**18**] is that it has traces from the year 2009 when APT was not a very common term. APT started appearing on the surface of commercial internet technologies in 2010 when the APT threat Stuxnet was discovered [**2**]. Moreover, this PCAP capture of normal traffic contains not-too-old data set which can be utilized to synthesize APT mixed data set with a reasonable valid traffic patterns. In addition, DARPA synthesized this data set to simulate real Internet traffic and has a duration of 10 days from November 3 to November 12, 2009. Total size of DARPA PCAP files is 6 TB and contains HTTP, SMTP and DNS data. In this work, 3 GB of DARPA data set is used from November 3, 2009. For APT traffic, 603.2 MB files of various APTs are used from Contagio [**15**].

3.2 Feature Vector Extraction

The authors have investigated the detection of APTs by performing an in-depth analysis of TCP session data. The feature vector used for the classification comprises of two metrics: one of them being total number of data packets transferred during a single TCP session and the other one was the duration of a complete session. It is reported in [**14**] that APT activity generated a small count of data packets in short lived TCP window/session or small count in large TCP window/session, whereas, normal internet traffic exhibited patterns of large number of packets in short duration. Tshark [**26**], a command line tool extensively utilized for the analysis of PCAP files, was used to extract the required features. The obtained data was then labelled manually (for normal and attack packets) based on the information gathered from the APT files. About 40% of the total data was used for training purpose for the algorithm discussed in the subsequent sections. 25% data was used for cross validation and the rest 35% was used for testing the results.

3.3 Removal of Noise

An important phase of the data retrieval and feature extraction involved removal of the noise, which, in this particular case, consists of two categories of packets as follows:

1) Removal of the TCP packets having zero length, because they do not count towards the actual data packets that were exchanged between host and server.
2) Removal of the re-transmitted data packets, because at the receiver end either the re-transmitted packets are discarded if they were already received or they are kept in case the original packets were not received. In essence, the total count of the actual data packets remains the same.

Once this pre-processing phase is completed, the data is ready to be fed to the classification engine. The synthesis of data set according to Fig 1 is performed using the data set information available in Table 1.

S. No.	Traffic Name	Total Packets	Total TCP Sessions	Total TCP Duration
1	Taidoor	116	63	14.9623
2	Xtreme Rat	2578	8	523.5481
3	Dark Comet	2	1	0.1249
4	Gh0st-gif	3	3	0
5	IXESHE	3	3	0
6	Vidgrab	245	1	0
7	LURK	210	42	1156.4667
8	**Normal Traffic**	658677	12796	22716.4547
9	Poison Ivy	89	1	2888.3428
10	DNS Watch	247	2	1202.648
11	9002	3585	1	369.6251
12	Pingbed	19	19	363.6382
13	Variant Letsgo	277	1	0
14	Mediana	266	133	12.1138
15	**Normal Traffic**	662219	12812	41.466
16	Ghost v2000 var	26	25	2013.0399
17	Hupigon	66	33	.2886
18	Scieron	60	3	0.0093
19	**Normal Traffic**	667195	12372	69.9476
20	Scieron	64	3	1554.7916
21	Sanny	531	8	124.8862
22	Netraveler	68	28	47.7137

Table 1: Information of data set after removing noise.

There are total 38,358 such sessions for each set of source and IP destination containing both normal and APT data set. Out of these session, APT session count is 378 while 37,980 sessions belong to normal traffic. As shown in Fig 2 and Fig 3, a distribution of these sessions is plotted where each point on the plots represents total data packets exchanged during a TCP session for both normal and APT traffic respectively. Analyzing Fig 2 and Fig 3, it is evident that predominantly APT samples lie in the region of normal traffic samples. This is done so that APT can masquerade the characteristics of legitimate TCP sessions. APT opens and closes TCP connection with the command and control server with minimal data transfer to ensure that it remains low while normal connections complete their packet exchange in single connection typically.

