Cluster Comput (2017) 20:597–609
DOI 10.1007/s10586-016-0716-0

DFA-AD: a distributed framework architecture for the detection of advanced persistent threats

Pradip Kumar Sharma[1] · **Seo Yeon Moon**[1] · **Daesung Moon**[2] · **Jong Hyuk Park**[1]

Received: 15 November 2016 / Revised: 30 November 2016 / Accepted: 7 December 2016 / Published online: 20 December 2016

Abstract Advanced persistent threats (APTs) are target-oriented and advanced cyber-attacks which often leverage the bot control and customized malware techniques in order to control and remotely access valuable information. APTs generally use various attack techniques to gain access to the unauthorized system and then progressively spread throughout the network. The prime objectives of APT attacks are to steal intellectual property, legal documents, sensitive internal business and other data. If an attack is successfully launched on a system, the timely detection of attack is extremely important to stop APTs from further spreading and for mitigating its impact. On the other hand, internet of things (IoT) devices quickly become ubiquitous while IoT services become pervasive. Their prosperity has not gone unnoticed, and the number of attacks and threats against IoT devices and services are also increasing. Cyber-attacks are not new to IoT, but as the IoT will be deeply intertwined in our societies and lives, it becomes essential to take cyber defense seriously. In this paper, we propose a novel distributed framework architecture for the detection of APTs named as distributed framework architecture for APTs detection (DFA-AD), which is a promising basis for modern intrusion detection systems. In contrast to other approaches, the DFA-AD technique for detecting APT attack is based on multiple parallel classifiers, which classify the events in a distributed environment and event correlation among those events. Each classifier method is focused on detecting the APT's attack technique independently. The evaluation results show that the proposed approach achieves greater effectiveness and accuracy.

Keywords Advanced persistent threats · Internet of things · Genetic programming · Classification and regression trees · Support vector machines · Dynamic Bayesian game model

✉ Jong Hyuk Park
jhpark1@seoultech.ac.kr

Pradip Kumar Sharma
pradip@seoultech.ac.kr

Seo Yeon Moon
moon.sy0621@seoultech.ac.kr

Daesung Moon
daesung@etri.re.kr

[1] Department of Computer Science and Engineering, Seoul National University of Science and Technology (SeoulTech), Seoul, Korea

[2] Department Network Security Research Team, Electronics and Telecommunications Research Institute, Daejeon, Korea

1 Introduction

The recent rapid development of the internet of things (IoT) and its ability to offer different types of services make it the fastest technology with an enormous impact on business environments and social life. IoT has gradually penetrated all aspects of modern people life, such as business, health and education, with the storage of sensitive information about people and financial transactions, companies, product marketing and development. The widespread distribution of connected devices in the IoT has created a huge demand for robust security in light of the developing interest of millions of associated devices and services around the world. IoT gadgets, when used to anchor a malware attack, can open our association to a few noteworthy attack vectors. These can incorporate low password exploitation, remote code execution, hidden monitoring functions, reverse engineering hardware, and man-in-the-middle. Advanced malware needs a single entry point, think of it as "patient zero" and then this anchor point can be utilized over and again to launch

attacks to anything connected to that network. DDoS is a standout amongst the most widely recognized strategies for cyber-attacking to bring down a site. At that point there are advanced persistent threats (APTs), which are fundamentally similar to DDoS attacks on steroids.

APT is a sort of unseen continuous and long-term penetrative network attack that can bypass the existing security devices detection system. It can modify and steal the sensitive data as well as specifically physical damage the target system. For example, a 2010 Nuclear power plant was attacked by Stuxnet [1]. Due to the intensive destructiveness, the systematic investigation to establish facts of APT has gotten more attention.

APTs are typically characterized by an advanced skillset, vast resources and extreme stealth [2]. The noticeable characteristics of APT are the lengthy and longtime process of an entire attack, so it is a double-edged sword for the protectors. At one end, the long period attacks build the trouble of detection, and at the other end, it offers more opportunities for discovering the attack. If the protectors can appropriately foresee the attack being conducted before the objective is accomplished, the system can be proactively secured against an APT attack.

At present, the cost of malicious activity over internet and cyber-attacks has been assessed to be around 1 trillion USD per annum around the world [3]. APTs are currently the greatest threat to organizations and governments [4]. These APTs create issues for existing detection techniques since the present methodologies depend on the known pattern or signatures of attacks. However, APTs normally APTs habitually use new security openings for attacks. Due to an effective APT attack, financial losses can be high as it is supported by numerous past exploration findings on APTs [5–7].

This paper presents an architecture framework for the detection of APTs in a distributed environment. In this research, we focus on contributing to intrusion detection systems, mainly for the detection of APT attacks. The motive of this research is to provide a new architecture for an intrusion detection system that processes network traffic and is intelligent enough to identify APT attacks. The APT attack detection technique is based on a different classifier and the correlation among the events. Applying voting scheme, the method will provide the result of network traffic. All of these processes will work in a distributed environment. The possibilities for utilizing this technique as a part of an APT's detection process are immense and unexplored.

The rest of this paper is organized as follows: In Chapter 2, the APT attack cases, detection methods, architectural requirements and related existing research are introduced. The distributed framework architecture for the detection of APTs is proposed in Chapter 3. Chapter 4 describes the experimental analysis and comparison. Lastly, the conclusion of the paper is presented in Chapter 5.

2 Related work

2.1 APT attack cases

APT attacks have achieved the goal by using the methods utilized by a variety of malicious codes. Many modules, such as the data gathering module, system management module authority, and antivirus detection evasion module, and different Zero-day vulnerabilities are used here.

Spear phishing Spear phishing attacks, meaning it is based on tricking the user as a "spear", "phishing" is a combination of. First, the attacker, the members involved, to send a malicious e-mail disguised as reliable information. The idea is to spy on it secretly and to steal the confidential information of individuals or organizations. Then, the recipient during the download of trick attachment of the message content. As a result, it has been infected with the recipient has downloaded due to file. After infection, the attacker controls the data in the remote, try to steal [8]. A Spear phishing e-mail contains the malicious file with the normal file and double extension. When the malicious files of the double extension are executed, to create it, it will be run on the path of the normal document file. At the same time, normal document files that have been secretly executed may fail in the user to recognize it and may install an additional malicious file called conhost.exe to a temporary path. This malicious code is, you wait for additional commands from the remote address. Finally, In order to steal information or install additional malware variants, the attacker can remotely control or provide additional commands [9].

Duqu This has the ability to collect information. This is to avoid the monitor using the digital signature of the C-Media look legitimate [10]. This object is the stage of information gathering for the attack. In order to hide, zero-day vulnerability attacked attaching itself to the MS document file and delivered via e-mail. It generally infects after the user views the document file that contains a TTF font file, which is where Duqu is hidden. Data about this vulnerability is listed as CVE-2011-3402. A vulnerability in MS Windows that can permit for codes to be run on the vulnerable system for an attack has been found. Duqu collects information on the file, input key, process, network nad other logs. The gathered information is then stored and encrypted with ~DQ (num).tmp in the log file. A key component of logging gathers vital data as a password. The information collected is utilized to procure the power for the invader to be able to access another system in the network. When exchanging information with the remote address, it runs the downloads and uploads of the data to a JPG file [11].

