Decepticon: A Hidden Markov Model Approach to Counter Advanced Persistent Threats

Rudra Prasad Baksi(✉) and Shambhu J. Upadhyaya

University at Buffalo, SUNY, Buffalo, NY 14260, USA
{rudrapra,shambhu}@buffalo.edu

Abstract. Deception has been proposed in the literature as an effective defense mechanism to address Advanced Persistent Threats (APT). However, administering deception in a cost-effective manner requires a good understanding of the attack landscape. The attacks mounted by APT groups are highly diverse and sophisticated in nature and can render traditional signature based intrusion detection systems useless. This necessitates the development of behavior oriented defense mechanisms. In this paper, we develop Decepticon (Deception-based countermeasure) a Hidden Markov Model based framework where the indicators of compromise (IoC) are used as the observable features to aid in detection. This framework would help in selecting an appropriate deception script when faced with APTs or other similar malware and trigger an appropriate defensive response. The effectiveness of the model and the associated framework is demonstrated by considering ransomware as the offending APT in a networked system.

Keywords: Advanced Persistent Threats (APT) · Computer security · Cyber-security · Hidden Markov Model (HMM) · Ransomware

1 Introduction

Advanced Persistent Threats (APT) are a form of quiet invaders [20] and are a big nuisance to industries and government organizations. They silently perform reconnaissance, quietly invade, and keep a communication channel open in order to communicate with the command and control (C&C) centers. The attackers control the behavior of the malware from the C&C centers. APTs carry out *targeted attacks* to achieve their goal. They are quite persistent in their efforts of achieving the goals and in doing so they might come with a contingency plan to which they resort to upon discovery [2]. Such type of attacks has become prevalent and frequent, owing to the fact that malware-as-a-service (MaaS) are easily available, which provide the attackers with the necessary framework and infrastructure to create attacks [14,22]. APTs come in different forms and formats. In this paper we focus on the mitigation of ransomware that qualifies as an APT [2].

S. K. Sahay et al. (Eds.): SKM 2019, CCIS 1186, pp. 38–54, 2020.
https://doi.org/10.1007/978-981-15-3817-9_3

According to FireEye, 4,192 attacks were detected in a particular year, which were mounted by groups that can confidently be classified as APT groups [4]. They were also able to detect 17,995 different infections by APT groups. The attacks thereafter have been increasing by leaps-and-bounds. RSA Security LLC suffered financial losses of about $66.3 Million when it became a victim of an APT attack [31]. According to a study by Ponemon Institute, the average financial losses suffered by a company owing to the damaged reputation after an APT often amounts to $9.4 Million [16]. WannaCry, Petya and NotPetya are ransomware campaigns that graduated to become APTs and wreaked havoc and collected huge amounts of ransom causing considerable financial losses to the victims [2]. WannaCry collected ransom in BitCoins. According to certain reports, between May 12, 2017 and May 17, 2017, the attackers collected $75,000 to $80,000 in ransoms [7,28]. With time the cost of financial damage suffered by the companies is expected to go higher up. Both industries and government organizations are known to suffer significantly. In case of government agencies, the damage could be beyond mere financial losses; the attacks might even threaten national security.

These aforementioned factors and incidents outline a great threat to the critical infrastructure as a whole, be it government or industry. The problems are intense and the attacks are adaptive in nature, requiring a holistic approach to address them. On the contrary, it is not necessary to put the entire defense framework into the same defense mode every time the system comes under attack because deploying a sophisticated defense mechanism indiscreetly to fend off attacks will severely affect performance and degrade the quality of service (QoS). The idea is to deploy the most sophisticated countermeasure against the most severe form of attack. Lesser sophisticated countermeasures taking care of the less severe attacks would not only be economical but also might help in preserving the quality of service (QoS) of the system. In the same vein, system security through different forms of information isolation has been studied for quite sometime [19]. Isolation can be achieved through software or hardware [18]. But with advanced attacks from APT groups which are highly adaptive in nature, they have been successful in attacking physically isolated systems as well. One such example is the Stuxnet campaign that took place in the Iranian nuclear facility [3,9,13]. Therefore, a need for a new form of defensive strategy arose. Researchers have looked into various approaches to repel highly sophisticated attacks. One of the approaches is the use of *deception* as a defense tool.

In this paper, the aforementioned research ideas, namely, *isolation* and *deception*, are used to confront intricate attacks arising from APT groups. The paper puts forward a basic architecture, which deceives the attacker into believing in its success, while surreptitiously triggering a fix to thwart the attack. To make the defense-system cost-effective, the defender must have knowledge about the attack scenario. The information about the status of a malware helps a defender to develop an efficient attack averting strategy. This paper presents Decepticon, a Hidden Markov Model (HMM) based deceptive countermeasure which uses indicators of compromise (IoC) that will serve as observable features for detection

and mitigation of APTs. The major contributions of the paper are the design of a hardware-based defense framework and a HMM-based ransomware type APT detection tool. The framework is a special case of the Kidemonas architecture [1] and uses the concept of smart-box from [21] for surreptitious reporting and triggering of defensive scripts on being attacked. The paper is organized as follows. Section 2 discusses some related work in this area. Section 3 presents the new deception architecture. Section 4 describes the HMM based detection system. Section 5 discusses the architecture's usage in detection and mitigation of APTs. Finally, Sect. 6 concludes the paper and paves way for future work.