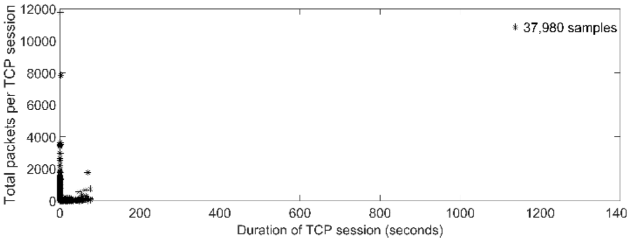

Fig 2: Total packets vs. Duration per TCP session for Normal Traffic (37980 samples)

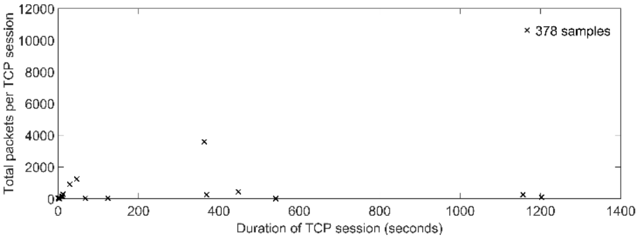

Fig 3: Total packets vs. Duration per TCP session for APT Traffic (378 samples)

4. ANOMALY CLASSIFICATION ALGORITHMS

Two approaches have been employed to test the data set and a comparison of results follows this section. The first approach uses traditional supervised learning algorithms from the class of Instance Based Learner i.e. k-Nearest Neighbours and the other approach is based on exploiting the multifractal nature of the internet data series using the Correlation based Fractal dimension. In the following sections, the malicious/attack class is referred to as the positive class and the negative class is the benign/normal data class.

4.1 k- Nearest Neighbors (k-NN)

k-nearest neighbours is a variation of the instance based learner algorithm, which uses training examples, also known as instances. The process of learning involves the addition of new examples. The aim is to classify an unknown instance based on the similarity index, often a distance, with respect to the training instances. A predicted class label is then associated with the new example. The variation of k in k-NN represents the number of neighbor instances that needs to be compared, and the result of the votes of the majority class elements is considered the class label. Mathematically, it can be expressed as:

$$\widehat{y(x)} = y_{n^*} \qquad (1)$$

$$n^* = \arg\min \mathrm{dist}(x, x_n) \qquad (2)$$

The k-Nearest Neighbours algorithm is a non-parametric classification method that is sensitive to the local geometrical structure of the data. The major drawback of this algorithm is that it is slow in determining nearest neighbours for high dimensionality data. Moreover, it is essential to save all the training data therefore memory management is critical in case of large datasets [**13**]. Also, k-NN works best when the data set has low variance and low bias [**24**]. All the above cited problems in a traditional k-NN algorithm are addressed with modifications i.e. high dimensional k-NN in [**17**] and time and memory scalability is addressed in [**25**] and [**30**]. However, in this paper, the authors investigate only the detection performance of traditional k-NN and the proposed algorithm. Based on the hypothesis that an APT threat will disguise its characteristics to that of a normal traffic, traditional k-NN is chosen to validate the proposed algorithm.

4.2 Correlation Fractal Dimension Based Algorithm

4.2.1 Background

The basic requirement of the "correlation fractal dimension based algorithm" is a reference dataset of the features which is accurately labelled as well. Each new data point is classified as anomalous or benign by comparing the correlation fractal dimensions of the corresponding dataset. The algorithm first calculates the correlation fractal dimension of the attack and normal reference datasets, separately, and forms a prototypical measure for each class. To classify new input samples, the methodology computes the correlation fractal dimension of the new samples with reference data set and compares them to the prototypical measures of the normal and attack data sets. The class for which there is a minimal change in the fractal dimension indicates that the point belongs to that particular class. This can also be regarded as finding the similarity index of the new sample with each class and choosing the class to which the input is most similar [**29**].

