A Machine Learning Approach to Predict Advanced Malware

Mehmet Barış Yaman, Can Yılmaz Altınığne, and Şerif Bahtiyar*
Department of Computer Engineering, Istanbul Technical University
Maslak, Istanbul, 34469, Turkey
{yamanmeh, altinignec, bahtiyars}@itu.edu.tr

***Abstract*—The pervasive usage of communication systems has created many different networks with huge connectivity options for entities and services on the Internet. The options have been powered with new computing technologies that have advanced existing malware and have created new malware more than ever. The new malware has extremely different properties and it uses many stealthy methods to hide its traces during an attack. Therefore, it is impossible to detect the new malware, which we called advanced malware, with existing anti-malware systems and prevent the attacks accordingly. In this paper, we extract distinctive features of advanced malware to predict the type of malware. We present a novel machine learning approach to predict the type of malware. We also experimentally analyze malware to show correlations among features that may be used to predict advanced malware. The analyses results show that correlations among the distinguishing properties will be used to predict the type of malware.**

***Index Terms*—Advanced Malware, Machine Learning, API Call, Prediction, Classification**

I. Introduction

Advanced malware is a new type of complex malicious software having very advanced properties, which has become apparent in the wild for few decades. The main purpose of advanced malware is to use them for targeted attacks with high success ratio. Particularly, critical systems are main targets of advanced malware, where many different attack vectors are used to accomplish the attacks [1]. Additionally, conventional intrusion detection systems and anti-malware systems are unable to detect advanced malware since it has exceptionally complex structure. Therefore, critical systems have suffered from advanced malware considerably. For instance, financial systems and critical infrastructures are some of the targets, where there attacks with advanced malware have seen [2], [3].

Recently, malware contribute to some stages of many complex targeted attacks. Anti-malware systems and intrusion detection systems detect some of the attacks and corresponding malware. Actually, attackers use conventional malware with the complex attacks, which attacks are able to be detected with some existing mechanisms. In this paper, we distinguish malware in two categories, namely conventional malware and advanced malware as in [4]. Conventional malware is malicious software that are already categorized in literature, such as virus, worm, and etc. [5]. Moreover, this type of malware is almost always detectable with adequate anti-malware systems [6]. On the other hand, advanced malware has been undetectable until the attack is completed with existing anti-malware systems and intrusion detection [4].

The grand challenge is to detect advanced malware before it completes its task, such as detecting advanced malware before successful targeted attacks. Therefore, new detection mechanisms are needed for advanced malware. In this paper, machine learning algorithms are used to extract information for detecting advanced malware based on features of conventional malware and advanced malware instances seen in the wild. The paper contains two main contributions. The first one is to determine distinguishing properties of advanced malware that may be used with machine learning algorithms. The second contribution is to determine correlations between features of conventional malware and advanced malware. We expect that the correlations may be used to predict the type of malware and prevent attacks with advanced malware.

The rest of the paper is organized as follows. Section II contains the evaluation of malware. Next section is devoted to a model to predict advanced malware. Section IV is about analysis of malware with machine learning algorithms. We conclude the paper in the last section.

II. The Evaluation of Malware

In this section, we categorize malware according to its properties for extracting distinguishing features of advanced malware, which may be used with machine learning mechanisms for detection purposes. Table I contains a comparison of malware categories.

A. Conventional Malware

Malware is a malicious software used to deliberately harm computer systems, harvest critical data and system resources, manipulate network transactions and access private information of individuals. Worm, virus, Trojan horse, spyware, botnet, and rootkit are instances of conventional malware.

Viruses are malicious software that can replicate themselves whenever they are active. Virus needs a host to survive so it is primitive malicious software. Therefore, virus may have a simple goal. For example, virus infected software may give rise errors to the system [5]. On the other hand, worm is a stand-alone malicious software [7].

Trojan horse either opens doors to other malware or steals information from infected host. It provides remote access to

*Corresponding author.

TABLE I
COMPARISON OF MALWARE CATEGORIES

Properties	Advanced Malware	Combo Malware	Conventional Malware
Stealth	Use of stolen signature and others	Use methods of conventional malware	Depends on the type of malware
Creation	Codes from existing and unknown malware	Borrowing codes from existing malware	Generating code from known malware
Size	Generally bigger size	Sum of components size	Smaller size
Propagation	Many methods such as fragmentation	Conventional malware components	Depends on the type of malware

the infected systems. Unlike worm and virus, Trojan horse does not replicate itself [7]. There is many other malicious software that is used for espionage. Key-loggers monitor information from the system by recording keystrokes on the infected machine once the system is infected. Spyware affects the system or machine to monitor a wide range of critical information. Backdoors or trapdoors may settle as a part of a system and gain access for malware owner to pass authentication controls.