2 Preliminaries and Related Work

In this section, some preliminaries are given on malware, APT, TPM hardware, deception and HMM, which are used to develop the Decepticon architecture in Sect. 3. Related work on these topics is also briefly reviewed.

Malware created by the APT groups do not carry out the attacks in a single stage. The "Cyber Kill Chain" framework developed by Lockheed Martin describes an APT through a seven stage life cycle [11]. The model describes the beginning of the attack through a *reconnaissance* phase wherein the malware gathers information about the system. This is followed by the *weaponization* phase, thereupon creating a remote access malware that can be controlled by the attacker. The *delivery* phase denotes the intrusion of the malware into the system. In the *exploitation* phase, the malware exploits the vulnerabilities that exist in the system. The *installation* phase signifies the escalation of privileges on the part of the malware and installation of back-doors to maintain a communication with the command and control (C&C) centers to receive further instructions. The *command and control* phase implies the access of the target system gained by the attackers from the C&C centers. Finally, in the *actions on objective* phase, the intruder mounts the final assault on the system. LogRythm describes an APT through a five stage life cycle [17]. Lancaster University describes APT through a three stage life cycle [26]. Baksi and Upadhyaya [2] describe APT through the characteristics exhibited by a sophisticated malware.

Ransomware are a type of malware which infiltrate a system and hold critical data for a ransom. Primarily there are three simpler types of ransomware, namely *the locker*, *the crypto* and *the hybrid* [32]. The locker variant of the ransomware locks the entire system and denies the user access to the system. The crypto form of the malware, targets specific files and/or folders and encrypts them, thereby denying the user any access to those encrypted resources. The hybrid version of ransomware possesses the capabilities of both types of ransomware. It can encrypt and lock targeted resources and/or the entire system. But the ransomware under consideration in this paper is a more advanced form of malware. In addition to possessing the features of ransomware, they are more sophisticated to have a contingency plan of attack on being discovered [2]. They also perform the attack through multiple stages and generally are controlled by the attackers from the C&C centers. They qualify as APTs.

The TPM or the Trusted Platform Module is a hardware component designed following the guidelines of the security consortium, the Trusted Computing Group [30]. The TPM comes with essential cryptographic potential. It can generate cryptographic keys, both symmetric and asymmetric keys. It also has the capability of generating random numbers when required and can store cryptographic credentials. It also provides hashing capabilities. The primary functionalities of TPM include verification of platform integrity, safeguarding encryption keys, and preservation of password and user credentials. Figure 1 gives a simplified schematic of the TPM version 1.2 specifications of which are laid down by the Trusted Computing Group (TCG). TPMs today come in different incarnations which depends on the type of device and the manufacturer. Intel Software Guard Extension (Intel SGX) and ARM TrustZone are versions of TPM like hardware components which come with certain functionalities in addition to the ones already mentioned for TPMs [8,12,29,33]. They provide a Trusted Execution Environment, which are generally outside the purview of high-priority OS instructions but can be accessed using the user credentials. Therefore, in general it can be assumed, even if the OS is compromised, that the hardware component is outside the purview of the attacker.

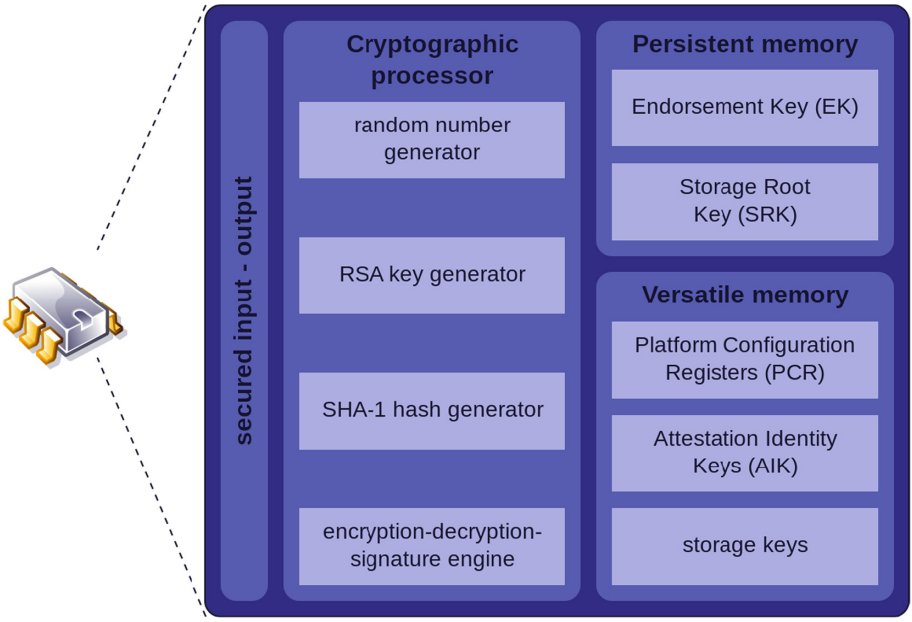

Fig. 1. A simplified schema of TPM [24]

Deception can often be considered as a potential weapon against sophisticated attacks and it is an important area of research. In [5], the authors use deception as a potential weapon to fight against denial of service (DoS) attacks.