Watering-hole The idea of a watering hole attack is practically identical to a predator waiting at a watering hole in a desert, as the predator knows that its victims will need

to go to the watering hole. In the same way, rather than actively sending malicious emails, attackers can recognize third party websites that are repeatedly visited by the targeted persons, and then try to infect one of these websites with malware. Finally, when the infected web pages are seen by victims, the delivery achieves [12]. The usage of watering hole attacks has been observed in numerous APT campaigns [13–15,37].

Carbanak This is an APT attack that is aimed at financial organizations. This attack gets close to the target system using a phishing email. Furthermore, it utilized weakness as a part of Microsoft word and office. Carbanak was installed a backdoor on the user's computer, which was not visible in the svchost.exe process. In order to prevent the malware safety program on the target system is not detected, if attempts to privilege escalation exploit the Windows vulnerability series of action. Moreover, the client's keystroke data with screenshot each 20 s the data was accumulated. They usage of remote desktop protocol (RDP) to sustain an association with an infected computer. As a result, they can regularly steal money [16].

2.2 Anomaly detection methodologies

Anomaly detection (AD) is an appropriate issue that has been attempted in various areas of application and research. Its significance lies in the fact that data anomalies interpreted to significant and often critical, actionable information in a wide range of application areas. For an analysis of the AD methodologies, the reader is referred to [17], covering all specifically developed for the more generic application. For example, areas for applying application AD approaches include: banking—credit card fraud detection, medicine—finding tumors in magnetic resonances, cybersecurity - traffic analysis of the network to detect intrusions, and space—Sensor conduct examination to avoid spaceship failure.

Different specialized methodologies can also be found such as statistical, clustering-based and classification-based techniques. Further classification-based methodology can be categorized into different approaches such as bayesian networks, artificial neural networks, rule-based systems and support vector machines.

When choosing an AD technique, more focus must be given to the particular characteristics of the problem, such as the sort of anomalies sought, the type of data, device computing capacity, and so on. In the situations where several characteristics that define the behavior of the dataset are accessible, the recognition technique should preferably incorporate data in the properties. Moreover, inconsistencies can be recognized by examining the attributes separately or by considering them as a whole.

2.3 Research issues and challenges

The main motive of this study is to introduce a novel method that identifies all potential APT attacks. To accomplish the goal we need to address the research issues and challenges listed below.

Detection technique To detect the potential techniques utilized throughout the APT attack lifecycle, it must be determined which detection approaches can be used.

Flexibility and extensibility The attackers try to discover new procedures to launch an attack every time. Things being what they are, it must then be determined by what means we would be able to build the type of detection system that is result of our methodology that is both flexible and extensible [38]. For this reason, we need every detection method to be autonomous from alternate systems. Moreover, at whatever time we can enhance or add a novel technique to the system and correlate with the alternative techniques in the frame correlation for detecting a new approach that has been utilized as a part of APT attack lifecycle.

Real-time network traffic The detection system must be able to withstand real-time detection. This is because if an attempted attack is detected immediately, it can be much easier to retrace the attacker, avoid further break-ins and prevent or minimize the damage.

Effectiveness The effectiveness of the scheme means the capability to recognize APT attacks should be accurate and highly efficient. The effectiveness of the scheme includes high accuracy and low false notices. We believe that the chance in case of a false positive is less when there is an immediate connection to different steps or relationships among the events. Keeping in mind the end goal to accomplish efficiency, we need to detect the appropriate rules for association among the events, and this will be dependent upon the assessment of every technique.

2.4 Existing research

Mohamed Abomhara et. al.'s research [41] presented about security threats and challenges in IoT. The authors have attempted to classify the types of threat, in addition to analyzing and characterizing the intruders and attacks faced by IoT devices and services. Yunsick Sung et al. [42] proposed a secure architecture that is required for protection against various threats. Protection includes threat prevention, access control and data protection that campaigns against network attacks such as ARP and DDoS spoofing.

Shun-Te Liu et al.'s research [18] presented how to recognize the possible infected hosts N of the attack, and their strategy depends on the learning obtained from past APT attacks. They built up an engine to search for an APT analyst in order to rapidly uncover the probable infected hosts in view of the learning gained about a known APT infected

host by enhancing the execution as far as detection rates and false positives.

In 2011, various targeted attacks were identified by Symantec. Olivier Thonnard et al. introduced this big corpus of APTs has been studied deeply [19]. On the basis of advanced analytics TRIAGE, they can characterize several cutting-edge threats to attack campaigns that can be carried out by the same people. They inspected the flow and elements of those attacks and introduced new thoughts as being business as usual for the attackers required in those campaigns.

Martin Lee et al.'s research [20] described that APTs beside the same target could be connected by using an undirected graph. Furthermore, it is conceivable to recognize clusters and build an APT map that can reveal the activities of one group of malware writers.

To identify a potential targeted attack, Marco Balduzzi et al. [21] proposed a new system to detect a potential APT attack based on the information collected on the host side. The system is based on clustering approaches to deal with the host groups that have related behaviors against doubtful assets that they require, such as drive-by downloads or exploit kits, and command and control servers. The system was named SPuNge, which compares industry data and sites where guests work as the government or oil and gas to detect the attack interesting exercises.

An abridged version of the original analysis Duqu was introduced in [22]. A European organization has been targeted by a novel malware, Duqu, and sensitive data was stolen. The authors presented a Duqu tool that they created to detect Duqu and its variations.

Ping Wang et al. [23] developed an automatic malware detection system by training an SVM classifier based on behavioral signatures. The cross-check and acceptance method utilizes 60 genuine malware to tackle the exactness issue. The proposed system is comprised of three types of system modules including monitoring, backup, and recovery. The backup system offers reinforcement for vital system files in client and server. The monitoring system utilizes the prior reinforcement information of the system to recover client information and the operating system. The recovery system module uses technical reverse engineering to retrieve the infected process malware.

Youngjoon Ki et al. [24] evaluated the malware API call sequence and introduced a system that decides whether a file is unlawful in relation to a current system. The system obtains the information from the API that occurs during the execution time of the program and evaluates its DNA grouping data.

Alberto Dainotti et al.'s research [27] presented an automated system to detect anomalies based on the volume of network traffic caused by DoS attacks. They showed how the proposed scheme is able to improve the existing compromise between false alarm operations and the success rates and at the same time provide information on the anomaly time and the identification of anomalies near future.

Anukool Lakhina et al. [28] introduced a general technique for diagnosing faults. The technique is based on a separation of the large space occupied by a set of traffic monitoring networks in disjointed subspaces corresponding to normal and abnormal network conditions.

Walter de Donato et al. [29] designed an open source identification engine traffic tool for classifying network traffic. Designed architecture and features focused on evaluation, comparison, and the combination of different traffic classification techniques that can be applied to both direct traffic and traffic traces have been previously captured.

