Quickest Detection of Advanced Persistent Threats: A Semi-Markov Game Approach

Dinuka Sahabandu
Electrical and Computer Engineering
University of Washington
Seattle, WA 98195, USA
sdinuka@uw.edu

Joey Allen
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA
jallen309@gatech.edu

Shana Moothedath
Electrical and Computer Engineering
University of Washington
Seattle, WA 98195, USA
sm15@uw.edu

Linda Bushnell
Electrical and Computer Engineering
University of Washington
Seattle, WA 98195, USA
lb2@uw.edu

Wenke Lee
College of Computing
Georgia Institute of Technology
Atlanta, GA 30332, USA
wenke@cc.gatech.edu

Radha Poovendran
Electrical and Computer Engineering
University of Washington
Seattle, WA 98195, USA
rp3@uw.edu

***Abstract*—Advanced Persistent Threats (APTs) are stealthy, sophisticated, long-term, multi-stage attacks that threaten the security of sensitive information. Dynamic Information Flow Tracking (DIFT) has been proposed as a promising mechanism to detect and prevent various cyber attacks in computer systems. DIFT tracks suspicious information flows in the system and generates security analysis when anomalous behavior is detected. The number of information flows in a system is typically large and the amount of resources (such as memory, processing power and storage) required for analyzing different flows at different system locations varies. Hence, efficient use of resources is essential to maintain an acceptable level of system performance when using DIFT. On the other hand, the quickest detection of APTs is crucial as APTs are persistent and the damage caused to the system is more when the attacker spends more time in the system. We address the problem of detecting APTs and model the trade-off between resource efficiency and quickest detection of APTs. We propose a game model that captures the interaction of APT and a DIFT-based defender as a two-player, multi-stage, zero-sum, Stackelberg semi-Markov game. Our game considers the performance parameters such as false-negatives generated by DIFT and the time required for executing various operations in the system. We propose a two-time scale Q-learning algorithm that converges to a Stackelberg equilibrium under infinite horizon, limiting average payoff criteria. We validate our model and algorithm on a real-word attack dataset obtained using Refinable Attack INvestigation (RAIN) framework.**

***Index Terms*—Stackelberg zero-sum semi-Markov games, Limiting average reward criteria, Q-learning, Dynamic Information Flow Tracking, Advanced Persistent Threats**

I. Introduction

An Advanced persistent threat (APT) is a prolonged and targeted cyber attack in which an intruder gains illicit access to a system and remains undetected for an extended period of time [1]. The intention of an APT attack is usually to monitor network activity and continuously mine highly sensitive data [2]. APT attacks consist of multiple stages that are initiated by a reconnaissance stage to establish a foothold in the system [3]. Attackers then move laterally through the system, exploring and planning the best attack strategy for the desired data. The lateral movement of APTs consists of multiple stages followed by data exfiltration, which is continued for long period of time until finally detected [4]. Defending against APTs is a challenging task as they are specifically designed to evade conventional security mechanisms such as firewalls, anti-virus software, and intrusion-detection systems that rely on signatures cannot detect them.

APTs introduce information flows in the form of data-flow commands and control-flow commands while interacting with the system, and these are continuously recorded in the log file of the system. Dynamic Information Flow Tracking (DIFT) [5] is a widely accepted detection mechanism against APTs. DIFT uses the information traces recorded in the system log for performing the security analysis [6]. The key idea behind DIFT is that it taints (tags) all suspicious input/data channels and tracks the propagation of the tainted information flows through the system. DIFT generates security analysis using the pre-specified set of security policies whenever it observes an unauthorized use of tainted data. While the security policies incorporated in the DIFT mechanism cover a wide range of attacks, it may not be capable of verifying the authenticity of information flows against all possible attacks at every location resulting in the generation of false-negatives. For instance, while the security rules for buffer overflow protection [7] can be verified at some locations in the system, the security rules for web application vulnerabilities [8] cannot be verified at those locations. Further, performing security analysis at every location is time consuming and hence not practical. Consequently, there is a trade-off regarding *when* and *where* to generate security analysis in the system.