The authors have analyzed the deceptive strategy using a game-theoretic model based on the signaling game with perfect Bayesian equilibrium (PBE) to investigate the implications of deception to counter the attacks. Deception as a defensive strategy has been used in [23], wherein the authors have used deceptive measures to lure the attackers to high-interaction honeypots for designing a malware detection system.

Hidden Markov Models (HMM) have been historically been used for speech recognition [15,25]. It has also been applied for handwritten character and word recognition [6]. The biggest advantage that comes with HMM is that, in a process wherein the stages are not visible to the observer, certain observable features can be used to predict the stage of the process at a certain instance. Owing to this advantage, HMM-based techniques have often been used for the analysis of sophisticated malware. Metamorphic virus can be an annoyance. A metamorphic virus is capable of changing its code and become a new variant of itself without changing the functionalities. The changes are not exactly visible to the observer and therefore observable characteristics play an important role in the analysis. HMM has been used for detection and analysis of such metamorphic viruses [27].

3 The Architecture

The Trusted Computing Group (TCG) laid down the specifications for Trusted Platform Module (TPM) with an idea of creating a trusted computing environment [30]. These specifications were capitalized on to create a deception based architecture, Kidemonas [1], which provides isolation to malware detection systems so that the detection can occur outside the purview of the attacker and the intrusion can be surreptitiously reported to the user or the system administrator.

In this paper the capabilities of Kidemonas are extended to realize a cost-effective system to detect intrusions from advanced persistent threats. In a business enterprise type environment, Kidemonas gives the system administrator the capability to run different forms of intrusion detection on different computing units, and the information regarding intrusion is shared with the system administrator and the other computing units through a separate channel as shown in [1]. This is called the peer communication network comprising of a link-layer communicating unit present on each computing unit called the peer communication unit (PCU). A computing unit in this scenario refers to a computer or a server or basically any computing unit which forms a node in the networked system in a corporate network monitored by a single user or a single group of users working collectively for the same purpose. To make the defense strategies cost-effective, we use the smart-box proposed in [21]. The idea is whenever a form of intrusion is detected, it is reported to the system administrator silently, who in turn uses the smart-box to trigger an appropriate defensive response from the repository. The repository is a storage unit for defense strategies that could be triggered to defend the system at the event of an attack on the system. The defense strategies range from simply blocking certain processes to defending against intricate attacks. The smart-box on learning from the nature of the attack and the status

of the malware can trigger an effective response which would be economical in terms of time and resources being used. Smart-box is the decision making unit regarding defensive strategies depending upon the characteristics of the malware.

Figure 2 represents the hardware based defense architecture, and we call it Decepticon. The aim is to deceive ransomware type APTs. Kidemonas [1] is a more generic architecture to counter any APT, Decepticon is a customized version to have a HMM based ransomware detection tool (which is subsumed under the Enclave in the figure and discussed in the next section), and a smart-box to trigger defensive actions depending upon the severity of the attack. If the attack is determined to be of a simple nature, the smart-box triggers a simple response to counter it, and if the attack is sophisticated in nature, then it triggers an elaborate response.

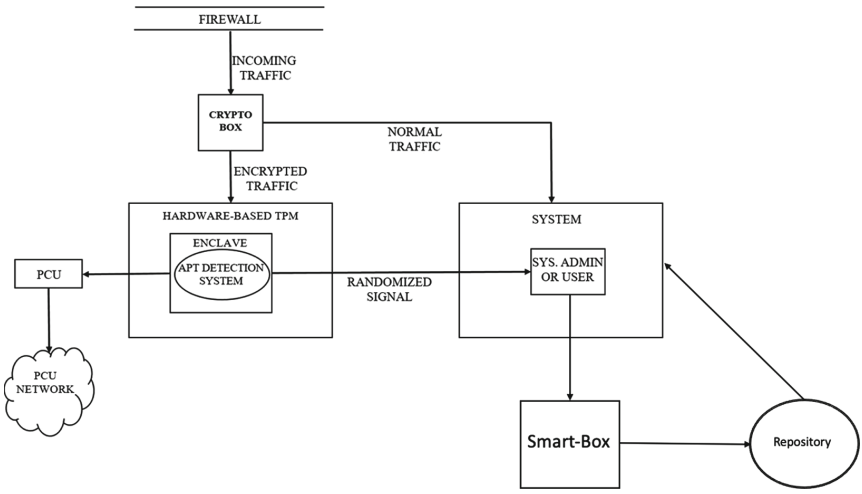

Fig. 2. Decepticon architecture

The firewall (Fig. 2) performs signature based detection. If the malware is able to get past the firewall along with the legitimate traffic, it reaches the crypto-box. The crypto-box makes a copy of the incoming traffic and sends the normal traffic to the system. The copied traffic is encrypted and sent to the hardware-based TPM. The encryption is performed using the public-key of the endorsement key of the hardware-based TPM. In the TPM, the ciphertext is decrypted using the private-key component of the endorsement key of the TPM. The analysis of the traffic is done by the HMM based detection tool. Any form of intrusion being detected is sent to the peer communication unit (PCU) and from there to the PCU network, so as to inform every node in the networked system about the form of intrusion. The PCU network is accessible only through the PCU, which in turn is accessible through the hardware-based TPM. At the same time, a surreptitious reporting is done to the user or the system administrator.

The system administrator then uses his/her storage root keys (SRK) to gain access to the TPM to gain knowledge and the nature of the intrusion that has taken place.