An automated combination of methods for traffic classification is presented by the authors in [30]. They examined six intelligent combinations of algorithms applied to the traffic classification of traditional and newer approaches using either the content of the packets or the statistical properties of the flow. A genetic programming-based framework has been presented to generate a function to combine a whole [31]. The system evolves with a combiner, which does not need other training phases after the classifiers are formed together with the heterogeneous components.

Botnets often sweep across large segments of the address space of the Internet for various purposes such as infecting or compromising hosts, to compile a list of future targets or the recruitment of a host in a botnet. Awareness about changing botnet characteristics and the diffusion of technology can improve our ability to navigate and mitigate their impacts. Alberto Dainotti et al.'s research [32] presented the documentation and visualization of the behavioral aspects of a current generating botnet, and to analyze in depth the many synergistic characteristics of its extremely well-coordinated analysis.

Ruchika Mehresh et al. [39] proposed a survival architecture against APT in a distributed environment. It involves surreptitious and tamper-proof detection and verification of node-to-node suspicious events. Zakiah Zulkefli et al. [40] introduced a schema that how the behavior of the user can contribute to the APT attack. The researchers studied the available mitigation approaches and its problems in the fight against APT attack by examining the root cause of the attack in the bring your own device (BYOD) environment.

3 DFA-AD distributed framework architecture

Conventional network attacks always attach and encroach the target system quickly and cause substantial irregularities. The attack era is short, so it is not necessary to consider the attack time, which is the component for evaluating the result of the attack. However, it is not similar to APT. APT attackers devote much time to hiding the anomaly of their attack

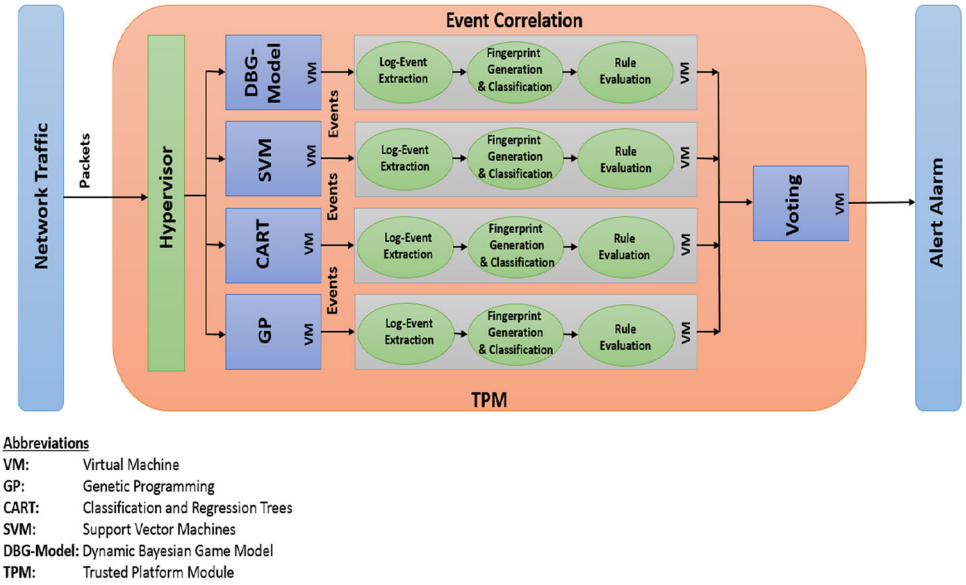


Fig. 1 DFA-AD distributed framework architecture for APT attack detection

behavior to accomplish a persistent attack. In this work, we are focusing on providing an intrusion detection framework, especially for APT attack detection. The main objective is to offer a new intrusion detection system that processes the network traffic and that is intelligent enough to identify an APT attack. The recognition of an APT attack depends on the relationship between the events that are generated by different classifier methods.

3.1 Design overview

To address these research issues and challenges, we designed and are proposing a new framework architecture for an intrusion detection system of network traffic for APT attacks in a distributed environment. As shown in Fig. 1, our proposed approach is consists of three main phases. We will perform this intrusion detection process in a distributed environment in the trusted platform module (TPM), where it stays hidden from the attackers. The TPMs that arrive as integrated into the modern motherboard of computer systems can be applied to implement secure node to node communication.

In the initial phase, we collected, processed, and analyzed the network traffic packets to identify all possible strategies that could be utilized as a part of an APT attack lifecycle. To perform this task, we have four different recognition methods, where each recognition method applies their own detection schemes to detect all the technique used in the different steps in an APT attack. Each method of detection is autonomous and independent of other methods. The outputs of these classifier methods should be submitted to the next event correlation phase.

In the second phase, we have the event correlation modules. This module takes all the events provided by the outputs of all detection classifier methods as an input and correlates all of them individually as indicated by the principles specified by the administrator to raise a caution on APT attack discovery. We can specify the principles by taking into account the evaluation of each technique. Once we events based on specified rules, the outputs should be submitted to the next voting phase.

In the third phase, based on the information provided by event correlation for different methods, the voting service will analyze and determine the final result. Voting among the different methods of detection triggers an alert on an APT attack. We took into account that this voting technique will lessen the rate of false positives and enhance the accuracy of our detection system.

3.2 Building classifier

To build a classifier for non-suspicious or suspicious behavior, we introduced a technique here which uses the premise that APT-infected traffic will have a tendency to be anomalous. This technique will train the classifier using labelled data with the help of experts.

As shown in Fig. 2, to train and build a classifier, we are using different steps as followed. Whenever the new traffic data gather, we assigned the anomaly score to those data using score metric in anomaly score metric process. We used the metric definition to assign metric score as described in the literature [35,36]. Once we assigned the metric score, we will divide the data into two categories based on some defined threshold value. If the metric score is less than the threshold value, then we will consider that data into the gray set (or labelled data). Moreover, the data whose metric score is greater or equal to threshold value will consider the dark

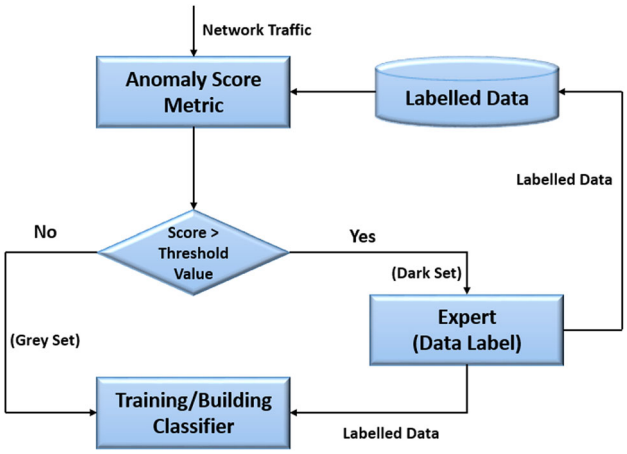


Fig. 2 Training/building classifier

set. We used to label dark data set by the help of a human expert. Based on the dark labelled data set, we will train and rebuild our classifier and update our labelled data storage accordingly.

3.3 Classifiers methods

We used four unique techniques to carry out the classifications cases, as described below.