The authors have chosen the correlation dimension, D_c for computation of the similarity index. To compute the correlation dimension, first fix a frame of reference in which all the samples exist. Then, cover it by N_r volume elements (vels), such that each element has a resolution of scaling factor $\boldsymbol{r}$. Assuming that the jth vel is intersected by the fractal with a frequency n_j, the probability of the jth vel can be defined as follows [**28**] :

$$p_j = \lim_{N_T \to \infty} \frac{n_j}{N_T} \qquad (3)$$

$$N_T = \sum_{j=1}^{N_r} n_j \qquad (4)$$

If the following power law relationship holds between the sum of squared probabilities over all the vels with scaling diameter **r** [**27**]:

$$\left(\sum_{j=1}^{N_r} p_j^{\,2}\right)^{-1} \sim \left(\frac{1}{r}\right)^{D_c} \qquad (5)$$

Then, the correlation dimension is given by the following relation [27] [28] [29]:

$$D_c := \lim_{r\to 0} \frac{-\log \sum_{j=1}^{N_r}\left(p_j^{\,2}\right)}{\log\left(\frac{1}{r}\right)} \qquad (6)$$

It is important to note here that the correlation fractal dimension is estimated with finite values of the scaling factor r. However, it is ensured that the value of scaling factor is at least chosen 5 in order to estimate a linear log-log relationship [27]. Similarly probability of the jth vel in equation (3) is computed accordingly.

4.2.2 Pseudo Code

```
1) Initialize a labelled reference dataset, R, comprising of at least 30-40% of the total data to be classified.
2) Compute the prior Correlation Fractal Dimension, fd_prior_anom, of only anomalous data samples in R.
3) Compute the Correlation Fractal Dimension, fd_prior_norm, of only normal data samples in R.
4) Load a set S of data points in the main memory.
5) For each point or sample p in S:
    a) Re-calculate the Correlation Fractal Dimension, fd_posterior_anom, by adding p to anomalous values of R. Compute the change in Correlation Fractal Dimension, fd_anom = abs(fd_posterior_anom - fd_prior_anom).
    b) Re-calculate the Correlation Fractal Dimension, fd_posterior_norm, by adding p to normal values of R. Compute the change in Correlation Fractal Dimension, fd_norm = abs(fd_posterior_norm - fd_prior_norm).
    c) If, fd_anom < fd_norm,
           classify p as anomalous.
    d) If, fd_norm ≤ fd_anom,
           classify p as normal.
```

5. RESULTS

The results in terms of true and false detection rates using k=3 neighbours for k-NN and Correlation Fractal dimension based algorithm is given in Table 2.

Algorithm	True Positive Rate	True Negative Rate	False Positive Rate	False Negative Rate
k-NN	92.83 %	93.66 %	6.34 %	7.16 %
Correlation Fractal Dimension	93.58 %	94.43 %	5.57 %	6.41 %

Table 2: Classification performance results of kNN

5.1 Classification Performance Comparison of both algorithms

The comparison of the classification results shows noticeable performance improvement using the fractal dimension based algorithms, as evident from Table 3. As shown, Accuracy, Sensitivity and Specify have slight improvement but Precision and F-measure have shown more than 12% improvement.

S. No.	Metric	k-NN	Correlation Fractal Dimension	% Improvement
1	Accuracy	93.65%	94.42%	0.82%
2	Sensitivity	92.83%	93.58%	0.81%
3	Specificity	93.66%	94.43%	0.82%
4	Precision	13.35%	15.02%	12.51%
5	F-measure	26.66%	29.99%	12.49%

Table 3: Algorithm classification performance comparison chart

F-measure which is an indicator of how accurate is the algorithm in learning the positive class in an imbalanced data set (low ratio of positive samples vs. negative samples), has shown larger improvement in our algorithm. It is important to mention that fractal dimension based algorithm performs better than simple Euclidean dimension based algorithm primarily because of its capability to extract multiscale measurement which is better than mono scale Euclidean measure. Multifractal analysis extracts hidden information about a measure that is not possible in a mono scale analysis i.e. Euclidean measures. Moreover, our algorithm also shows that the TCP based connections show multifractality that makes it possible to estimate fractal dimensions.