The security of the entire system relies on the fact that the private key component of the endorsement key of the TPM, which was created when it was manufactured, never leaves the TPM. The security also relies on the fact that the storage root keys (SRK) created by the user, when he/she took the ownership of the TPM, is kept safely guarded.

Figure 3 shows a snapshot of a networked system in a corporate network. This representation shows multiple computing units connected to a single access point. Each computing unit is connected to other computing units through the PCU network, which is also used to inform each other of any form of intrusion in the system. Figure 3 shows different versions of detection tools running on different computing units; some of them running Decepticon while others are running the generic Kidemonas style APT detection tools.

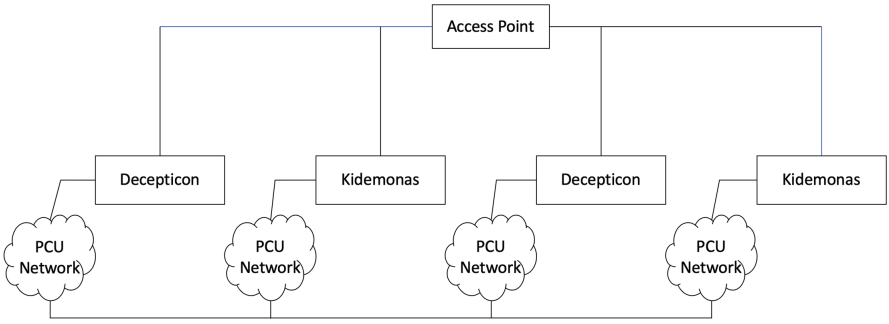

Fig. 3. The system

4 The Ransomware Detection Model

The threat model under consideration in this paper primarily deals with ransomware which qualify as advanced persistent threats. This means that the attack mounted would be highly sophisticated and persistent in nature. Such attacks can render the traditional signature based intrusion detection systems useless. To deal with APTs that have no prior history, behavior-oriented defense systems are a necessity. APTs are generally mounted in multiple stages unlike the more common threats. The knowledge of the stage in which an APT is currently in, is a utilitarian information for the defender to make an informed decision about the defense strategy. These attacks are mounted by the quiet invaders [20] and they subtly graduate through different stages. Therefore, the difficulty arises in figuring out the status of the malware. One can look into the behavioral changes and using those as observable can help make an informed

decision. To help the defender in making that informed decision, we develop a Hidden Markov Model (HMM) based intrusion detection tool. This tool will help the defender discern the status of the malware with certain probability, which would define its confidence in choosing the defensive action.

The proposed HMM has N number of hidden states and M number of observables. The model can be denoted by $\lambda = (A, B, \pi)$, where

- A is an $N \times N$ matrix that gives the transition probabilities, characterizing the transition of each hidden state to another. Hence, it is called the transition matrix.
- B is an $N \times M$ that gives the emission probabilities for each hidden state. Hence, it is called the emission probability matrix.
- π is a $1 \times N$ matrix that contains the initial probability distribution for each of the hidden states.

This detection model strictly deals with ransomware. It intends to figure out whether a malware is a ransomware or not, and if it is a ransomware then is it a ransomware that has graduated to become an APT. Moreover, the model also investigates that if the ransomware is an APT then is it still pursuing its attack as a ransomware or would resort to a contingency plan of attack. Taking all these into consideration we formulate the model using the following parameters:

- The value of N is 4 which denotes that there are 4 hidden states being considered in this model $Z = \{z_1, z_2, z_3, z_4\}$
- The value of M is 5 which denotes that the number of observable random variables is 5, stated by $X = \{x_1, x_2, x_3, x_4, x_5\}$
- α_{ij} denotes the transition probability of the malware from i^{th} latent state to j^{th} latent state, where $i \in \{1, 4\}$ and $j \in \{1, 4\}$
- β_{ir} denotes the emission probability of i^{th} latent state manifesting r^{th} observable behavior, where $i \in \{1, 4\}$ and $r \in \{1, 5\}$

The hidden or latent states of the malware are as follows:

- The first state z_1 is where it is just a malware, regardless of the fact whichever form of malware it graduates to.
- The second state z_2 is where the malware becomes a ransomware.
- The third stage z_3 is the one wherein the ransomware has graduated to become an APT.
- The fourth and the final hidden state in this model is denoted by z_4, wherein the attacker chooses to execute the contingency plan of attack instead of mounting a ransomware attack on the victim. This is an important stage, wherein a ransomware, which has graduated to become an APT, is choosing to execute a contingency plan of attack.

The hidden states of the malware are often outside the purview of the defender's intrusion detection system and hence, the term hidden state, which entailed the use of Hidden Markov Model based intrusion detection model for ransomware. For the model, as discussed earlier, the observable behavioral states

are used to ascertain the status of the malware. The set of observable states is given by $X = \{x_1, x_2, x_3, x_4, x_5\}$. Following are the details regarding individual observable state used to design the model:

- x_1: Reconnaissance
- x_2: Interaction with honeypots or real-databases which are of high value
- x_3: Backdoor implants and/or back-channel traffic
- x_4: If the strategy of "Campaign Abort" exists
- x_5: Existence of any other contingency plan of attack