As APTs are stealthy and remain persistent in the system for a long time [9], a quick detection of APTs is important to minimize the damage caused by the attack. Additionally,

This work was supported by ONR grant N00014-16-1-2710 P00002 and DARPA TC grant DARPA FA8650-15-C-7556.

limited availability of resources for defense along with the performance and memory overhead imposed by the defense mechanism on the system require a resource efficient detection technique. An analytical model of DIFT and its interaction with adversarial information flows may enable evaluation of the effectiveness of flow-tainting mechanisms, as well as the design of optimal trapping policies facilitating quick detection of APTs in a resource efficient way. Although there exists games that model the resource efficient detection of APTs [10], [11], [12], [13], they do not address the trade-off between resource efficient and quickest detection of APTs.

In this paper, we provide such an analytical model for DIFT for optimal selection of trapping locations in the system to perform security analysis so as to maximize the probability of detection while minimizing the cost of detection, the time for detection, and false-negatives. Our framework is based on the following insights. 1) The effectiveness of the detection depends on the adversary's strategy, while the adversary's probability of evading detection will be determined by the defender's strategy. This strategic interaction motivates a game-theoretic approach. 2) The efficiency of the defender also depends on the effectiveness of performing security analysis at different locations in the system, which is determined by rate of false-negatives. Hence, the game model is stochastic in nature and the transition probabilities are governed by the false-negative rates at different locations in the system. 3) The game unfolds at multiple stages between the entry point and the exit point of the attack. Each step of the APT attack is a stage in the multi-stage game model, which is characterized by a unique set of critical locations and critical infrastructures of the system, referred to as *destinations*. 4) The time required for executing various system operations, e.g., read, write, along with the time spent by the APT at different locations in the system gives a measure of transition time between the different states of the game, which can be captured as the sojourn time of a semi-Markov game. To this end, we formulate a semi-Markov game model that is played on an *information flow* graph that describes the feasible transitions between processes in the system. At each stage, the adversary decides which process to transition to, while the defender decides whether to trap the information flow or not, at the cost of spending the defense resources and generation of false-negatives. The contributions of this paper are the following:

- We model the strategic interactions between DIFT and APT as a two-player, zero-sum, Stackelberg semi-Markov game. The new semi-Markov game framework captures the cost of performing security analysis, false-negatives generated by DIFT, and the time required for executing various operations in the system (sojourn times) and thereby models the trade-off between resource efficiency and quickest detection of APTs.
- We analyze stationary Stackelberg equilibrium of the game in infinite horizon using limiting average payoff criteria. We present a two-time scale, Q-learning-based algorithm to compute Stackelberg equilibrium of the DIFT vs. APT game and prove the convergence of the algorithm.
- To validate our approach, we provide simulations of the model and the proposed algorithm on *ScreenGrab* log data obtained from RAIN [6] framework.

The remainder of the paper is organized as follows: Section II presents the related work. Section III details preliminaries of information flow graph, the attacker model, and the DIFT detection system. Section IV formulates the zero-sum, semi-Markov game between APTs and DIFT. Section V describes the solution concept, limiting average payoff, and Stackelberg equilibrium used to analyze the game. Section VI presents a two-time scale, Q-learning algorithm to calculate a Stackelberg equilibrium of the zero-sum, semi-Markov game between the players, APT and DIFT. Section VII provides the simulation results of the model and the proposed algorithm using real-world attack dataset. Section VIII presents the concluding remarks and future directions.

II. Related Work

Stackelberg games are widely used in security research to model interactions between two rational players when one player (follower) can observe the policy of the other player (leader). In security problems the defender plays the role of the leader and the adversary acts as the follower. The model of adversary having more information compared to the defender is preferred in security games. The concept of Stackelberg Security Games (SSGs) is introduced in [14] to allocate resources among eight terminals of the Los Angeles International Airport. Recently, SSGs have been used for solving large scale, complex real-life problems such as protecting biodiversity in conservation areas that span over 2500 square kilometers [15] and screening 800 million airport passengers annually throughout USA [16]. While these models consider static (bi-matrix) Stackelberg games, Stackelberg equilibrium has been also studied in Markov (stochastic) games in adversarial patrolling games [17] and moving target defenses [18].

