Süleyman Demirel Üniversitesi
Fen Bilimleri Enstitüsü Dergisi
Cilt 22, Özel Sayı, 519-528, 2018

Süleyman Demirel University
Journal of Natural and Applied Sciences
Volume 22, Special Issue, 519-528, 2018

DOI: 10.19113/sdufbed.98173

A Flow Based Approach to Detect Advanced Persistent Threats in Communication Systems

Şerif BAHTİYAR[*1]

[1]İstanbul Technical University, Department of Computer Engineering, 34469, Maslak, İstanbul

(Alınış / Received: 15.03.2018, Kabul / Accepted: 28.08.2018, Online Yayınlanma / Published Online: 21.09.2018)

Keywords
Security,
Malware,
Advanced persistent threat,
Attack,
Detection,
Communication

Abstract: The expansive usage of the Internet has set the stage for advanced persistent threats that has increased costs considerably in cyber space. Most of the time, entities exchange information and they are controlled remotely via many communication systems with a rich connectivity options on the Internet. Intruders accomplish advanced persistent threats by using such a rich connectivity options. These threats are extremely complex and they have unique features. Detecting such threats and corresponding attacks are therefore very difficult that circumstance makes classical intrusion detection systems impossible to deal with them. In this paper, a flow-based approach to detect advanced persistent threats is presented with a new model, namely FD-APT. The approach considers advanced persistent threats based attacks that are carried out with advanced malware. Moreover, FD-APT model distinguishes properties of malware types. The new approach is also analyzed with two case studies to highlight capabilities of FD-APT. The analyses results show that FD-APT helps to detect advanced persistent threats that are based on advanced malware.

Haberleşme Sistemlerinde Gelişmiş Sürekli Tehditleri Tespit Etmek İçin Akış Tabanlı Bir Yaklaşım

Anahtar Kelimeler
Güvenlik,
Zararlı Yazılım,
Gelişmiş sürekli tehdit,
Saldırı,
Tespit,
Haberleşme

Özet: Internet'in yaygın kullanımı, gelişmiş sürekli tehditlerin ortaya çıkmasına ve dolayısı ile siber uzaydaki maliyetlerin önemli ölçüde artmasına sebep olmaktadır. Çoğu zaman Internet, haberleşme sistemlerini kullanarak etmenler için bilgi alış verişini gerçekleştirmektedir ve bunların uzaktan kontrolü için zengin bir bağlantı seçeneği sunmaktadır. Saldırganlar, gelişmiş sürekli tehditleri bu zengin bağlantı seçeneği ile ellerinde bulundururlar. Bu tehditler son derece karmaşıktır ve benzersiz özelliklere sahiptirler. Bundan dolayı bu tehditleri tespit etmek son derece zordur, öyle ki klasik saldırı tespit sistemlerinin bunları tespit etmesi olanaksızdır. Bu makalede, gelişmiş sürekli tehditleri tespit etmek için akış tabanlı bir yaklaşım ve adını FD-APT verdiğimiz ilgili bir model önerilmiştir. Önerilen yaklaşım, gelişmiş sürekli tehditler tabanlı ve gelişmiş zararlı yazılımlar ile yapılan saldırıları dikkate almaktadır. Üstelik FD-APT gelişmiş zararlı yazılımların ayırt edici özelliklerine göre tasarlanmıştır. Önerilen yeni yaklaşımın kabiliyetlerini ortaya çıkarmak için iki örnek olay ile analiz çalışması yapılmıştır. Analiz sonuçları göstermektedir ki FD-APT gelişmiş zararlı yazılım tabanlı gelişmiş sürekli tehditlerin tespitine yardımcı olabilmektedir.

1. Introduction

The Internet contains a huge number of devices that run many services containing various security flaws. These devices are interconnected with many different communication systems. For example, surveillance cameras are now able to connect to the Internet running software and they can be integrated with different services. The devices may contain cutting edge technologies, such as mobile code. These technologies are now pervasive and the codes offer diverse opportunities to advanced persistent threats (APTs) with advanced malware that can easily propagate over the Internet. This makes systems vulnerable to attacks that possess advanced persistent threats more than ever. Such attacks may

* Corresponding author: bahtiyars@itu.edu.tr

result in significant economic losses and social effects [1].

Advanced malware has many unique features. For instance, it cannot be easily detected by conventional automated analysis mechanisms and malware detection tools [2]. Specifically, existing malware detection approaches and models are insufficient to detect advanced malware, such as Stuxnet and Duqu. Therefore, the challenge is to find an approach and to design a corresponding model to detect advanced persistent threats that use advanced malware to attack. The motivation of this work is the lack of such an approach and a model for cyber space that use various communication systems.

The contribution of this paper is a new flow-based approach to detect advanced persistent threats and a novel model to detect advanced persistent threats in communication systems. The model helps to detect advanced persistent threats based on features of advanced malware. The proposed model is analyzed with case studies. Particularly, an application is designed and is implemented to simulate the detection of advanced persistent threats. The analyses results show that the proposed model may help to detect advanced persistent threats in communication systems.

The rest of the paper is organized as follows. Section 2 is a brief overview of malware and advanced persistent threats. Section 3 describes the proposed approach. Next section is devoted to the analysis purpose. The paper is concluded in Section 5.