Figure 4 shows the HMM based ransomware detection model. Now that we have our latent and observable stages, and the associated parameters, we can determine the transition probability matrix A and the emission probability matrix B. With the aforementioned parameters we have the following:

$$A = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} \\ \alpha_{41} & \alpha_{42} & \alpha_{43} & \alpha_{44} \end{bmatrix}$$

$$B = \begin{bmatrix} \beta_{11} & \beta_{12} & \beta_{13} & \beta_{14} & \beta_{15} \\ \beta_{21} & \beta_{22} & \beta_{23} & \beta_{24} & \beta_{25} \\ \beta_{31} & \beta_{32} & \beta_{33} & \beta_{34} & \beta_{35} \\ \beta_{41} & \beta_{42} & \beta_{43} & \beta_{44} & \beta_{45} \end{bmatrix}$$

Transition Probability:

$$T(ij) = p(z_{k+1} = j | z_k = i)$$

where $i \in \{1, 2, 3, 4\}$ and $j \in \{1, 2, 3, 4\}$

Emission Probability:

$$\varepsilon_i(x) = p(X_k = x | z_k = i)$$

where $\varepsilon_i(x)$ is the probability distribution on X and $i \in \{1, 2, 3, 4\}$

Initial Probability Distribution:

$$\pi(i) = p(z_1 = i)$$

where $i \in \{1, 2, 3, 4\}$

The joint probability distribution is given by:

$$p(z_1, ..., z_4, x_1, ..., x_5) = \pi(1) \prod_{k=1}^{3} T(z_{k+1} | z_k) \prod_{n=1}^{5} \varepsilon_z(x_n)$$

The transition probabilities considered for this paper are updated as in the following matrix:

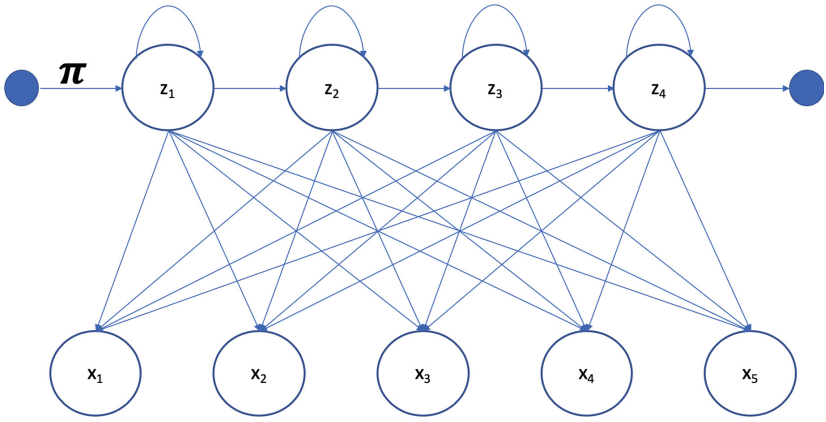

Fig. 4. HMM based ransomware detection model

$$A = \begin{bmatrix} \alpha_{11} & \alpha_{12} & 0 & 0 \\ 0 & \alpha_{22} & \alpha_{23} & 0 \\ 0 & 0 & \alpha_{33} & \alpha_{34} \\ 0 & 0 & 0 & \alpha_{44} \end{bmatrix}$$

The transition probability from stage z_1 to stage z_3 is 0 owing to the fact that it has to first go through stage z_2 as it will portray the features of ransomware anyway. If it portrays features of any other form of malware, then it stays in this stage as the detection of other forms of malware is outside the scope of this model. Similarly, the transition probability of stage z_1 to stage z_4 is also zero, as the malware cannot directly make a transition to the final stage without becoming a ransomware first. According to the assumption made in this model, effectively the malware can remain in some other form of malware or become a ransomware.

The transition probability of stage z_2 to z_1 is assumed to be zero. The basis for the assumption is, if the model can depict characteristics of some other form of malware, which is not a ransomware, then it is effectively stage z_4. Hence, any behavior of this type is categorized under phase z_4. The same reasoning applies to the transition probabilities of stages z_3 to z_1 and stage z_4 to z_1. The transition probability of stage z_2 to z_4 is 0, owing to the fact that in stage z_2 it is already a ransomware, and if the attacker is planning to execute a contingency plan of attack then it is effectively stage z_3 as it has already graduated to become an APT [2].

The transition probabilities of stages z_3 to z_2 and stage z_4 to z_2 are assumed to be 0. In stage z_3 the ransomware has graduated to become an APT. On reaching this stage, the ransomware will execute ransomware type APT attack and/or will abort the campaign upon discovery. In stage z_4 the ransomware type APT has decided to execute some other form of attack as a contingency plan of action owing to a belief of being discovered by the defender. The assumption

here is that once a ransomware has graduated to become an APT, it cannot be considered as a simple ransomware, even though it executes a ransomware style attack and/or resort to a contingency plan of attack. Even if the attacker executes a contingency plan of attack which effectively is a ransomware attack, then there is a high possibility that the newer form of ransomware attack would be somewhat different from the primary form of attack, and therefore we assume this as an alternate form of attack and the model denotes the stage to be z_4.

The emission probability matrix and the initial distribution probabilities are not left to any assumption because the attack distribution probabilities as well as the probabilities with which the observable features might be visible depending on the type of resources, the system, the attack framework and the duration of the attack.