A game-theoretic framework was proposed in the literature to model interaction of APTs with the system [19], [20]. While [19] modeled a deceptive APT, a mimicry attack by APT was considered in [20]. Game-theoretic models are introduced to model the interaction of APTs and a DIFT-based detection system in [10], [11], [12]. The game models in [10], [11], [12] are non-stochastic as false-negatives generated by DIFT are not considered. Recently, a stochastic game model was proposed in [13] where the notion of conditional branching in programs was considered. However, [13] does not capture the time spent by the attacker in the system. Analyzing the trade-off between resource efficient detection and quickest detection of APTs require embedding time into the game model.

Multi-agent reinforcement learning (MARL) algorithms are proposed in the literature to obtain both Nash Equilibrium (NE) strategies and Stackelberg equilibrium strategies of Markov games. In [21] authors provide a value-function-based reinforcement learning algorithm to solve for NE of discounted zero-sum Markov games. Recently, a two-time scale algorithm

to compute an NE of a discounted Markov game is given in [22]. The convergence of MARL algorithms for nonzero-sum games is guaranteed only for special cases where the NE of the game is unique [23]. In [24], a Q-learning-based algorithm is provided to compute discounted Stackelberg equilibrium in Markov games when discounted payoffs are considered. Convergence of the Q-learning algorithm in [24] is shown for the cases of zero-sum games, team games, and for nonzero-sum games with unique best response map for the follower. While papers [21]-[24] deal with discounted reward structure, to analyze strategic interactions of adversaries with long-term goals, such as APTs, average reward criteria is more suitable as it captures the long-run rewards.

III. Preliminaries

In this section, we introduce the information flow graph, the defender model, and the adversary model considered in this paper.

A. Information Flow Graph (IFG)

An *information flow graph* (IFG) is a directed graph whose nodes represent the different components of the computer system (e.g., programs, files, network sockets) and edges represent the feasibility of transferring information flows between the nodes. IFG is constructed from the system logs that capture the history of a system's execution in terms of the spatio-temporal relationships between processes and objects (files and network endpoints) [6].

Collecting and visualizing log data for security analysis is known as provenance-enhanced auditing and it is heavily desired by large enterprises and government agencies due to its ability to answer two key security questions: (i) how an attack infiltrated their systems and (ii) what are the ramifications of the attack. Unfortunately, classical auditing systems cannot efficiently answer these questions; this is because they cannot effectively embed the causal relationships into uniform records, like provenance-enhanced systems [6].

When causal relationships are extracted from audit logs into provenance graphs [6], security-experts can run provenance-dependent queries to derive the origin of an attack and the ramifications of an attack. Identifying the origins of an attack is completed by doing a backward traversal, which analyzes the ancestral dependencies of the attack. Forward analysis techniques traverse through the graph in the forward direction, which effectively determines the ramifications of the attack. Then a point to point analysis technique that extract the intersections of the graphs resulting from forward and backward analysis is used to build an IFG [6]. We use the IFG to depict the victim system in our DIFT vs. APT game. We construct IFG using the system log data collected using RAIN framework [6].

B. Adversary Model: Advanced Persistent Threat (APT)

Advanced persistent threats (APTs) are sophisticated attackers, such as groups of experienced hackers, that establish an illicit, long-term presence in a system in order to mine valuable information/intelligence. The targets of APTS, which are very specifically chosen, typically include large enterprises and governmental organizations. The attacker spends time and resources to identify the vulnerabilities that it can exploit to gain access into the system and to design an attack that will likely remain undetected for a long period of time. These attacks are stealthy and differ from the conventional cyber attacks in complexity and their ability to evade the intrusion detection systems by adopting a nominal system behavior.

An APT attack consists of different attack stages: Initial Compromise, Foothold Establishment, Privilege Escalation, Internal Reconnaissance, Lateral Movement, and Attack Completion. Once the attacker has completed the initial compromise, it will establish a persistent presence by opening up a communication channel with their Command & Control (C&C) server (foothold establishment). Then the attacker collects user credentials such as usernames and passwords to access components of the system that contain critical and sensitive information about the system. APTs are equipped with advanced reconnaissance techniques that are used to study the system behavior and spy on the detection mechanisms employed. We consider an APT attack that consists of multiple attack stages.