A. Genetic programming

Genetic programming (GP) is a powerful and flexible evolutionary technique with a few elements that can be suitable and important for the evolution of classifiers [25]. GP is a subclass of genetic algorithms, which uses replication and mutation to develop structures. These programs consist of terminals (inputs and constants) and nodes (mathematical functions). GP deals with individual computer programs as genetic individuals that are potentially able to change or recombine to form new individuals.

GP arbitrarily generates an initial population of solutions. To produce new populations, the initial population is manipulated using several genetic operators. These operators include mutation, reproduction, dropping condition, and crossover. The whole process of developing from one population to the next population is known as a generation. An abstract level description of a GP algorithm can be split into a number of sequential steps, as described below.

- Create an arbitrary population of rules or programs using the symbolic expressions that are the initial population.
- Assess each rule or program by allocating a fitness value based on a pre-defined fitness function that can measure the capability of the program or rule to solve the problem.
- Copy existing programs into the new generation by using a reproduction operator.
- Generate the new population with a mutation or crossover or other operators from a randomly selected set of parents.
- Until a fixed number of generations have been completed, or a predefined termination criterion has been satisfied, repeat Steps 2 and onwards for the new population.

B. Classification and regression trees

In many areas, decision tree classifiers (DTCs) are utilized for classification issues. The most significant feature of a DTC is its ability to divide a complex decision-making process into a group of less complicated decisions, hence given that a solution which is often lighter to interpret. Classification and regression trees (CARTs) is the method in which the growth of branches builds the tree and prunes the iterative. At each internal mode, CART only allows for either a linear combination of features or a single feature.

Let's assume that there are M observations in the characteristics of learning sample data and that P_j is the total number of observations belonging to class j, $j = \overline{1, \mathrm{J}}$. Then set the class probabilities in Eq. 1 is as follows:

$$\left\{\pi\left(ij\right)\right\}_{\mathrm{j}=1}^{j=I} = \left\{\frac{P_j}{P}\right\}_{\mathrm{j}=1}^{\mathrm{j}=I} \tag{1}$$

That is part of the observations belonging to specific class related to a total number of observations.

Let $P\left(t\right)$ be the number of observations in nodes t and $P_j\left(t\right)$. The number of observations belongs to the jth class in the same node t. Then, the combined probability of the event that an observation of jth class comes into node t in Eq. 2 is:

$$p\left(j,t\right) = \pi\left(j\right)\frac{P_j\left(t\right)}{P_j} \tag{2}$$

Thus, $p\left(t\right) = \sum_{j=1}^{I} p\left(j,t\right)$ and the conditional probability of an observation to belong to node t given that its class j is computed in Eq. 3 is as follows:

$$p\left(j|t\right) = \frac{p\left(j,t\right)}{p\left(t\right)} = \frac{P_j\left(t\right)}{P\left(t\right)} \tag{3}$$

That is the proportion of class j in node t. It is noticeable that $\sum_{\mathrm{j}=1}^{I} p\left(j|t\right) = 1$.

C. Support vector machines

Support vector Machines mSVMs) are one of the most supervised learning module machines that analyze data and identify patterns. Given a set of training samples, we only have data from a class and test the new data class SVM is

the best method to see if it is similar or not compared to the data sample training. Thus, in the detection of abnormalities, class the SVM approach is applied to classify an anomalous packet aberrant [34].

For classification in view of the class hyperplanes, SVM algorithms are commonly utilized. It can be demonstrated that the ideal hyperplane is characterized of as having the most extreme partition edge between the two classes and as having the smallest limit. It can be constructed by solving a constrained quadratic optimization problem. It merits underlining the critical way of one of this algorithm: both the final decision function and the quadratic programming problem depend just on the dot products among the patterns. This is accurately what lets this technique be summed up to the nonlinear case.

D. Dynamic bayesian game model

Based on the diverse requirements of the application, the game models are typically classified into one of the following two different viewpoints. First, the game model can be split into a dynamic and static game based on the players choosing behaviors. The game is a static game if the player has selected the behaviors at the same time; otherwise, it is a dynamic game. Second, the game model can be split into an incomplete and complete information game based on the players having the whole information of others. With respect to APT, players individually select their behavior based on the current system status and the known information, which is a dynamic procedure. Prior to the attack being triggered, research attackers search the special users and target system thoroughly and comprehensively. The attackers have more data about the module than protectors, which created the incompleteness as well as asymmetry. Therefore, the dynamic Bayesian game model (DBG-Model) and incomplete information are required for APT analysis.

The dynamic game process can be illustrated as follows depending on the state of the system and the known information; every player selects the behavior of space behavior respectively. Each chosen behavior offers the most extreme advantage and activates the system status change. With respect to the changed system status, the players select another behavior. According to this method, the DBG-Model can be defined by a group of seven components.

$$\text{DBG-Model} = (S, \text{E}, \text{V}, \text{B}, \text{X}, \text{W}, R)$$

where,

- $S = (s_1, s_2, \ldots, s_n)$ denotes the status space of target system, which consists of all of the conceivable system statuses. Throughout the attack procedure, the system status shift take after with the players' behaviors. The shift rule is limited by the use of system vulnerabilities, the system nodes of the connection and the capabilities of players.
- $E = \{E_A, E_D\}$ refers to the entities that can choose autonomous behavior during the game process. It can be a group, an association, or one individual. Here, we considered protectors D and attackers A.
- $\text{V} = (V_A, V_D)$ refers to the rewards granted to players. It is the essential key variable in the model. The players choose the behaviors to amplify their own results.
- $B = \{B_A, B_D\}$ denotes the behavior space of the players, while $B_A = \left(b_A^1, b_A^2, \ldots, b_A^k\right)$, $B_D = \left(b_D^1, b_D^2, \ldots, b_D^l\right)$ refers respectively to all attackers and defenders of behavior.
- $\text{X} = \{X_A, X_B\}$ denotes the earlier probability of every player, which is utilized to assessing the type of other players on the basis of known information.
- $\text{W} = \{W_A, W_B\}$ denotes the players' type space, where $W_A = (w_1, w_2, \ldots, w_m)$, $W_B = (w_1, w_2, \ldots, w_n)$ respectively represents all types of attackers and defenders. The players that come into different types refer to different behavior levels.
- R is the result of the game model, which can be used to predict the behavior of players.

3.4 Event correlation

Figure 1 provides a conceptional overview of the event correlation phase in a DFA-AD framework. This visualization divides the event correlation phase into three different stages as follows: Log-event extraction, Fingerprint generation and classification, and Rule evaluation.

A. Log-event extraction

An essential unit of logging data such as one XML-element, one binary log data record, or a line for logging based on the line, is called a log-atom L_x, and L_x is comprised of a unique series of symbols e in Eq. 4.

$$L_x = e_1, \ldots, e_n \tag{4}$$

Furthermore, the log event L_y in Eq. 5 is the combination of a log-atom L_x with the timestamp t. L_y defines when L_x has been made.

$$L_y = \langle L_x, t \rangle \tag{5}$$

The first assignment gathers connection information from different sources or single source distributed in the controlled network and sends one by one to the next stage in the Event Correlation phase.

B. Fingerprint generation and classification

In this stage, vectorises each log-atom using P. In Eq. 6, P is a search pattern that is a substring of the log-atom L_x.

$$P = e_{1+j}, \ldots, e_{m+j} \tag{6}$$

where $0 \leq j$ and $m + j \leq n$.