5 Discussion

The Decepticon architecture makes the system scalable in nature and easy to use due to its reliance on commercial off-the-shelf components (COTS) such as the TPM. In a corporate environment, where multiple systems are connected to a single gateway, Kidemonas style systems make the environment more secured and scalable in nature. In such an environment, all systems are connected to each other through the PCU network. This also gives the environment the capability to run different types of intrusion detection system on different systems. The information regarding any intrusion is conveyed to all other systems in the networked environment through the PCU network. The PCU network is outside the purview of the attacker, owing to the fact that it doesn't use the regular communication network.

The scalability of Kidemonas style architecture helps in future proofing of the entire system. If needed more computing units can be added to the entire system which would be secure in nature. The transition and emission probabilities once calculated, would provide the defender with valuable information about the malware that would help the user to trigger a cost-effective response from the repository through a smart-box. The biggest advantage for the defender is *awareness*, *security* and cost-effective *countermeasure*. Once the model is put to application in the real world, it would yield numerical values for the transition and emission probability matrices. This helps the defender to make an informed decision, without compromising the quality of service of the system.

A crucial feature manifested by APTs is the existence of a contingency plan of attack [2]. A simple ransomware can be taken care of with the existing infrastructure and defense strategies. But a ransomware type APT might come with a *contingency plan*. A contingency plan of attack is an alternate attack strategy, which the attacker might resort to, if it believes that the defender is able to thwart the primary attack campaign. The type of alternate campaign the attacker might resort to can be completely different from the primary attack strategy. If the attacker is spooked, it can execute the contingency plan and that can inflict unwanted but significant damage to the victim. This warranted the

need for a probabilistic behavioral oriented detection tool and a surreptitious intrusion reporting architecture, which has been presented in this paper.

We further illustrate the utility of our research using WannaCry as a use case. WannaCry is a ransomware type APT [2]. The series of attacks carried out by WannaCry in 2017 is known as "WannaCry Campaign." The attack started on May 12, 2017 and ended on May 17, 2017. Over this period, the attackers earned somewhere between $75,000 to $80,000 from ransom [28]. To begin with, the first things to look for would be the observable features.

- x_1 in this case would flag any process or program searching for the *EternalBlue* vulnerabilities if at all they exist in the system.
- x_2 would flag any process that are actually interacting with the SMBv1 vulnerabilities [28]. It can also denote any process that is interacting with honeypots with similar vulnerabilities.
- x_3 feature manifests the existence of *DoublePulsar* back-door implant tool in the system and/or existence of back-channel communication between the malware and its command and control (C&C) centers.
- x_4 feature denotes the "Campaign Abort" strategy by the malware if it finds itself in a sand-boxed environment.
- x_5 feature is a bit tricky to predict or discern before it has actually been manifested by the attackers. In the context of WannaCry this can be the DDoS attack mounted on the server that hosted the "Kill-Switch" [10].

Once we have the observable features, we can use the tool to predict the hidden/latent states for the WannaCry Campaign.

- z_1 denotes the stage where it can be any malware.
- z_2 denotes the stage where it has manifested the features of being a ransomware.
- z_3 signifies the stage where the ransomware has qualified to become an APT with primary intention of executing ransomware attack or aborting the campaign upon discovery (which in this case is "Do Nothing" strategy when the malware "believes" that it is being run in a sand-boxed environment).
- z_4 manifests the intention of the attacker of executing some other form attack as a contingency plan of attack. In the context of WannaCry the contingency plan of attack is the DDoS attack mounted in the server hosting "Kill-Switch."

Through detailed experiments, the transition probability matrix, the emission probability matrix and the initial distribution matrix can be calculated and put to application in the real world scenario. Every time the status of the malware is detected, a cost-effective countermeasure could be deployed. In the context of WannaCry, following are the countermeasures that could be employed once the status of the malware is known:

- When it is at the stage of malware, simple patching of the system would help. Microsoft had release a patch update as soon as it had learned of the vulnerability.

- When the malware is graduating to become a ransomware then backing-up of the important databases would help.
- As the ransomware graduates to become an APT, blocking back-door traffic along with patching the system as well as maintaining a back-up of the database would help. Also triggering the "Kill-Switch" might help.
- In the final stage, APT proceeds to execute the contingency plan of attack which in this case is the DDoS attack mounted on server hosting the "Kill-Switch." The countermeasure in this case is all the countermeasures applicable for the previous stage as well as another defensive action would be to protect the server which hosts the "Kill-Switch."

The idea is to anticipate the state of the malware and take preventive action. For this purpose, one can use a classifier. Using the feature set for a given state, one can do online prediction of the state of the malware. But a Hidden Markov Model (HMM) based IDS would also be able to provide more behavioral data regarding the malware and in case of an APT, the behavioral pattern of the attacker can be logged and analyzed through the probability matrices. The Decepticon architecture is scalable in nature as shown in Fig. 3. Therefore, it is safe to assume there will be multiple nodes in a networked environment and each of them would be running a Kidemonas or a Decepticon type IDS individually. The intrusion detection happens outside the purview of the attacker. The paper doesn't claim that the APT detection system would be successful all the time. There can be advanced form of attacks, which might defeat the IDS itself, wherein the IDS fails to identify the attack and gives out false negative. In that case, the system comes under attack. But once the attack has occurred, a copy of the malware still exists on the Decepticon architecture. That malware can then be analyzed and attacks on systems with similar vulnerabilities can be thwarted. As shown in Fig. 5, if one system is under attack, the information is communicated to the other nodes in the networked environment through the PCU network and preventive action can be taken to save the remaining nodes. The probability matrix can be updated for future use. The detection system can be trained on past attacks, extracting features and updating the probability matrices. When the IDS gives out false positives, then also the performance is not affected as it happens outside the system.