2. Malware and Advanced Persistent Threats

Malware is malicious software that is inserted into a system with the intent of compromising confidentiality, integrity, or availability of the victim's data, applications, or operating system [3]. According to Symantec's 2016 Internet Security Threat Report, more than 430 million new unique pieces of malware was discovered in the wild [4]. On the other hand, an advanced persistent threat is a set of advanced processes and software of a particular person or an organization attacking specific targets. Therefore, Recently, APTs are carried out with advanced malware like Stuxnet.

Advanced persistent threats carried out with advanced malware are a special kind of APTs. Therefore, it is crucially important to understand and distinguish conventional malware and advanced malware. In this section, types of conventional malware are described to be able to explain advanced malware and differences between conventional malware and advanced malware. Then, instances of advanced malware in the wild and their properties are explained to show relationships among advanced malware, APT, and conventional malware.

2.1. Conventional malware

Conventional malware refers to known malicious software, which exists for a long time. Moreover, all distinguishing properties of conventional malware are known. Actually, it is a very significant issue to identify properties of different malware types to be able to detect them accurately. Types of conventional malware are generally categorized as virus, worm, Trojan horse, keystroke loggers, backdoor, rootkit, and spyware [5].

A *Virus* is one of the most known malware type and it is generally used to refer all types of malware, which referral is incorrect. A virus is malicious software that can replicate itself by inserting its copies into victims' software or data files. Viruses may need a user interaction to activate themselves, such as opening a file or running a program [3]. This type of malicious software is generally attached to an executable file and it infects other targets [6]. Additionally, payloads of viruses are designed for different purposes, such as deleting, altering, damaging or corrupting files and data of victims [7].

A *Worm* is standalone malicious software that copies itself and infects other computing systems. Specifically, worms can infect many other systems, which are connected to each other via communication networks. This may result in excessive traffic on the networks. Since worms are standalone malware, they survive without any host software. This property is one of the main difference between a worm and a virus. An infected system with a worm may consume many system resources. Simply, a worm may consume the memory of infected computer, which may result in denial of service [7, 8].

Table 1. Malware types and their properties.

Malware/ Property	Virus	Trojan	Worm	Spyware	Adware	Bot	Rootkit
Self-replication	+	-	+	-	-	+	+
Self- propagation	+	+	+	-	-	+	+
Stealth	-	+	+	+	-	-	+
Executable	-	-	+	-	-	+	+
Insert a payload to the target	+	-	+	-	+	+	+
Advertising	-	-	-	-	+	+	-
Need host	+	-	-	-	+	-	-
Hidden components	-	-	+	-	-	-	+
Espionage	+	+	+	+	-	+	+
Denial of service	+	+	+	-	+	+	+

A *Trojan Horse* is malware that seems legitimate at first sight, but it is malicious software which has special goals like information stealing. Unlike a virus and a worm, a Trojan Horse does not replicate itself. Most of the time, it is used to create a backdoor on the targeted system in order to enable attackers or other malware to access the targeted system [7]. Kaspersky Lab classifies Trojan Horses according to their actions as follows [9]:

- *Backdoor* provides a remote-control facility for the attacker. Therefore, resources of a targeted system become vulnerable to attacks of malign users.
- *Exploit* makes use of vulnerabilities of an infected system.
- *Rootkit* is malware, which is designed to hide other malware and it prevents detections of malware.

Spyware is malicious software that is used to collect sensitive information by tracking actions of users. It sends gathered information to the owner of malware [5]. Spyware is generally installed on a targeted system without any permission of the system owner. For instance, spyware may be installed when a legitimate user of the targeted system downloads free software from the Internet.

Adware is malicious software for advertisement purposes. It is commonly used to download or display advertisements on infected systems when a user is online [10]. Commercial companies generally use adware for such purposes.

A *Bot* is malicious software that allows an attacker to take control of an infected computer system. The computer is usually a part of a networked system that is called Botnet [11].

Table 1 contains features of different types of conventional malware. In the table, "+" represents existence of the property for malware that is specified on columns. On the other hand, "-" means has no specified property, which are given on rows. It is clear that some types of conventional malware may have limited properties like Adware. On the other hand, some types of conventional malware may have more advanced properties than others, such as bots. Existing antimalware systems use these properties to distinguish types of malware. Moreover, current intrusion detection systems consider these properties to identify the origin of attacks. However, these properties are inadequate to detect advanced persistent threats that are carried out with advanced malware.

In this paper, all properties of conventional malware are considered in the new approach and the model. Detailed analyses of each conventional malware property is beyond the scope of the paper.

Table 2. Advanced malware instances.