The vectorization procedure converts a log-atom L_x into an multidimensional pattern vector, which is named fingerprint $\overrightarrow{F}$. $\overrightarrow{F}$ is generated after the fingerprint is classified after L_x is vectorised. Classification is the procedure that decides C_{L_x}. In Eq. 7, the set of all event classes C for log-atom L_x have a place with. One L_x can have a place with a huge number of classes. For example, a log-atom may be a "ssh event service", an "IP-zone X event service", and an "incoming connection event" in the meantime. Every event class that belongs L_x, encrypts a specific type of data that was initially encoded in L_x. Note that L_x is classified, not L_y because categorization is an independent timestamp.

$$C_{L_x} = \{C | L_x \in C\} \tag{7}$$

In the case where a log-atom does not have a place with any class at all then L_x is rejected and not utilized for further assessment. Since it cannot be mapped to any class C of the existing event, the information encrypted in L_x is lost. The combination of a value $\overrightarrow{C_v}$ and a mask $\overrightarrow{C_m}$ is defined as an event class C in Eq. 8.

$$C = \langle \overrightarrow{C_v}, \overrightarrow{C_m} \rangle \tag{8}$$

The event mask $\overrightarrow{C_m}$ works as a filter and determines which research models must be to considered as significant for classification in the separate class in Eq. 9. The value $\overrightarrow{C_v}$ chooses for all appropriate research designs, if they are not allowed to be part of $\overrightarrow{F}$ or enforced on $\overrightarrow{F}$, for L_x to be classified by C as shown in Eq. 10.

$$\overrightarrow{C_m} = p_1, \ldots, p_n \tag{9}$$

$$\text{where } p_j = \begin{cases} 0 & \textit{if P at j is irrelevant} \\ 1 & \textit{if P at j is relevant} \end{cases}$$

$$\overrightarrow{C_v} = p_1, \ldots, p_n \tag{10}$$

$$\text{where } p_j = \begin{cases} 0 & \textit{if P at j is prohibited or irrelevant} \\ 1 & \textit{if P at j is enforced} \end{cases}$$

If the condition in Eq. 11 holds, each generated fingerprint F'' is classified by C.

$$\overrightarrow{C_v} = \overrightarrow{F} \wedge \overrightarrow{C_m} \tag{11}$$

C. Rule evaluation

The third stage of event correlation changes the unique log-events focus the relationships between them. Hypothesis H in Eq. 12 is a non- approved rule correlation between the two classes of different events. A hypothesis is utilized to assess the events that have been triggered by the processed log events. If $E^{C_{cond}} \rightarrow E^{C_{impt}}$ holds in t_w, then one assessment of a hypothesis is thusly composed as the test. The time window t_w represents the time span relative to t at which L_y that fired $E^{C_{cond}}$ occurred, wherein the involvement must take. The system automatically builds the correlation assumptions and after that tests them so as to learn more about event dependencies. In general, note that $t_w > 0$ fails. Several hypotheses make assumptions about events that must take place prior to other events. At the time of generation of H, t_w is fixed.

$$H = \langle C_{cond}, C_{impt}, \rightarrow, t_w \rangle \tag{12}$$

The system assesses the event flow against assumptions H without interruption. This assessment process provides for a Q_j event queue for every existing assumption H_j. All queues Q_j pay attention to events related to the individual hypothesis. A periodic assessment process occurs at the inputs of the respective Q_j. The assessment is conducted until no defined cases hold. At that point there are no more assessments to be done in the present state. The rule is fully assessed until another event is obtained by Q_j.

In this paper, we present the complete event correlation phase in the form of a finite-state machine. The state machine is shown in Fig. 3. Based on the above discussion, we categorized the three stages of the event correlation phase into five different states, which are as follows: Event extraction, pattern extraction & merging, event class generation, hypothesis generation and Rule evaluation.

3.5 DFA-AD work's flow

DFA-AD is a distributed framework architecture for APT attack detection. In DFA-AD, when the new network traffic

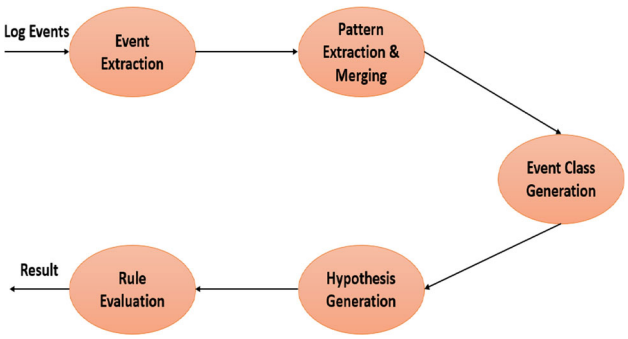
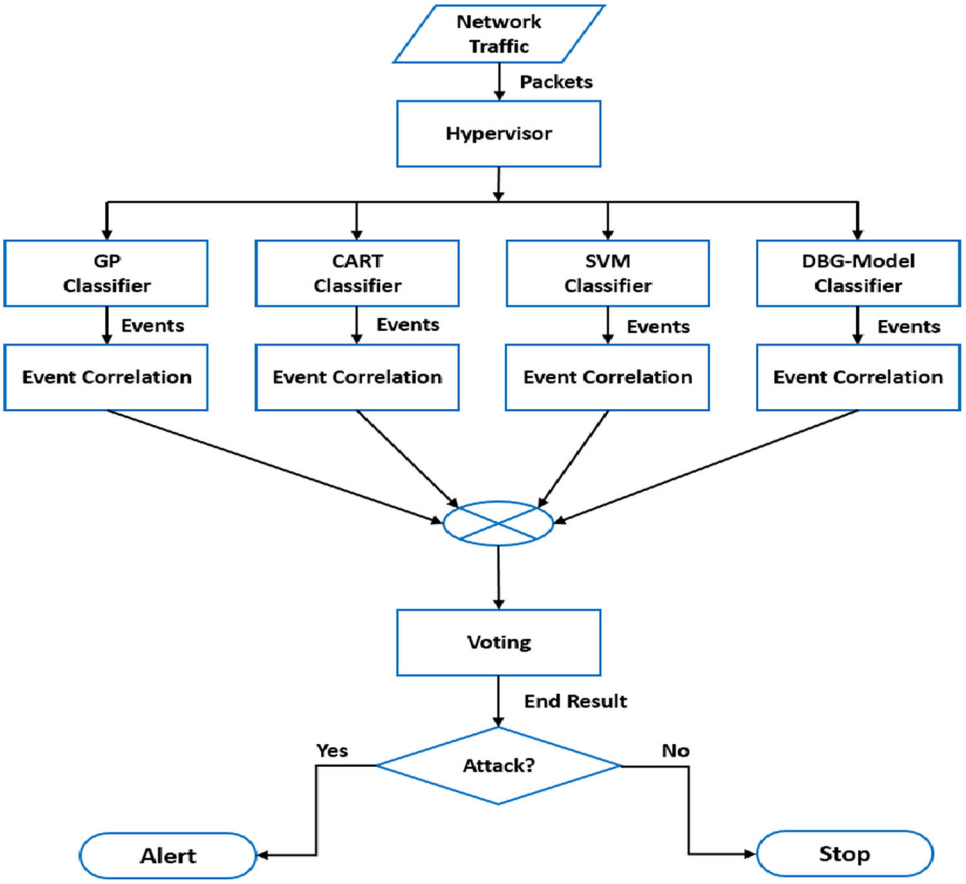