Rootkit inserts a set of software codes to the targeted system for gaining administrators privileges for remote control while hiding its existence. Similarly, a botnet has remote control facility. Botnets are remotely controlled computer network systems. These networks may be used for different purposes, such as spam e-mails and denial of service attacks [8]. All these malware types are detectable with some anti-malware mechanisms that are already running in the wild.

B. Combo Malware

Combo malware is a combination of many conventional malware. For instance, combo malware may be combinations of virus and worm. This type of malware is created by borrowing code from existing conventional malware [9]. For example, Lion and Bugbear.B malware may be categorized as combo malware. Lion malware is composed of Linux worm and rootkit. On the other hand, Bugbear.B is a combination of worm, virus, and backdoor [9]. Thus, their payload, size, and propagation depend on the components accordingly.

C. Advanced Malware

Advanced malware is sophisticated malicious software that has exceptionally different structure than conventional malware. The major properties of advanced malware are complexity, goal orientation, modular, stealth, being written in multi-languages, use of cryptography, and use of multiple vulnerabilities [4]. Advanced malware has dynamic nature therefore its components, infection mechanisms, and payload

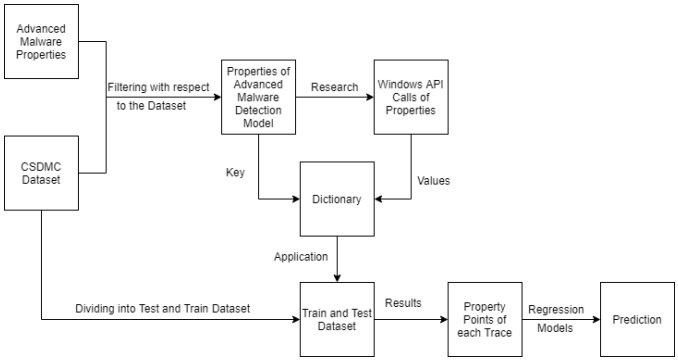

Fig. 1. The high level algorithm to predict advanced malware with machine learning.

properties may change over time. Moreover, initial instances of advanced malware have greater size than conventional malware. These make the detection of advanced malware almost always impossible with conventional malware detection tools. On the other hand, advanced malware and conventional malware has some common properties that may be used to detect the presence of advanced malware on a specific host.

Stuxnet, Duqu, Flame, and Red October are the most common examples of advanced malware seen in the wild [10]. They share some common properties but they also have some differences. For example, Stuxnet is the one which infects removable drivers, local area networks, programmable logic controllers(PLC). It exploits vulnerabilities of systems that do not provide secure message verification and source authentication [11]. Additionally, Stuxnet has rootkit functionality, self-replication property over the network and it infects programs and uses encryption methods. The main purpose of Stuxnet is sabotage [10].

Duqu was found in September 2011 by CrySys. It has Command and Control servers. Moreover, Duqu has an auto destruction component that is based on time triggering mechanism, which measures time taken after the infection. In some resources, it is considered as stealthy spyware [12]. Interestingly, it has only manual replication property. Additionally, Duqu uses AES algorithm for encryption operations. The main goal of Duqu is information gathering [10]. In other words, its goal is espionage.

Flame was discovered in May 2012, which had been considered active for the last 5-8 years. Flame is different than the other advanced malware in terms of the size, which is approximately 20 megabytes [13]. Like Duqu, it replicates itself manually. It uses encryption mechanisms. Flame is also used for information gathering [10]. Red October was discovered in October 2012 and considered as active since May 2007. Initially, it infected Microsoft Office programs and Java. The malware also replicates manually. It has also keylogging module and encryption property. It is used for espionage [10]. Since advanced malware has additional properties than conventional malware, it is also considered as a cyber weapon.