Malware/ Property	Stuxnet	Duqu	Flame
Data separation	Built in	Add-on, for espionage	Multiple libraries, SQLite3, dll files
Size	Smaller than 1 MB	Smaller than 100KB	Approximately 20 MB
Self- propagation	Remote Procedure Calls, WinCC, Databases, Siemens industrial control systems	Not known yet	Multiple exploits and propagation methods
Dropper characteristic	Installs signed kernel drivers	Installs signed kernel drivers	Use a large program
Command and control	Communicate over HTTP	HTTP, HTTPS, a custom communication	SSL + SSH
Self- destruction	20120624 (Hard coded) to June 24, 2012	Uninstall after 36 days	Hide traces and self-destruction
Interaction with control systems	Siemens SCADA	-	Many command and control servers
Run specific code	CreateProcessAPI (Windows)	CreateProcessAPI (Windows)	Windows OS applications
Load a module	+	-	+
Access to specific location	+	+	+
Digital signature use	+	+	Not found yet
Interact with a component on the target	+	+	+
Infection mechanisms	USB, PDF, Drivers	Unknown but expected to be like Stuxnet	Ethernet, USB, Bluetooth
0-day exploit use	+	Not found yet	Not found yet
Sabotage	+	-	-
Written language	MSVC++ (C and C++), unknown language	Python, Ada, Lua	Lua, C++
Use of cryptography	+	+	+

2.2. Advanced malware and advanced persistent threat

In this subsection, advanced persistent threats are explained in more details. Roughly, an APT is a threat that contains many different attacks. Specifically, advanced persistent threats are increasingly sophisticated attacks that may use diverse number of vulnerabilities. Therefore, a novice reader may confuse among an APT, a threat, and an attack. APTs are carried out by hostile organizations that may have the following goals:

- Gaining access to targeted systems.
- Maintaining a foothold in targeted systems to enable future use of them and their control.
- Performing a denial of service attack by reducing performances of targeted systems.

Initially, Stuxnet was categorized as a worm, which was first reported in June 2010. It appears to be first advanced malicious software designed to attack a specific target. Particularly, a nuclear power plant is the target of Stuxnet. On the other hand, it affected many countries including Iran, Indonesia, India, Pakistan, Germany, China, and the United States. A lack of publicly available information on the damage caused by Stuxnet in these countries makes it difficult to determine the goal of Stuxnet precisely [12]. Actually, this is the main obstacle for researchers to be able to design antimalware models related to such malware and related advanced persistent threats.

Advanced malware is highly modular, therefore, this property allows owners of sophisticated malware to customize advanced malware for targeted attacks. Modularity of malware means a new era for malware creators that opens a new business model. Malware creators can work simultaneously on different parts of advanced malware and they may share them with each other to accomplish more effective advanced persistent threats. For example APTs carried out with advanced malware like Stuxnet include various pieces of specific codes [13], which are expected to be written by different creators.

Advanced malware has more complex properties than conventional malware [12, 14, 15, 16]. Therefore, the grand challenge for defenders is the lack of information related to these properties. Moreover, it is not clear how advanced malware establishes advanced persistent threats on different computing systems. In this paper, distinguishing properties of advanced malware in the wild are presented in Table 2 in order to detect APTs carried out with advanced malware. Moreover, a flow based approach and a corresponding detection model is designed according to these properties. The main idea behind this approach and the model is to help extending classical antimalware and intrusion detection systems against advanced persistent threats carried out with advanced malware by using distinguishing properties of advanced malware.

2.3. Properties of advanced malware

Significant properties of advanced malware are used to detect advanced persistent threats. In the proposed model, different algorithms may be used to compute effects of each property. Note that determining numerical values of the effects are out of scope of this paper. Some properties of advanced malware that are used to determine the effects of advanced persistent threats carried out with advanced malware are explained as follows:

- **Self-Destruction**: Advanced malware has self-destruction mechanisms to remove its traces on infected systems after accomplishing the defined tasks. Investigations show that each advanced malware has a self-destruct module known as a suicide mechanism. Many antimalware developers know this fact, such as Kaspersky, Symantec, and McAfee. Therefore, we expect that this feature will help to detect APTs accomplished with advanced malware in communication systems.

- **Self-Replication**: One of the most significant feature of advanced malware is the self-replication property if conditions hold. This property is also an important feature of worms. Therefore, some instances of advanced malware are classified as worm, such as Stuxnet. In conventional malware detection systems, self-replication property is used to distinguish worm and Trojan Horse. Thus, this is another significant property that is used to detect APTs carried out with advanced malware.

- **Self-Propagation**: Advanced malicious software has many methods to propagate [17]. For this reason, identifying the propagation method of advanced malware may be used to limit its effects on infected systems and specifically effects of advanced malware on the targeted system. This property is very significant for intrusion prevention systems.

- **Hiding Itself**: Existing instances of advanced malware in the wild show that malware can hide its traces on infected systems. For instance, advanced malware may load malware payload into mouse gestures or it may load a kernel-mode driver that hooks the kernel-mode handler for queries of system process information. Most of the time, advanced malware uses rootkit properties to hide itself. The proposed model uses this property to determine the type of malware.

- **Command and Control**: Advanced malware has an owner who may change its goal and its attack vectors like botnets. Therefore, it is a very significant issue to find the command and control center(s) of advanced malware to prevent advanced persistent threats.