Fig. 3 The states of event correlation's workflow

Fig. 4 DFA-AD's workflow



packets arrive from the "diver's" sources in the network using different data collection techniques, it directly sends to the TPM for analysis. As we discussed above, the whole process is divided into three different phases. In the first phase, when the hypervisor receives the new packets, it sends out duplicate copies of each packet to the four different method classifiers. These method classifiers will process the network traffic packets independently from each other and generate events. Once the method classifiers complete the classification process, all the generated events will send out for the process in the next phase. In the second phase, all the generated events are processed parallel to search patterns, generate even classes, hypothesis and finally evaluate the rules. This phase is known as Event correlation. Once the event correlation process is over, each VM thread sends the output result to the next Voting phase. In the Voting phase, based on the results received from the event correlation decide whether there is any attack or not and generate alarm single accordingly. As an outcome, the system comes to know that if there is any malicious activity or attack. The complete workflow is depicted in Fig. 4.

4 Experiment and analysis

We simulated a distributed generic peer-to-peer network to quantify the execution of our proposed solution. All nodes are associated with each other directly or through a series of companions. To execute SimJava (2.0) simulations, we used a 62-bit Windows system with an i5 processor and 16 GB RAM. Using the technique suggested by Ruchika Mehresh et al. [33], we derived the simulation parameters. The objective of simulation is to assess the scalability and time performance of our solution.

To verify the effectiveness and correctness of the architecture of the proposed framework, this evaluation process uses different sets of data that have been generated by semi-synthetic generation technical data, as described and assessed by Florian Skopik et al. [26]. As we know, threats to modern ICT systems are rapidly evolving these days. The organizations are not primarily worried about virus infestation, but rather progressively need to manage targeted attacks. Such attacks are specifically intended to remain under the radar of standard ICT security frameworks. We chose these

data sets to mimic real systems and interactions between human users as closely as possible so as to generate network flow, system events, and the operating data of complex ICT services situations. This data is a key essential for the assessment of cutting edge interruption recognition and counteractive action frameworks. With these sets of data generated, a detection performance system can be accurately assessed and tuned for very specific settings. A virtual network of information and communication technology is stimulated by virtual users using scripts and is run on different VMs.

In the following assessment, two sets of data were recorded on a clean system. The decision of two distinctive data sets gives the evaluator the way to demonstrate steady results over different frameworks. An additional data set was recorded as anomalies were injected into the system that was being monitored at different times. The data set consists of two main parts: the training phase and the attack phase. In the phase of training data set, no abnormalities were injected. In the attack phase, the different types of anomalies were injected several times in different time slots.

4.1 Parameter evaluation

The first step in this section assesses the various parameters of the implementation of the prototype. One objective is to define a steady configuration of the framework. Another goal is to get a practical impression of the influence of various parameters on the system model and the outcomes created by the framework. Although the datasets depend on an ICT framework, the recorded setting does not contain any backup facilities or periodic tasks, except for the simulated user input. Based on the evaluation of the data generation approach [26], the activities of the mimicked clients achieve a demand appropriation. The system settings can be divided into the categories as described below.

Input dependent parameters of entries that have a different optimal setting that relies on the structure and size of the investigated dataset.

Input independent parameters that have an ideal setting that is not influenced by the structure and multifaceted nature of the entire analyzed data.

Utility parameters that are not influenced by the immediate after effect of the framework. Illustrations are the parameters defining how the framework can get to the database to ensure that execution occurs.

The accompanying assessment focuses on the setting's dependent parameters. Beginning with a base configuration, every parameter is independently modified to analyze its impacts on the resulting system.

4.2 Metrics

True positive rate (TPR) This is the ratio between the number of events that have been accurately classified as positive and the total number of events that can be classified as positive. The denominator of Eq. 16 is, therefore, the sum of the true positive (TP) and false negative (FN).

$$TPR = \frac{TP}{TP + FN} \tag{16}$$

where TP is the number of events, they are properly identified, and FN is the number of events that are incorrectly rejected.

False positive rate (FPR) This is the ratio of the number of positive events to the number of events that should have been negative. The denominator of Eq. 17 is the sum of the false positive (FP) and true negative (TN).

$$FPR = \frac{FP}{FP + TN} \tag{17}$$

where FP is the number of events, those misidentified, and TN is the number of events that have been completely rejected.

Precision, Recall, and F-measure Precision is the effectiveness of Zero-day detection system instances. It is also referred to as a positive detection value. Precision is the correctly classified to the total sample classified of the category. Moreover, recall is the fraction of correctly classified samples to a number of correct samples, and the F-measure is defined in Eq. 18 as the harmonic mean of precision and recall for measuring the accuracy of the system. The value of the F-measure is between 0 to 1. If the value of F-measure is closer to 1, then it is considered to have good accuracy and nearer to 0 is poor.

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{18}$$

4.3 Comparison analysis

To assess the overall performance of the proposed approach as compared to other methods, we tested all methods independently. Since some of these methods are stochastic, to minimize the effects of the variables selected randomly, a number of tests were carried out. Figures 4 and 5 show the outcomes in a scatter diagram. To assess the effects that parameter changes have on the results, indicators were calculated. The injected anomalies were detected, and the detection rates of our proposed approach were constant as compared to other methods.

We used voting threshold (VT) value 1 in Fig. 5 and VT value 2 in Fig. 6. As we can see in the graph, the accuracy rate of DFA-AD is better as compared to all the other methods. Figure 7 compares the DFA-AD approach at different