6. CONCLUSIONS

This paper is focused on the classification of APT based anomalous traffic patterns with high accuracy and reliability using the feature vector obtained through the processing of TCP/IP session information. A new correlation fractal dimension based algorithm is proposed and results have been compared with one of the traditional machine learning algorithms i.e. k-NN. The obtained results have shown that fractal method provides better performance in terms of reducing both false positives and false negative. Also, the proposed algorithms performs better than k-NN with highly imbalanced data set. Future work includes testing of the data series using other machine learning algorithms like Naive Bayes, Support Vector machines, Decision Trees and Boosted classifiers to validate the performance of the proposed methodology. Moreover, using more feature vector, other fractal dimensions like Self-Similarity dimension, Hausdorff dimension, Information dimension and variance fractal dimension can be explored for classification of advanced malicious data.

7. ACKNOWLEDGMENTS

This work is supported in part through a research fellowship from Mitacs-Accelerate Canada. Authors are also thankful to PREDICT USA for providing state-of-the-art data sets. Authors are also thankful to Mila Parkour of Contagio malware blog for providing traces of APT attacks.

8. REFERENCES

[1] Anna Sperotto, Ramin Sadre, and Aiko Pras, "Anomaly Characterization in Flow-Based Traffic Time Series," in *Lecture Notes in Computer Science, IP Operations and Management*, vol. 5275, 2008, pp. 15-27.

[2] Beth E. Binde, Russ McRee, and Terrence J. O'Connor, "Assessing Outbound Traffic to Uncover Advanced Persistent Threat - Joint Written Project," SANS Technology Institute, 2011. [Online]. http://www.sans.edu/student-files/projects/JWP-Binde-McRee-OConnor.pdf

[3] Brad Miller, Ling Huang, A. D. Joseph, and J. D. Tygar, "I Know Why You Went to the Clinic: Risks and Realization of HTTPS Traffic Analysis," vol. 8555, pp. 143-163, 2014.

[4] Colin Tankard, "Advanced Persistent threats and how to monitor and deter them," *Network Security*, vol. 2011, no. 8, pp. 16–19, August 2011.

[5] Damballa Inc. (2010) Advanced Persistent Threat (APT).

[6] Daniel Barbara and Ping Chen, "Using the fractal dimension to cluster datasets," in *Proceedings of International conference on Knowledge discovery and data mining*, 2000, pp. 260-264.

[7] Deana Shick and Angela Horneman , "Investigating Advanced Persistent Threat 1 (APT1)," CERT Division, Software Engineering Institute, Carnegie Mellon University, USA, 2014.

[8] Eric M. Hutchins, Michael J. Clopperty, and Rohan M. Amin, "Intelligence-Driven Computer Network Defence Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains," in *6th Annual International Conference on Information Warfare and Security*, Washington, DC, USA, 2011.

[9] Fatima Barcelo-Rico, Anna I. Esparcia-Alcazar, and Antonio Villalon-Huerta, "Semi-Supervised Classification System for the Detection of Advanced Persistent Threats," *Recent Advances in Computational Intelligence in Defense and Security*, pp. 225-248, December 2015.

[10] George Nychis, Vyas Sekar, David G. Andersen, Hyong Kim, and Hui Zhang, "An Empirical Evaluation of Entropy-based Traffic Anomaly Detection," in *Internet Measurement Conference*, Greece, 2008.

[11] Ignasi Paredes-Oliva, Ismael Castell-Uroz, Pere Barlet-Ros, Xenofontas A. Dimitropoulos, and Josep Sole-Pareta , "Practical anomaly detection based on classifying frequent traffic patterns," in *IEEE conference on Computer Communications Workshops*, 2012, pp. 49-54.

[12] Ivo Friedberg, Florian Skopik, Giuseppe Settanni, and Roman Fiedler, "Combating advanced persistent threats: From network event correlation to incident detection," *Computers & Security*, vol. 48, pp. 35–57, February 2015.