- **Infection Methods**: Advanced malware may have many infection methods. For example, Stuxnet typically injects the entire DLL into another process and then it just calls the particular export. Stuxnet can inject itself into an existing or newly created arbitrary process or it can infect a predefined trusted process. During the injection process, Stuxnet may keep the injected code in the trusted process or it may instruct the trusted process to inject the code into another running process. The trusted process may consist of a set of default Windows processes and a variety of security products. Some trusted processes are Kaspersky KAV (avp.exe), MacAfee (Mcshield.exe), AntiVir (avguard.exe), BitDefender (bdagent.exe), Etrust (UmxCfg.exe), F-Secure (fsdfwd.exe), Symantec (rtvscan.exe), Symantec Common Client (ccSvcHst.exe), Eset NOD32 (ekrn.exe), Trend Pc-Cillin (tmpproxy.exe). The proposed model uses this property to identify advanced malware.

These properties are determined according to available information about advanced malware that is known in the wild. The model may be updated if new properties about advanced malware are discovered.

3. FD-APT: A Flow Model to Detect Advanced Persistent Threats

This section contains a flow-based approach to detect advanced persistent threats and a corresponding model to extend detections of APTs. The flow indicates information flow related to advanced malware and attacks. The model takes information about advanced malware from antimalware systems. Intrusion detection systems provides attack information to the model. Therefore, an FD-APT implementation may be an add-on for intrusion detection systems or intrusion prevention systems.

The proposed approach takes into account distinguishing properties of advanced malware. The properties are analyzed systematically to detect advanced persistent threats. A new flow model to detect APTs based on the new approach is presented to show the applicability of the approach, namely Flow based Detection of Advanced Persistent Threats (FD-APT). The model evaluates advanced persistent threats according to the properties of advanced malware within multiple levels. Figure 1 shows an instance for the structure of the flow model.

FD-APT has six levels that are used to evaluate some properties of advanced malware related to a specific advanced persistent threat. This structure represents relationships among properties of advanced malware. Each level increases the detection probability of an attack more precisely. In the model, there are two information sources. The first one is malware analysis tools and the other one is attack data obtained from network sensors on communication systems.

Note that there is no complete data about advanced malware because of limited available information in the literature and a small number of advanced malware instances in the wild. Therefore, classical antimalware systems are unable to detect and to provide adequate amount of real data about advanced malware. Similarly, existing network sensors, which may run under the control of intrusion detection systems, do not supply attack data about advanced persistent threats that are carried out with advanced malware. Moreover, attack data reveal sensitive information about targeted systems, therefore owners of the systems under attack do not reveal attack data if they have. In this paper, the flow based approach may be extended with new discoveries about advanced malware. Furthermore, the approach may be used to discover new properties of advanced malware and related advanced persistent threats. On the other hand, FD-APT is designed according to available information about advanced malware and the flow based approach. Since we have no complete information about advanced malware and corresponding APTs, FD-APT is analyzed with synthetic data.

Figure 1 is an example for the construction of FD-APT model. The detailed explanation for each level of FD-APT is as follows:

- ***Level 0*** identifies known advanced malware. Additionally, this level determines suspicious behaviors related to an attack. Level 0 may run anomaly detection algorithms to determine such behaviors. A suspicious behavior is a behavior, where a system behaves abnormally or it is an unexpected behavior. More precisely, a suspicious behavior is related to attack information and malware information. Intrusion detection algorithms, machine learning algorithms, or newly created specific algorithms may be used to determine suspicious behaviors.

- ***Level 1*** is related to advanced malware properties. This level identifies properties of advanced malware directly, such as written language, data separation and size of code, self-destruction, and self-replication properties.

- ***Level 2*** determines the complexity of malware and the complexity of an advanced persistent threat. Therefore, level 2 considers properties related to the construction of an attack and the behavior of malware, such as an encryption method and the complexity of an APT.