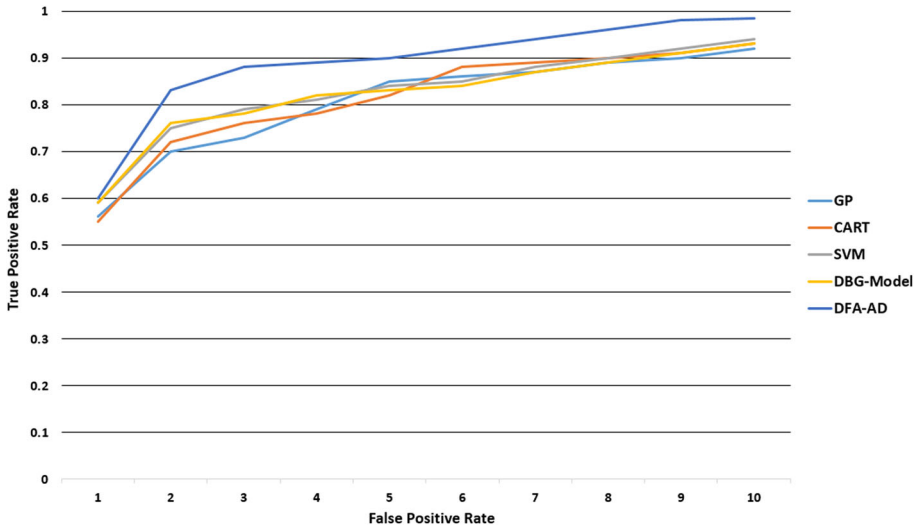


Fig. 5 TPR versus FPR at VT=1

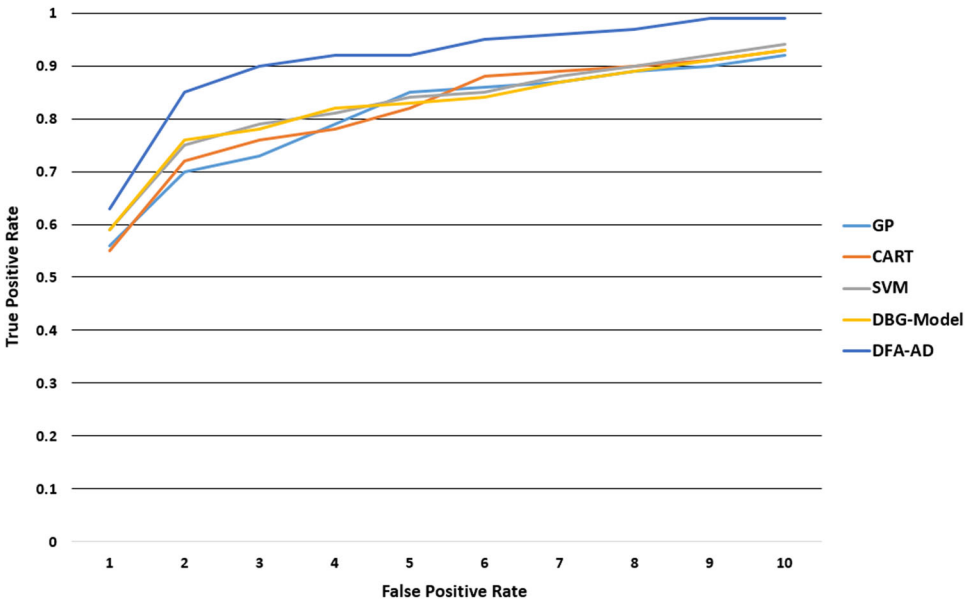


Fig. 6 TPR versus FPR at VT=2

VT values. It shows that the DFA-AD approach achieved a constantly high accuracy rate at the VT equal to 2.

The results support that if the average rate of FPR is less, the voting threshold will be greater than or equal to 2. The overall results show that the average percentage of TPR is high, and FPR is low as compared to all the other methods.

The evaluation showed that based on the assessments result, we could conclude that the proposed framework architecture scheme in a distributed environment is able to detect abnormalities that are the result of realistic APT attacks. By combing the multiple classifiers, the approach is able to efficiently and more accurately detect APTs. Because it works in a distribute environment, our approach increases the performance and level of accuracy.

We recorded the values of F-Measure, recall, and precision for Zero-day malware are shown in Fig. 8, to evaluate

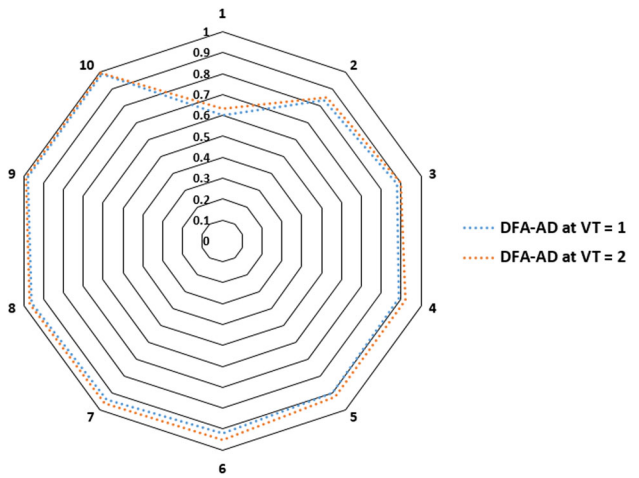


Fig. 7 DFA-AD accuracy rate at VT= 1 & 2

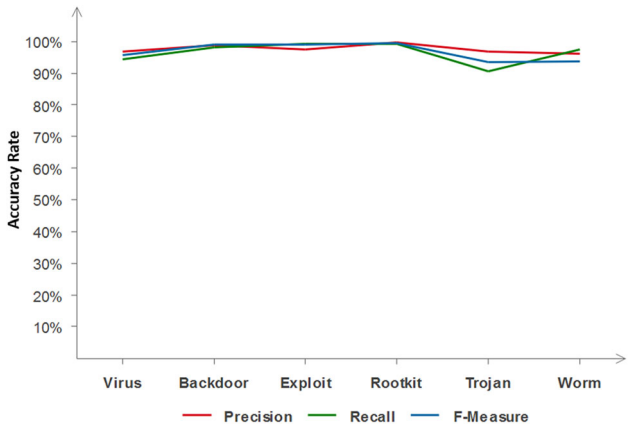


Fig. 8 Detection accuracy for Zero-day malware

the accuracy of the proposed DFA-AD approach. Malicious packets were directed to the implemented system. These packages were incorporated into tcpdump file for Snort to read. The results were exceptionally encouraging accomplishing the best detection rate of almost 98.5% with 0.024 false positive rates and in the most pessimistic scenario, recognition rate was 89.7% with 0.034 false positives. Therefore, the proposed framework has accomplished high accuracy with almost negligible false positives.

5 Conclusion

In this paper, a novel APT attack detection architecture that utilizes an event correlation technique from the events generated by various classifier methods in a distributed manner in ICT networks has been presented.

After studying the current state-of-the-art, we have concluded that signature-based detection methods and preventive security mechanisms are not sufficient to deal with novel, targeted and persistent threats. The proposed detection technique of an APT attack is based on the different parallel classifiers and the correlation between the events to detect the possible techniques used in an APT attack lifecycle. An experiment was designed to verify the efficiency of our proposed approach. The evaluation results show that DFA-AD achieved higher accuracy and effectiveness compared to other methods.

The opportunity to use this approach to detect of APTs is big and still unexplored. Future work could include optimizing and improving the used techniques to adjust them to the problem considered.

Acknowledgements This work was supported by Institute for Information & communications Technology Promotion (IITP) Grant funded by the Korea government (MSIP) (No.R-20160222-002755, Cloud based Security Intelligence Technology Development for the Customized Security Service Provisioning.