III. A Model to Predict Advanced Malware

The model to predict advanced malware is based on distinguishing features of malware. After analyses of advanced malware, we define five features related to advanced malware, which are conventional malware arsenal, behavior instability, stealth, metamorphic engine, and closeness to Stuxnet. These features are expected to be used for the detection purpose. Conventional malware arsenal represents many activities of malware, such as screen capture, anti-debugging, downloader, DLL injection and dropper [14]. Advanced malware has not only new features but also it uses conventional malware feature. Figure 1 shows the high level algorithm to predict advanced malware.

Behavior instability is a significant feature to distinguish advanced malware. In our model, we take into account read/write files, search file to infect, load register, modify file attributes, get file information, distribute global/virtual memory, copy/delete files, and access to files [15], [16]. We experimentally analyze this feature to show its correlations with other features.

Stealth property is one of the key characteristics of advanced malware. Advanced malware has lots of propagation mechanisms and it uses many hiding techniques, which is a distinguishing property of advanced malware. Having a metamorphic engine may help to hide the traces of advanced malware. Therefore, extracting correlations among features are key to design detection mechanisms for advanced malware.

Many instances of advanced malware share similar features with Stuxnet. Therefore, we observed that if malware is close to Stuxnet,this malware is also close to be advanced malware. For instance, Stuxnet has some rootkit functionality, XOR encryption, DLL PE excitability, and self-replication over the network, and use of removable devices [10].

We use five properties to predict advanced malware, which may be used for the detection. Our prediction model is based on correlations among these five properties of malware, which are represented with $M = \{X_1, X_2, X_3, X_4, X_5\}$. Additionally, the model is for Windows platforms and the five properties are represented with Windows API calls. We compute the prediction with equation 1.

$$A_i(X_i, P) = \prod_{\forall X_i, X_i \in M} S_i(X_i, P) D_i(X_i, P) \quad (1)$$

$$S_i(X_i, P) = \frac{X_i TypeSpecificAPICallsOfP}{AllAPICallsOfX} \quad (2)$$

$$D_i(X_i, P) = \frac{X_i TypeAPICallsOfP}{AllAPICallsOfP} \quad (3)$$

$A_i(X_i, P)$ represents the closeness score for malware P and property X_i. For instance, $A_1(X_i, P)$ represents Stuxnet closeness to stealth property if X_i is the stealth property. In these equations, all values are between zero and one. One represents maximum closeness whereas zero represents no correlation.

IV. Analysis of Malware Features with Machine Learning

We analyzed malware instances to extract correlations among features of malware to predict potential advanced malware. Specifically, we used regression algorithms to test the proposed model on malware dataset. We applied linear, polynomial, and random forest regressions with 3 estimators.

A. Dataset

We use a dataset to show correlations among malware features. The dataset was created during a data mining competition at the International Conference on Neural Information Processing in 2010. This dataset contains API calls and a label for each software. The label indicates type of software, namely malware or benign. The dataset can be found in servers of Artificial Intelligence Laboratory at University of Arizona [17]. Note that types of malware are not specified on the dataset.

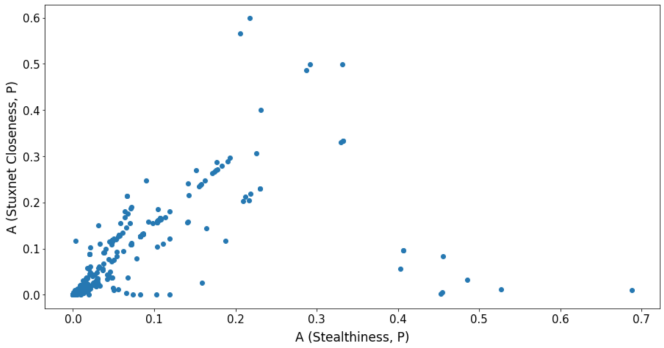

Fig. 2. The correlation between of Stuxnet Closeness and Stealthiness.

There are 622 malware instances in the dataset. Each line contains API calls that represents malware. In experiments, we discovered that there are correlations among the features, such as the correlation between Stuxnet Clossness and Stealthiness according to the proposed model and equation 1. Moreover, we found $A(Stealthiness, P) = 0.4$ and there is a correlation between $A(Stealthiness, P)$ and $A(StuxnetCloseness, P)$. In the region where $A(Stealthiness, P) > 0.4$, there are only 9 malware instances. We treated them as outliers and removed from our dataset. Then, there remain 613 malware instances. We used 450 of the instances as the training set and 163 of them as the test set. After we prepared training and test sets, we used regression algorithms to model the correlation between *A(Stealthiness, P)* and *A(Stuxnet Closeness, P)*.