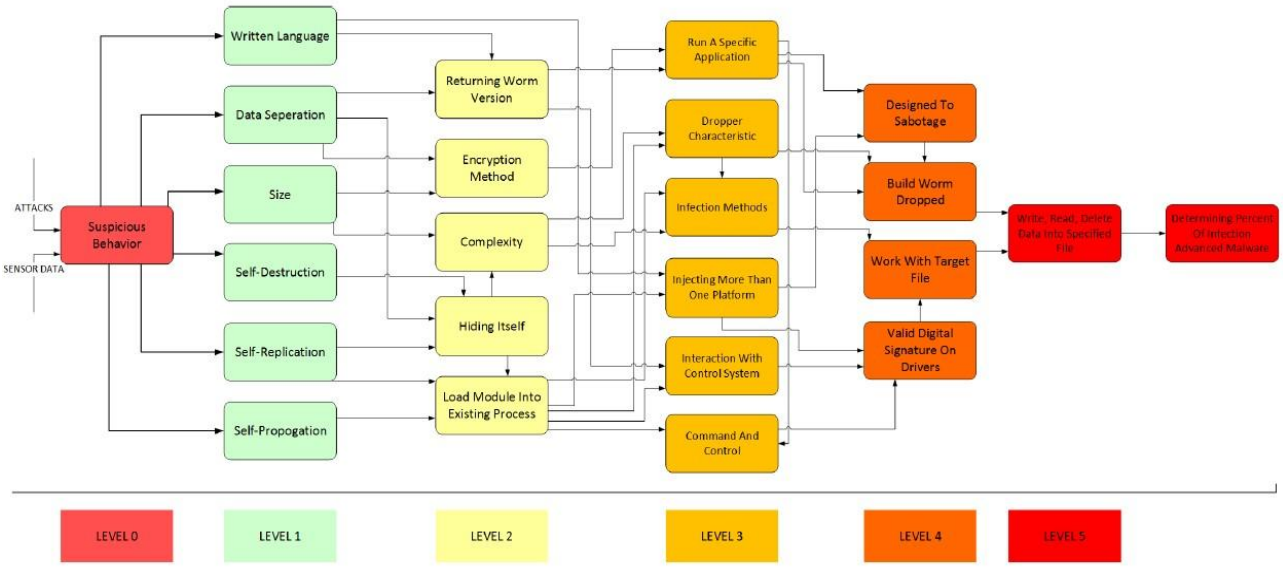

Figure 1. An example for the construction of the flow model to detect APTs [18].

- ***Level 3*** identifies payload properties of advanced malware. Therefore, properties in this level are the most connected ones in FD-APT model. For instance, "Run a Specific Application" property has five directly related features.

- ***Level 4*** is devoted to determine the type of an attack. Specifically, level 4 investigates modular properties of malware and its unique features. For example, the use of stolen digital signatures during an attack is one of these features.

- ***Level 5*** deals with the ultimate goal of an attack that may be designed to accomplish a sabotage like Stuxnet or espionage like Duqu. Therefore, dynamic decision algorithms may be used in this level. Specific algorithms may be designed to determine the ultimate goal or existing decision making algorithms may be used directly, however, decision algorithms that may be used in level 5 are out of scope of this paper.

In FD-APT, each level produces a value that has interval between zero and one, where zero means no risk related to an advanced persistent threat whereas one means the highest risk of an APT. Moreover, it may represent that advanced malware is a part of an APT. The risk of being a part of an APT related to advanced malware is proportional with the value. In the proposed model, the risk is a probability of being an APT related to specific advanced malware. In this paper, the risk of a level is represented with Ψ_L, where $L \in \{0, 1, 2, 3, 4, 5\}$ represents a level. Moreover, each property of advanced malware has a risk that is represented with ψ_p, where $p \in P$. P represents the set of properties of advanced malware as in equation (1).

$$P = \{Properties\ of\ advanced\ malware\} \qquad (1)$$

The risk value is inadequate to determine an attack related to an APT and corresponding advanced malware. FD-APT contains a threat level (threshold) to be able to determine an advanced persistent threat with advanced malware. An acceptable threat level depends on the type of a communication system and sensitive information that flows on the system. Therefore, each system may have different threshold levels. In FD-APT, the threshold is represented with τ, where $0 \leq \tau \leq 1$.

FD-APT model uses some ψ_{PS} to determine Ψ_L. The model has two types of functions to compute ψ_{PS}. For *Level 0*, FD-APT computes ψ_{PS} according to attacks on the system, conventional malware data, and advanced malware data. Intrusion detection sensors and anti-malware systems provide attack data and malware data, which are located on nodes over communication systems.

All attack data and all malware data are represented with vectors $\vec{A}$ and $\vec{M}$, respectively. Attack data are represented with α_p, where FD-APT computes α_p, $\forall p \in P$. α_p is an entry of $\vec{A}$, called the attack vector. In FD-APT, each attack has a correlation with properties of advanced malware. On the other hand, sensors data are represented with μ_p, where FD-APT computes μ_p, $\forall p \in P$. μ_p is an entry of $\vec{M}$, which is a property of malware vector related to sensors data. FD-APT computes the risk according to formula (2) for *Level 0*, where the output of the function is a vector, $\overrightarrow{\Psi 0}$. The entries of $\overrightarrow{\Psi 0}$ are ψ_{PS}.

$$\overrightarrow{\Psi 0} = \Lambda_0(\vec{A}, \vec{M}) \qquad (2)$$

$\overrightarrow{\Psi 0}$, $\vec{M}$, and $\vec{A}$ have the same size and all entries of the vectors have values between zero and one. Moreover, each entry of $\overrightarrow{\Psi 0}$ has different computation function, which is represented with λ_x, $\forall x \in P$.

$$\Lambda_0(\vec{A}, \vec{M}) = \bigcup_{\forall x \in P} \lambda_x(\vec{A}, \vec{M}) \quad (3)$$

An attack and sensors data about advanced malware may have different weights to compute λ_x in each system. Determining these weights is beyond the scope of this work. On the other hand, λ_x is a *weighted inner product* of $\vec{A}$ and $\vec{M}$.

$$\lambda_x(\vec{A}, \vec{M}) = \sum_{\forall y \in P} \omega_{x,y} \alpha_y \mu_y \quad (4)$$

Thus, the risk of Level 0 is computed as follows:

$$\Psi_0 = \|\vec{\Psi 0}\| \quad (5)$$

In FD-APT, the risks in all levels except Level 0 are computed with Ψ_Z according to data in $\vec{A}$, $\vec{M}$, and $\vec{\Psi_z}$, where $z \in \{P - \{0\}\}$.