[13] J. Zico Kolter and Marcus A. Maloof, "Learning to detect and classify malicious executables in the Wild," *Journal of Machine Learning Research*, vol. 7, pp. 2721-2744, 2006.

[14] McAfee Inc., "Combating Advanced Persistent Threats- How to prevent, detect, and remediate APTs," 2011.

[15] Mila Parkour. (2013) Contagio malware database. [Online]. https://www.mediafire.com/folder/c2az029ch6cke/TRAFFIC_PATTERNS_COLLECTION#734479hwy1b97

[16] Nart Villeneuve and James Bennett, "Detecting APT Activity with Network Traffic Analysis," Trend Micro Incorporated Research Paper, 2012.

[17] Nenad Tomasev and Krisztian Buza, "Hubness-aware kNN classification of high-dimensional data in presence of label noise," *Neurocomputing*, vol. 160, pp. 157–172, February 2015.

[18] PREDICT. (2009) DARPA Scalable Network Monitoring (SNM) Program Traffic.

[19] Przemysław Berezinski, Jozef Pawelec, Marek Małowidzki, and Rafał Piotrowski, "Entropy-Based Internet Traffic Anomaly Detection: A case study," in *Processings of 9th International Conference on Dependability and Complex Systems, Advances in Intelligent Systems and Computing*, vol. 268, Brunow, Poland, 2014, pp. 47-58.

[20] Ross Brewer, "Advanced persistent threats: minimising the damage," *Network Security*, vol. 2014, no. 4, 2014.

[21] Ruoyu Yan and Yingfeng Wang, "Hurst Parameter for Security Evaluation of LAN Traffic," *Information Technology Journal*, vol. 11, no. 2, 2012.

[22] Seyed Mahmoud Anisheh and Hamid Hassanpour, "Designing an Approach for Network Traffic Anomaly Detection," *International Journal of Computer Applications*, vol. 37, no. 3, 2012.

[23] Thuy T.T. Nguyen and Grenville Armitage, "A Survey of Techniques for Internet Traffic Classification using Machine Learning," vol. 10, no. 4, pp. 56-76, 2007.

[24] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning - Data Mining, Inference, and Prediction*, 2nd ed.: Springer, 2013.

[25] Ugur Demiryurek , Farnoush Banaei-Kashani, and Cyrus Shahabi, "Efficient k-nearest neighbor search in time-dependent spatial networks," in *21st international conference on Database and expert systems applications: Part I*, Bilbao, Spain, 2010.

[26] Wireshark. (2015) https://www.wireshark.org/docs/man-pages/tshark.html.

[27] Witold Kinsner, Graduate lectures on Fractal and Chaos Engineering, 2015.

[28] Witold Kinsner , "It's time for multiscale analysis and synthesis in cognitive systems," in *IEEE 10th Intl. Conf. Cognitive Informatics & Cognitive Computing (ICCI*CC11)*, Banff, AB, 2011, pp. 7-10.

[29] Witold Kinsner , "System Complexity and Its Measures: How Complex Is Complex," in *Advances in Cognitive Informatics and Cognitive Computing Studies in Computational Intelligence*, Yingxu Wang , Du Zhang, and Witold Kinsner, Eds.: Springer Berlin Heidelberg, 2010, vol. 323, pp. 265-295.

[30] Youngki Park, Sungchan Park, Sang-goo Lee, and Woosung Jung, "Greedy Filtering: A Scalable Algorithm for K-Nearest Neighbor Graph Construction," in *19th International Conference Database Systems for Advanced Applications-Part I*, vol. 8421, Bali, Indonesia, 2014, pp. 327-341.

[31] Yulios Zavala, Jeferson Wilian de Godoy Stênico, and Lee Luan Ling, "Internet Traffic Classification Using Multifractal Analysis Approach," vol. 3, no. 8, pp. 388-394, 2013.