There can be a classifier which does online predictions. As shown in Fig. 6, given the feature set at time $t + 1$ and the behavior observed till time t (the behavior observed through the feature sets manifested for the respective states) and the states observed till time t, the classifier can predict the state of the malware at time $t+1$. This would be immensely helpful in tailoring a preventive action against the malware. But the HMM based IDS can do more than that. It can be trained using similar attacks originating from different APT groups and/or can be trained on different attacks originating from the same APT group. This would not only help the defender to ascertain the state of the malware but also would give an insight regarding the behavior of the malware and/or the attacker.

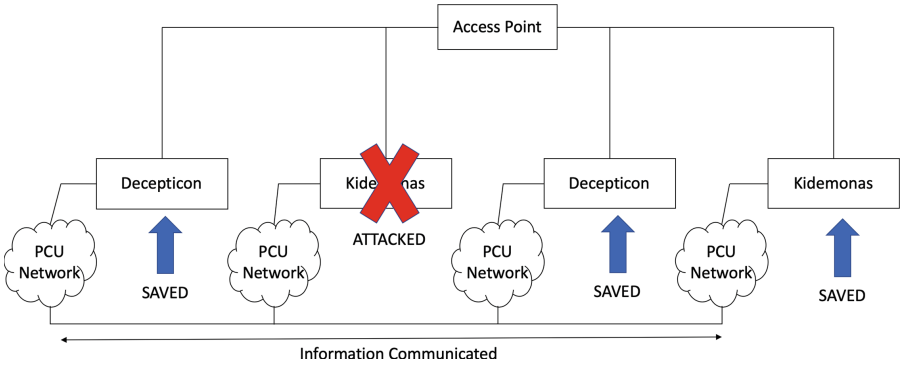

Fig. 5. A scenario wherein one system is under attack

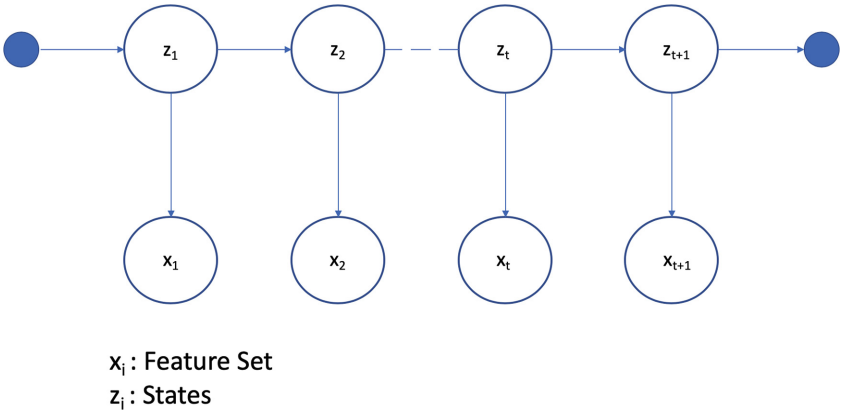

Fig. 6. Classifier based on-line predictive model

As illustrated above, our model shows the way the countermeasures become more sophisticated as and when the malware advances to the higher stages. The calculation of the transition probability and the emission probability matrices as well as the initial probability distribution is not done in this paper due to lack of real world data. The HMM based detection tool and the surreptitious reporting of the intrusion information by the Decepticon architecture pave the way for better security in the corporate environments as well as in the mission critical systems.

6 Conclusion and Future Work

The paper presents an architecture which incorporates the idea of isolation for the purpose of security. It employs deception as a defense technique through the use of hardware-based TPM. The architecture uses deception to surreptitiously report the attack detection to the system administrator. This dupes the attacker

into believing in its silent invasion while giving the defender valuable time to prepare for preventive strategy to thwart the attack. In this paper, we also developed an HMM based ransomware type APT detection tool.

The future work would be to create a test-bench for the analysis of the aforementioned type of malware using the proposed architecture and the detection system. Initially the experiments would be performed using customized software simulation tools. Currently commercially available TPMs have limited memory and processing capabilities. This would make running of process heavy detection models inside a TPM a difficult proposition, and hence, the choice of software simulation tools is a preferred option for initial experiments. No framework is without any drawbacks or limitations. The biggest drawback for the security in this framework is the existence of insider threat. An insider threat can defeat the system. This is another aspect that has to be taken care of in the future work.

Acknowledgment. This research is supported in part by the National Science Foundation under Grant No. DGE – 1754085. Usual disclaimers apply.