$$\vec{\Psi z} = \Lambda_z(\vec{A}, \vec{M}, \vec{\Psi 0}, \ldots, \overrightarrow{\Psi z - 1}) \quad (6)$$

$$\Lambda_z(\vec{A}, \vec{M}, \vec{\Psi 0}, \ldots, \overrightarrow{\Psi z - 1}) = \bigcup_{\forall x \in P} \lambda_x(.) \quad (7)$$

$$\lambda_x(.) = \lambda_x(\vec{A}, \vec{M}, \vec{\Psi 0}, \ldots, \overrightarrow{\Psi z - 1}) \quad (8)$$

$$= \sum_{\forall y \in P \wedge q} (\omega_{x,y,\alpha} \alpha_y + \omega_{x,y,\mu} \mu_y + \sum_{\forall q} \psi_{x,q}) \quad (9)$$

where $q \in \{0, \ldots, z-1\}$.

$$\Psi z = \|\vec{\Psi z}\| \quad (10)$$

One of the main difference of the proposed flow model is that attack data and sensors' data related to advanced malware are evaluated differently. The goal of this evaluation is to catch a potential advanced persistent threat that is used to determine the risk of a communication system related to advanced persistent threat. Therefore, risk vectors of each level are orthogonal to each other, which means $\vec{\Psi x} \perp \vec{\Psi y}$, where $x, y \in P \wedge x \neq y$. Note that two vectors u and v are orthogonal in an inner product space if

$$< u, v >= 0 \quad (11)$$

However, two vectors are not orthogonal with respect to the *weighted Euclidean inner* product. Therefore, determining the weights is crucially important in FD-APT. Otherwise, the risk vectors in each level will not be orthogonal, which means FD-APT model may not distinguish APTs carried out with advanced malware.

In, FD-APT, a potential APT and an attack are determined according to the risk level of *Level 5*.

4. Analysis of FD-APT

FD-APT model is analyzed according to the proposed approach. Actually, many other flow models may comply with the new flow approach but they need to represent significant properties of advanced malware described in this paper. The application in [18] is developed according to the proposed approach for analysis purposes. The main goal of this analysis is to show the applicability of the proposed approach related to the detection of advanced persistent threats. The analyses contain two cases, which are no attack case and an attack with advanced malware case.

Since there is no complete data set about attacks with advanced malware, these analyses aim to show potential APTs carried out with advanced malware. This will ensure us to develop countermeasures before APTs carried out with advanced malware become persistent. Thus, determining the effectiveness of FD-APT is beyond the scope of these analyses.

4.1. Data set and assumptions

Advanced persistent threats may be carried out by using conventional malware, advanced malware, or attack tools. In this paper, attacks with conventional malware and attack tools are not advanced persistent threats. Specifically, FD-APT model considers only APTs carried out only with advanced malware.

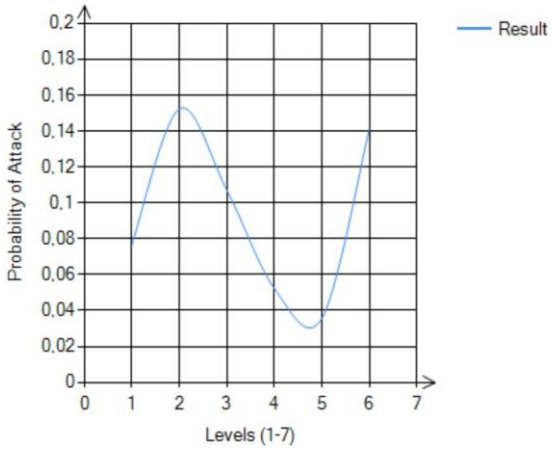

Figure 2. Probability of attacks in each level with the absence of advanced malware [18].

In FD-APT, the threat level, τ, depends highly on the targeted system. For instance, if a targeted system contains many services and if it has many potential vulnerabilities, the value of the threat level should be close to one. Otherwise, FD-APT may be unable to distinguish APTs and other attacks on the targeted system. Therefore, tuning the threat level depends on targeted system, which makes FD-APT model adaptive. In this paper, the value of the threat threshold is intentionally selected to show the applicability of FD-APT without taking into account any specific targeted system.

FD-APT model requires data from two sources, which are attack data about APTs and sensor data related to advanced malware obtained from communication networks. In literature, data from these resources are very limited and they are incomplete. Moreover, most of data are unavailable to the public. In this paper, synthetic data are generated and they are used to analyze FD-APT model. The data set is generated as follows:

- For no attack case, both attack data and sensors' data are randomly generated and they are uniformly distributed with mean 0.1.
- For attack with advanced malware case, both attack data and sensors' data are randomly generated and they are uniformly distributed with mean 0.82.
- Both attack data and sensors' data are obtained for each second.

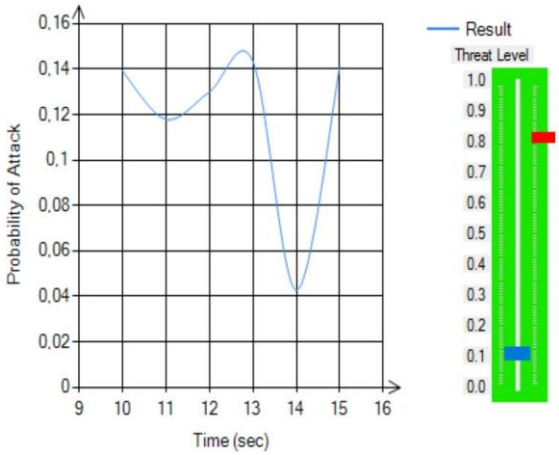

Figure 3. A continuous detection of an attack when there is no advanced malware in the environment [18].