B. Features

In this paper, we take into account Windows API calls that are directly related to the five properties, which are stealthy, conventional malware arsenal, Stuxnet closeness, behavioral instability and metamorphic engine. We omit other properties and corresponding Windows API calls. Features and related Windows API calls used in this work are as follows.

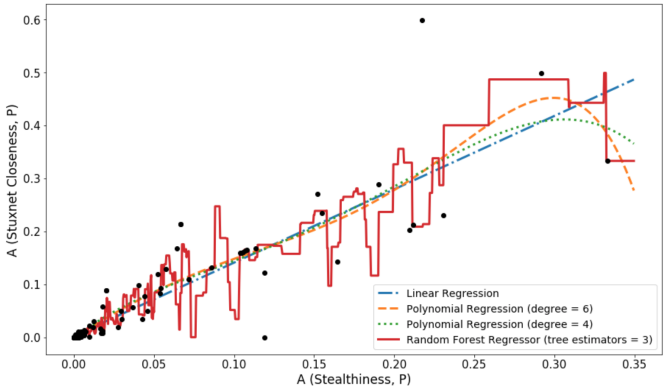

Fig. 3. Regression models on Stealthiness & Stuxnet Closeness.

- Stealthiness: FindFirstFileA, FindNextFileA, GetProcAddress, LoadLibraryA, OpenProcess, Sleep [18].
- Conventional Malware Arsenal: ShowWindow, GetWindow, WriteFile, WinExec, ShellExecuteA, OpenProcess, VirtualAlloc [14].
- Stuxnet Closeness: LoadLibraryW, LoadLibraryA, GetModuleHandle, GetProcAddress, VirtualAlloc, VirtualFree [19].
- Behavioral Instability: WriteFile, CreateFileA, CreateFileW, CloseServiceHandle, FindFirstFileA, FindNextFileA, FindClose, SearchPathA, SearchPathW, RegOpenKeyA, RegCreateKeyA, RegCreateKeyExa, RegCreateKeyExw, RegCreateKeyW, RegSetValueExa, RegSetValueExw, RegCloseKey, DeleteFileA, DeleteFileW, GetFileAttributesA, GetFileAttributesW, GetFileAttributesExA, GetFileAttributesExW, GetFileInformationByHandle, GetFileSize, GetFileType, GetFullPathNameA, GetFullPathNameW, GetLongPathNameW, GetShortPathNameA, GetShortPathNameW, GetTempFileNameA, GetTempPathA, GetTempPathW, GlobalAlloc, GlobalFree, VirtualAlloc, VirtualFree, CopyFileA, DeleteFileA, DeleteFileW, GetFileSize, GetFileType, ReadFile [15], [16].
- Metamorphic Engine: HeapAlloc, LocalFree, HeapCreate, GetStartupInfoA, GetCommandLineA, GetEnvironmentStringsW, FreeEnvironmentStringsW, GetModuleFileNameA, GetCurrentProcess, CloseServiceHandle, GetCurrentProcessId, GetProcessHeap, HeapReAlloc, SetFilePointer, SetFileAttributesA, GetFileAttributesW, FindFirstFileA, FindClose, SetThreadPriority, GetCurrentThreadId, GetProcAddress, GetModuleHandleA, ResumeThread, GetEnvironmentVariableA, ExitThread [20].

C. Experimental Evaluation of Malware Data

We applied three machine learning algorithms, linear regression, polynomial regression, and random forest regression to extract correlations among the properties of malware. The distribution of data is shown in Figure 2.

Stealth and Closeness to Stuxnet features are shown Figure 3. We use an outlier that filers the traces having more than 0.4 stealthy property to be able to handle correlations properly. The linear regression R^2 score is 0.817 for the test data. Polynomial regressions with degree 4 and 6 R^2 scores are 0.836 and 0.842 respectively. On the other hand, Random forest regression with three estimators has 0.785 R^2 score, which smaller than linear and polynomial regression scores. Table II contains scores of all machine learning algorithms applied to the dataset.