References

1. Baksi, R.P., Upadhyaya, S.J.: Kidemonas: the silent guardian. arXiv preprint arXiv:1712.00841 (2017)
2. Baksi, R.P., Upadhyaya, S.J.: A comprehensive model for elucidating advanced persistent threats (APT). In: Proceedings of the International Conference on Security and Management (SAM), pp. 245–251. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (2018)
3. Bencsáth, B., Pék, G., Buttyán, L., Felegyhazi, M.: The cousins of stuxnet: Duqu, flame, and gauss. Future Internet **4**(4), 971–1003 (2012)
4. Bennett, J.T., Moran, N., Villeneuve, N.: Poison ivy: assessing damage and extracting intelligence. FireEye Threat Research Blog (2013)
5. Çeker, H., Zhuang, J., Upadhyaya, S., La, Q.D., Soong, B.-H.: Deception-based game theoretical approach to mitigate DoS attacks. In: Zhu, Q., Alpcan, T., Panaousis, E., Tambe, M., Casey, W. (eds.) GameSec 2016. LNCS, vol. 9996, pp. 18–38. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-47413-7_2
6. Chen, M.Y., Kundu, A., Zhou, J.: Off-line handwritten word recognition using a hidden Markov model type stochastic network. IEEE Trans. Pattern Anal. Mach. Intell. **16**(5), 481–496 (1994)
7. Clark, Z.: The worm that spreads WanaCrypt0r. Malwarebytes Labs, May 2017. https://blog.malwarebytes.com/threat-analysis/2017/05/the-worm-that-spreadswanacrypt0r/
8. Costan, V., Devadas, S.: Intel SGX explained. IACR Cryptol. ePrint Arch. **2016**(086), 1–118 (2016)
9. Falliere, N., Murchu, L.O., Chien, E.: W32. Stuxnet dossier. White paper, Symantec Corporation, Security Response **5**(6), 29 (2011)
10. Greenberg, A.: Hackers are trying to reignite WannaCry with nonstop botnet attacks. Wired Security, May 2017. https://www.wired.com/2017/05/wannacry-ransomware-ddos-attack/

11. Hutchins, E.M., Cloppert, M.J., Amin, R.M.: Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Lead. Issues Inf. Warfare Secur. Res. **1**(1), 80 (2011)
12. Jang, J., et al.: PrivateZone: providing a private execution environment using arm trustzone. IEEE Trans. Depend. Secure Comput. **15**(5), 797–810 (2016)
13. Langner, R.: Stuxnet: dissecting a cyberwarfare weapon. IEEE Secur. Priv. **9**(3), 49–51 (2011)
14. Leonard, C.: 2015 threat report. Websense Security Labs (2015)
15. Ljolje, A., Levinson, S.E.: Development of an acoustic-phonetic hidden Markov model for continuous speech recognition. IEEE Trans. Sig. Process. **39**(1), 29–39 (1991)
16. Ponemon Institute LLC: The state of advanced persistent threats. Ponemon Institute Research Report, December 2013
17. LogRhythm: The APT lifecycle and its log trail. Technical report, July 2013
18. Lorch, J.R., Wang, Y.M., Verbowski, C., Wang, H.J., King, S.: Isolation environment-based information access, 20 September 2011. US Patent 8,024,815
19. Madnick, S.E., Donovan, J.J.: Application and analysis of the virtual machine approach to information system security and isolation. In: Proceedings of the Workshop on Virtual Computer Systems, pp. 210–224. ACM, New York (1973). https://doi.org/10.1145/800122.803961
20. Mehresh, R.: Schemes for surviving advanced persistent threats. Faculty of the Graduate School of the University at Buffalo, State University of New York (2013)
21. Mehresh, R., Upadhyaya, S.: A deception framework for survivability against next generation cyber attacks. In: Proceedings of the International Conference on Security and Management (SAM). p. 1. The Steering Committee of The World Congress in Computer Science, Computer Computer Engineering and Applied Computing (2012)
22. Messaoud, B.I., Guennoun, K., Wahbi, M., Sadik, M.: Advanced persistent threat: new analysis driven by life cycle phases and their challenges. In: 2016 International Conference on Advanced Communication Systems and Information Security (ACOSIS), pp. 1–6. IEEE (2016)
23. Pauna, A.: Improved self adaptive honeypots capable of detecting rootkit malware. In: 2012 9th International Conference on Communications (COMM), pp. 281–284. IEEE (2012)
24. Piolle, E.: Simplified schema of a trusted platform module (TPM). Wikipedia, September 2008. https://commons.wikimedia.org/wiki/File:TPM.svg
25. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. Proc. IEEE **77**(2), 257–286 (1989)
26. Rashid, A., et al.: Detecting and preventing data exfiltration (2014)
27. Kumar Sasidharan, S., Thomas, C.: A survey on metamorphic malware detection based on hidden Markov model. In: 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 357–362. IEEE (2018)
28. Secureworks: WCry Ransomware Campaign. Secureworks Inc., May 2017. https://www.secureworks.com/blog/wcry-ransomware-campaign
29. Shepherd, C., et al.: Secure and trusted execution: past, present, and future-a critical review in the context of the internet of things and cyber-physical systems. In: 2016 IEEE Trustcom/BigDataSE/ISPA, pp. 168–177. IEEE (2016)
30. TCG: TPM main specification. Trusted Computing Group, March 2011. https://trustedcomputinggroup.org/tpm-main-specification/

31. Vukalović, J., Delija, D.: Advanced persistent threats-detection and defense. In: 2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pp. 1324–1330. IEEE (2015)
32. Zakaria, W.Z.A., Abdollah, M.F., Mohd, O., Ariffin, A.F.M.: The rise of ransomware. In: Proceedings of the 2017 International Conference on Software and e-Business, pp. 66–70. ACM (2017)
33. Zhao, C., Saifuding, D., Tian, H., Zhang, Y., Xing, C.: On the performance of Intel SGX. In: 2016 13th Web Information Systems and Applications Conference (WISA), pp. 184–187. IEEE (2016)