4.2. Case: no attack with advanced malware

In this case, there is no attack on communication systems with advanced malware but there may be attacks with conventional malware. Therefore, the attack probability generated by each level may differ from each other, which means that this is not an APT with advanced malware. Figure 2 shows the probability of attacks in each level computed with artificial data representing sensors data and attack data. The results show that the maximum attack probability is 0.15, which is well below the possible APT level. In this case, 0.8 is the threshold that distinguishes advanced persistent threats from other attacks. Thus, the probability of an APT is low in this case. Moreover, conventional malware may be a part of this attack but advanced malware is not expected to participate to the attack.

Figure 2 shows the results about an infected file with conventional malware, which is analyzed with the application. The file is considered to traverse many communication networks and sensors on the networks gather information about the file. The results show that the probability of an attack does not exceed 0.15 at each level. Therefore, the probability of having advanced malware related to an APT is low. This figure shows an attack probability for a specific time only.

Figure 3 contains a continuous detection process related to the infected file with conventional malware, where the detection threshold is 0.8. In Figure 3, threat level represents the risk that is computed as the average of probabilities of attacks at all levels. If the average of the attack probability is greater than the threat threshold, this means that there is an advanced persistent threat related to the file. Otherwise, it means that there is no APT related to the file. For example, all results between 10 and 15 seconds are below the threshold for this case. Note that determining the threat threshold is beyond the scope of this paper.

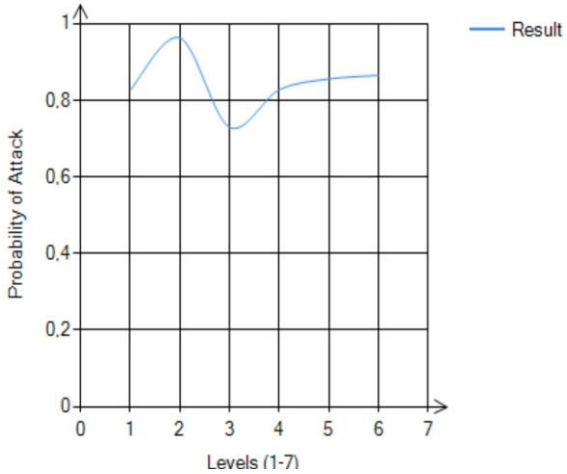

Figure 4. The probability of attacks in each level when the environment contains advanced malware [18].

This case shows that FD-APT model is able to distinguish attacks with conventional malware and advanced malware. Additionally, the case study shows that FD-APT model may be used to analyze an infected file to determine the type of malware. This information may be used to feed classical antimalware systems.

4.3. Case: an attack with advanced malware

In this case, there is an advanced persistent threat carried out with advanced malware. This means that the file is infected with advanced malware. The goal of this case study is to show how FD-APT model detects both APTs and advanced malware.

Figure 4 shows attack probabilities of advanced malware according to FD-APT model in each level. Particularly, level 3 has a lower attack probability, where the threshold is 0.8. The probability of an attack reaches its maximum at level 2 but the risk decreases sharply at the next level. Then, the

probability of an attack increases again above the threshold and it never decreases below the threshold. These results show that FD-APT model may detect APTs since the model takes into account properties of advanced malware related to APTs during different stages of attacks. Moreover, the results show that if advanced malware properties are inadequately determined then an antimalware system may not detect APTs. For example, if the antimalware system considers only malware properties in Level 3, it may not detect advanced malware and corresponding attacks, advanced persistent threats.

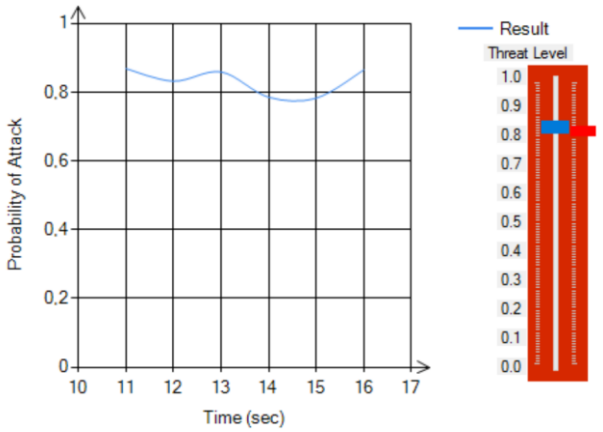

Figure 5. A continuous detection of an attack when there is advanced malware in the environment [18].

An application that implement FD-APT model may run continuously to detect APTs carried out with advanced malware. The detection process when a file is infected with an advanced malware is presented in Figure 5. Results in the figure shows that FD-APT almost always detects advanced persistent threats. However, the model may be tuned to detect APTs more precisely since the threat level is determined below the threshold level between 14 and 15 seconds. Specifically, almost all probabilities of attack results are above or near the detection threshold between 11 and 16 seconds. This means that the system is under attack with advanced malware in this case.