TABLE II
R^2 SCORES OF REGRESSION ALGORITHMS

Algorithm	R^2 scores
Polynomial Regression (d = 6)	0.842
Polynomial Regression (d = 4)	0.836
Linear Regression	0.817
Random Forest Reg.	0.785

The experimental results show that API calls related to Stealthy property and Stuxnet Closeness property are easily represented with linear and polynomial regressions. Actually, the smallest number of API calls related to these two features makes the features suitable for these types of regressions. Figure 4 shows all correlations between any two features. Analyses results show that Conventional Malware Arsenal (CMA) property differs from other properties. The main reason for this difference is that there are limited number of API calls related to this feature. On the other hand, there are many API calls related to behavioral instability and metamorphic engine features therefore they are not suitable to be analyzed with linear and polynomial regressions.

Analyses results show that there are different correlations between properties of advanced malware. Therefore, specific machine learning algorithms should be applied to particular pairs of malware properties to predict advanced malware. Additionally, machine learning algorithms must be tested with more data, particularly with advanced malware data, for prediction purposes.

V. CONCLUSION AND FUTURE WORK

Advanced malware has distinguishing properties than conventional malware therefore detecting advanced malware and preventing attack with that malware is exceptionally difficult. In this paper, we extract distinguishing properties of advanced malware according to known instances in the wild. Moreover, we analyzed existing malware data to show correlations among the distinguishing properties with machine learning algorithms. Analyses results show that our approach may support anti-malware system to predict advanced malware instances. Since there are limited information about advanced malware instances in the wild, we have been working to apply

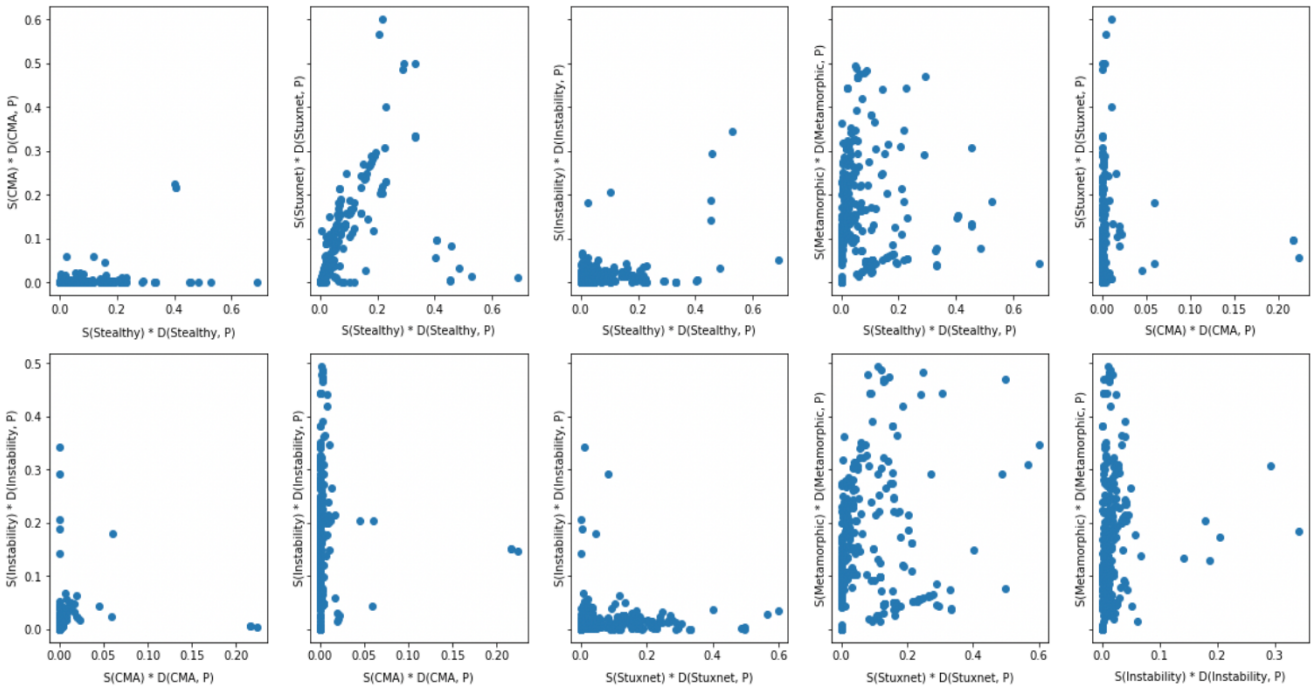

Fig. 4. Correlations between two properties of malware.

machine learning algorithms on codes of new found advanced malware instance to get better predictions.