References

1. Iran confirms Stuxnet found at Bushehr nuclear power plant. http://warincontext.org/2010/09/26/iran-confirms-stuxnet-found-at-bushehr-nuclear-power-plant/. Accessed Aug 2016
2. Brewer, R.: Advanced persistent threats: minimising the damage. Netw. Secur. **4**, 5–9 (2014)
3. Kshetri, N.: The global cybercrime industry: economic, institutional and strategic perspectives. Springer, New York (2010)
4. Fossi, M., et al.: Symantec internet security threat report trends for 2010. Semant. Enterproses Secur. **16**, 1–20 (2011)
5. Tankard, C.: Advanced persistent threats and how to monitor and deter them. Netw. Secur. **8**, 16–19 (2011)
6. Kaspersky Lab ZAO. Red October diplomatic cyber attacks investigation. https://securelist.com/analysis/publications/36740/red-october-diplomatic-cyber-attacks-investigation/. Accessed Jul 2016
7. Mandiant, A.P.T.: Exposing one of China's cyber espionage units. https://www.fireeye.com/content/dam/fireeye-www/services/pdfs/mandiant-apt1-report.pdf. Accessed Aug 2016
8. Parmar, B.: Protecting against spear-phishing. Comput. Fraud Secur. **1**, 8–11 (2012)
9. Caputo, D.D., et al.: Going spear phishing: exploring embedded training and awareness. IEEE Secur. Priv. **12**(1), 28–38 (2014)
10. Faisal, M., Ibrahim, M.: Stuxnet, duqu and beyond. Int. J. Sci. Eng. Investig. **1**, 75–78 (2012)
11. Bencsáth, B., et al.: The cousins of stuxnet: duqu, flame, and gauss. Future Internet. **4**, 971–1003 (2012)
12. O'Gorman, G.; McDonald, G.: The Elderwood project. symantec whitepaper. http://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/the-elderwood-project.pdf. Accessed Aug 2016
13. Gragido, W.: Lions at the Watering Hole: The VOHO Affair. RSA blog. http://blogs.rsa.com/lions-at-the-watering-hole-the-voho-affair/ (2012). Accessed Aug 2016
14. Internet explorer 8 exploit found in watering hole campaign targeting Chinese dissidents. https://www.fireeye.com/blog/threat-research/2013/03/internet-explorer-8-exploit-found-in-watering-hole-campaign-targeting-chinese-dissidents.html (2012). Accessed Aug 2016
15. Operation Snowman: DeputyDog Actor Compromises US Veterans of Foreign Wars Website. https://www.fireeye.com/blog/threat-research/2014/02/operation-snowman-deputydog-actor-compromises-us-veterans-of-foreign-wars-website.html (2014). Accessed Aug 2016
16. Kaspersky lab. https://securelist.com/files/2015/02/Carbanak_APT_eng.pdf. (2015). Accessed Aug 2016
17. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM Comput. Surv. **41**, 15–73 (2009)
18. Liu, S.T., Chen, Y.M., Lin, S.J.: A novel search engine to uncover potential victims for apt investigations. In: Proceeding of IFIP international conference on network and parallel computing. Springer, New York (2013)
19. Thonnard, O. et al.: September. Industrial espionage and targeted attacks: understanding the characteristics of an escalating threat. In: Proceeding of international workshop on recent advances in intrusion detection. Springer, Berlin (2012)
20. Lee, M., Lewis, D.: Clustering disparate attacks: mapping the activities of the advanced persistent threat. https://www.virusbulletin.com/uploads/pdf/conference_slides/2011/Lee-VB2011.pdf (2013). Accessed Jul 2016
21. Balduzzi, M., Ciangaglini, V., McArdle, R.: Targeted attacks detection with spunge. In: Proceeding of 2013 eleventh annual international conference on privacy, security and trust (PST), IEEE (2013)

22. Bencsáth, B., et al.: Duqu: analysis, detection, and lessons learned.In: Proceeding of ACM European workshop on system security (EuroSec) (2012)
23. Wang, P., Wang, Y.S.: Malware behavioural detection and vaccine development by using a support vector model classifier. J. Comput. Syst. Sci. **81**, 1012–1026 (2015)
24. Ki, Y., Kim, E., Kim, H.K.: A novel approach to detect malware based on API call sequence analysis. Int. J. Distrib. Sens. Netw. **2015**, 1–9 (2015)
25. Espejo, P.G., Ventura, S., Herrera, F.: A survey on the application of genetic programming to classification. IEEE Trans. Syst. Man Cybern. **40**, 121–144 (2010)
26. Skopik, F., et al.: Semi-synthetic data set generation for security software evaluation. In: Privacy, security and trust (PST). IEEE twelfth annual international conference on 2014
27. Dainotti, A., Pescapé, A., Ventre, G.: Nis04-1: Wavelet-based detection of dos attacks. In: Proceeding of global telecommunications conference, GLOBECOM '06. IEEE (2006)
28. Lakhina, A., Crovella, M., Diot, C.: Diagnosing network-wide traffic anomalies. In: Proceeding of ACM SIGCOMM computer communication review, ACM (2004)
29. De Donato, W., Pescapé, A., Dainotti, A.: Traffic identification engine: an open platform for traffic classification. IEEE Netw. **28**(2), 56–64 (2014)
30. Dainotti, A., Pescapé, A., Sansone, C.: Early classification of network traffic through multi-classification. In: Proceeding of international workshop on traffic monitoring and analysis. Springer, New York (2011)
31. Folino, G., Pisani, F. S.: Combining ensemble of classifiers by using genetic programming for cyber security applications. In: Proceeding of European conference on the applications of evolutionary computation. Springer International Publishing, New York (2015)
32. Dainotti, A., et al.: Analysis of a/0 stealth scan from a botnet. In: Proceedings of the 2012 ACM conference on internet measurement conference, ACM (2012)
33. Mehresh, R., et al.: Tamper-resistant monitoring for securing multi-core environments.In : Proceeding of international conference on security and management (SAM) (2011)
34. Tian, M., et al.: Using statistical analysis and support vector machine classification to detect complicated attacks. In: Proceeding of international conference on machine learning and cybernetics, IEEE (2004)
35. Ingham, K. L., Inoue, H.: Comparing anomaly detection techniques for http. In: Proceeding of international workshop on recent advances in intrusion detection. Springer, Berlin (2007)
36. Kruegel, C., Vigna, G.: Anomaly detection of web-based attacks. In: Proceedings of the 10th ACM conference on computer and communications security, ACM (2003)
37. Singh, S., et al.: A comprehensive study on APT attacks and countermeasures for future networks and communications: challenges and solutions. J. Supercomput. pp. 1–32 (2016)
38. Hu, P., et al.: Dynamic defense strategy against advanced persistent threat with insiders. In: Proceeding of 2015 IEEE conference on computer communications (INFOCOM), IEEE, pp. 747–755 (2015)
39. Mehresh, R., Shambhu, U.: Surviving advanced persistent threats in a distributed environment-architecture and analysis. Inform. Syst. Front. **17**(5), 987–995 (2015)
40. Zulkefli, Z., Singh, M.M., Malim, N.H.A.H.: Advanced persistent threat mitigation using multi level security-access control framework. In: Proceeding of international conference on computational science and its applications, pp. 90–105. Springer International Publishing, New York (2015)
41. Mohamed, A., Geir, M.K.: Cyber security and the internet of things: vulnerabilities, threats, intruders and attacks. J. Cyber Secur. **4**, 65–88 (2015)
42. Sung, Y., et al.: FS-open security: a taxonomic modeling of security threats in SDN for future sustainable computing. Sustainability **8**(9), 919–944 (2016)