Results in the both cases show that FD-APT model may be used to detect advanced persistent threats that are carried out with advanced malware. Specifically, FD-APT model takes malware properties and attack data on communication networks for detection purposes. Moreover, FD-APT may be implemented as an add-on to classical intrusion detection systems to enhance to the detection probability of advanced persistent threats that are based on advanced malware.

5. Conclusion and Future Work

Advanced persistent threats have become more pervasive with the increased connectivity of communication systems on the Internet. Rapid developments of advanced malware have complicated the detection of attacks based on APTs. This circumstance increases the cost of building secure and trusted cyber space via communication systems.

This paper is about the detection of advanced persistent threats that are carried out with advanced malware. Particularly, the focus of this works is APTs and the identification of advanced malware. The work has two main contributions. The first one is extracting correlations between APTs and advanced malware, which correlations are used to define a flow-based detection approach to detect APTs. The other contributions is a detection model of advanced persistent threats that model uses the flow based approach. The model is expected to help improving classical intrusion detection systems and antimalware systems to detect advanced malware based APTs. Moreover, the paper contains an implementation of FD-APT model that is used to analyze the proposed approach. The implementation is a proof of a potential applicability of the new approach. Furthermore, analyses results show that FD-APT model may increase the detection rate of advanced persistent threats. Thus, it will reduce the cost of securing cyber space.

Since there is no adequate information about advanced persistent threats that are carried out with advanced malware, the analyses were carried out with synthetic data. The author has been working to collect real data about advanced malware and related APTs to improve FD-APT as a future work.

Acknowledgment

The author would like to thank Ekrem Cihat Çetin and Fatih Deniz for their support to develop and implement the application for FD-APT. This work is supported by Istanbul Technical University under the BAP project, number MAB-2017-40642.

References

[1] Lagazio, M., Sherif, N., Cushman, M. 2014. A multi-level approach to understanding the impact of cyber crime on the financial sector. Computers & Security, 45, 58–74.

[2] Egele, M., Scholte, T., Kirda, E., Kruegel, C. 2012. A survey on automated dynamic malware-analysis techniques and tools. ACM Computing Surveys, 44(February 2012), 6:1–6:42.

[3] Mell, P. M., Kent, K., Nusbaum, J. 2005. Guide to Malware Incident Prevention and Handling., NIST SP, 800-83.

[4] Wood, P. 2016. Internet security threat report", Tech. rep., Symantec Corporation.

[5] Bahtiyar, Ş. 2016. Anatomy of targeted attacks with smart malware. Security and Communication Networks, 9(18), 6215 – 6226.

[6] Han, X., Tan, Q. 2010. Dynamical behavior of computer virus on Internet. Applied

Mathematics and Computation, 217(6), 2520–2526.

[7] Swain, B. 2009. What are malware, viruses, spyware, and cookies, and what differentiates them?. Symantec Tech. rep., 2009.

[8] Mishra, B. K., Pandey, S. K. 2011. Dynamic model of worms with vertical transmission in computer network. Applied Mathematics and Computation, 217(21), 8438–8446.

[9] Anonymous, 2017. Types of malware. Kaspersky Lab Technical Report. https://usa.kaspersky.com/internet-security-center/threats/types-of-malware#.WGQWiFOLTIV (Access: 13.12.2017)

[10] Anonymous, 2017. What is adware?. Kaspersky Lab Technical Report. https://usa.kaspersky.com/internet- security-center/threats/adware#.WGQsn1OLTIU. (Access: 13.12.2017)

[11] Li, Z., Goyal, A., Chen, Y., Paxson, V. 2011. Towards situational awareness of large-scale botnet probing events. IEEE Transactions on Information Forensics and Security, 6(1), 175–188.

[12] Kerr, P. K., Rollins, J., Theohary, C. A. 2010. The stuxnet computer worm: Harbinger of an emerging warfare capability. Technical Report.

[13] Bencsath, B., Pek, G., Gabor, L., Felegyhazi, M. 2011. Duqu: A stuxnet-like malware found in the wild. Technical Report.

[14] Anonymous, 2012. Stuxnet: Opening pandora's box?, https://cmu95752.wordpress.com/tag/stuxnet/, 2012 (Access: 24.02.2013)

[15] Combs, M. M. 2012. Impact of the stuxnet virus on industrial control systems. In XIII International forum Modern information society formation - problems, perspectives, innovation approaches, St.-Petersburg, RUSSIA, September 5–10.

[16] Anonymous, 2011. Duqu: The precursor to the next stuxnet, Symantec Technical Report. http://www.symantec.com/outbreak/?id=stuxnet (Access: 24.02.2013)

[17] Tangil, G. S., Tapiador, J. E., Lopez, P. P., Ribagorda, A. 2014. Evolution, Detection and Analysis of Malware for Smart Devices. IEEE Communications Surveys and Tutorials, 16(2), 961—987.

[18] Çetin, E. C. 2017. Identification and Automated Classification of Advanced Malware. BS Graduation Project, Istanbul Technical University, Department of Computer Engineering, İstanbul.