ACKNOWLEDGMENT

This work is supported by Istanbul Technical University under the BAP project, number MAB-2017-40642.

REFERENCES

[1] W. Yan, "Cas: A framework of online detecting advance malware families for cloud-based security," in *Communications in China (ICCC), 2012 1st IEEE International Conference on*. IEEE, 2012, pp. 220–225.

[2] J. Jansen and R. Leukfeldt, "Phishing and malware attacks on online banking customers in the netherlands: a qualitative analysis of factors leading to victimization," *International Journal of Cyber Criminology*, vol. 10, no. 1, p. 79, 2016.

[3] N. Kshetri and J. Voas, "Banking on availability," *Computer*, vol. 50, no. 1, pp. 76–80, 2017.

[4] Ş. Bahtiyar, "Anatomy of targeted attacks with smart malware," *Security and Communication Networks*, vol. 9, no. 18, pp. 6215–6226, 2016.

[5] V. Paxson, "Viruses and worms," 2011.

[6] T. Yagi, N. Tanimoto, T. Hariu, and M. Itoh, "Investigation and analysis of malware on websites," in *Web Systems Evolution (WSE), 2010 12th IEEE International Symposium on*. IEEE, 2010, pp. 73–81.

[7] CERT-UK, "An introduction to malware," www.cert.gov.uk/resources/best-practices/an-introduction-to-malware/, 2014.

[8] D. Plohmann, E. Gerhards-Padilla, and F. Leder, "Botnets: measurement, detection, disinfection and defence," in *ENISA workshop on*. Mar, 2011.

[9] E. Skoudis and L. Zeltser, *Malware: Fighting malicious code*. Prentice Hall Professional, 2004.

[10] N. Virvilis and D. Gritzalis, "The big four-what we did wrong in advanced persistent threat detection?" in *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*. IEEE, 2013, pp. 248–254.

[11] A. Clark, Q. Zhu, R. Poovendran, and T. Başar, "An impact-aware defense against stuxnet," in *American Control Conference (ACC), 2013*. IEEE, 2013, pp. 4140–4147.

[12] G. Bonfante, J.-Y. Marion, F. Sabatier, and A. Thierry, "Analysis and diversion of duqu's driver," in *Malicious and Unwanted Software:" The Americas"(MALWARE), 2013 8th International Conference on*. IEEE, 2013, pp. 109–115.

[13] N. Virvilis, D. Gritzalis, and T. Apostolopoulos, "Trusted computing vs. advanced persistent threats: Can a defender win this game?" in *Ubiquitous intelligence and computing, 2013 IEEE 10th international conference on and 10th international conference on autonomic and trusted computing (uic/atc)*. IEEE, 2013, pp. 396–403.

[14] S. Gupta, H. Sharma, and S. Kaur, "Malware characterization using windows api call sequences," in *International Conference on Security, Privacy, and Applied Cryptography Engineering*. Springer, 2016, pp. 271–280.

[15] C. Wang, J. Pang, R. Zhao, W. Fu, and X. Liu, "Malware detection based on suspicious behavior identification," in *Education Technology and Computer Science, 2009. ETCS'09. First International Workshop on*, vol. 2. IEEE, 2009, pp. 198–202.

[16] M. Alazab, S. Venkataraman, and P. Watters, "Towards understanding malware behaviour by the extraction of api calls," in *Cybercrime and Trustworthy Computing Workshop (CTC), 2010 Second*. IEEE, 2010, pp. 52–59.

[17] CSDMC2010 Malware API Sequence Dataset, University of Arizona Artificial Intelligence Lab, AZSecure-data. Available http://www.azsecure-data.org/other-data.html [January 2017].

[18] E. M. Rudd, A. Rozsa, M. Günther, and T. E. Boult, "A survey of stealth malware attacks, mitigation measures, and steps toward autonomous open world solutions," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 1145–1172, 2017.

[19] A. Matrosov, E. Rodionov, D. Harley, and J. Malcho, "Stuxnet under the microscope," *ESET LLC (September 2010)*, 2010.

[20] V. P. Nair, H. Jain, Y. K. Golecha, M. S. Gaur, and V. Laxmi, "Medusa: Metamorphic malware dynamic analysis usingsignature from api," in *Proceedings of the 3rd International Conference on Security of Information and Networks*. ACM, 2010, pp. 263–269.