C. Defender Model: Dynamic Information Flow Tracking

We consider a Dynamic Information Flow Tracking (DIFT) based defense mechanism [5]. DIFT consists of three main components: (i) tainting at tag sources, (ii) tag propagation policies, and (iii) trapping at tag sinks. Firstly, DIFT taints all the information flows incoming to the set of vulnerable input channels in the system which an adversary can exploit to enter into the system. Then, DIFT tracks the propagation of the tainted flows through the system during various system operations. Finally, DIFT performs security analysis called 'trapping' in order to verify the authenticity of the tagged information flows when an unauthorized usage of a tainted flow is detected. Tag propagation rules and tag check rules at the traps are defined by systems security experts and often called as the security policy of the DIFT. The security policy, i.e., tag check rules, of DIFT is based on the properties of the flow such as the terminal points of the flow and the path traversed. While DIFT is widely used for detection of cyber attackers (see [5], [6] and the references therein), its excessive memory and runtime overhead makes it difficult to integrate into ordinary systems. In this paper, we consider a memory constrained DIFT to defend the system against an APT attack.

IV. Problem Formulation

In this section, we present the two player non-cooperative Stackelberg semi-Markov game (Γ) formulation between the defender player (DIFT), denoted as $\mathcal{P}_D$, and the adversarial player (APT), denoted as $\mathcal{P}_A$.

A. Environment of the game and player roles

Let the directed graph $\mathcal{G} = (V_\mathcal{G}, E_\mathcal{G})$ denote the environment of Γ, where $V_\mathcal{G}$ and $E_\mathcal{G}$ represent the set of nodes,

$V_{\mathcal{G}} = \{v_0, v_1, v_2, \ldots, v_N\}$, and the set of edges, $E_{\mathcal{G}} \subseteq V_{\mathcal{G}} \times V_{\mathcal{G}}$, respectively. The set of nodes $\{v_i\} \subset V_{\mathcal{G}}$, for $i = 1,2,\ldots,N$, depict the nodes in the IFG. Let $\mathcal{D}_0 \subset V_{\mathcal{G}} \backslash \{v_0\}$ denote the set of vulnerable nodes in the IFG which are identified as $\mathcal{P}_A$'s possible entry points into the system. Define edge set $E_0 := \{(v_0, v_i) : v_i \in \mathcal{D}_0\}$, i.e., E_0 denotes the initial foothold of the attack. Then, the edge set $E_{\mathcal{G}} \setminus E_0$ consists of the edges in the IFG that represent the feasibility of transferring information flows between the nodes of the IFG. Further, we consider APT attacks that consist of M number of attack stages. Let the set $\mathcal{D}_j \subset V_{\mathcal{G}} \backslash \{v_0\}$, for $j = 1,2,\ldots,M$, represent the nodes in the IFG corresponding to the targets of the APT in stage j of the attack. A node v_i is called a j^{th} *destination node* if $v_i \in \mathcal{D}_j$, for $j = 1,2,\ldots,M$.

APTs often use advanced reconnaissance techniques to continuously probe the victim system and study the system's configuration profile such as operating system settings and the defense mechanisms deployed in the system. Therefore, we assume $\mathcal{P}_A$ is well aware of the defense policy used by $\mathcal{P}_D$. Hence in Γ, $\mathcal{P}_D$ is the *leader* player and $\mathcal{P}_A$ is the *follower* player.

B. State space and action space

The state of the game at time t refers to the position of the tagged information flow at time t. Let $\mathcal{S} := s_0 \cup \{s_i^j\}$, for $i = 1,2,\ldots,N$ and $j = 1,2,\ldots M$, define the state space of Γ. The state of the game at t is said to be $s_i^j \in \mathcal{S}$, if the position of the tagged information flow at time t is the i^{th} node of the IFG and the attack is in stage j. The state s_0 represents the node v_0 in $\mathcal{G}$ and the state of the game at time $t = 0$ is s_0.

Let $\mathcal{A}_D$ and $\mathcal{A}_A$ denote the action spaces of $\mathcal{P}_D$ and $\mathcal{P}_A$, respectively. Further, let $\mathcal{A}_D(s)$ ($\mathcal{A}_A(s)$), for $s \in \mathcal{S}$, denote the set of actions allowed for the player $\mathcal{P}_D$ ($\mathcal{P}_A$) in state s. Then, $\mathcal{A}_D(s)$ is defined as follows.

$$\mathcal{A}_D(s) = \begin{cases} \{0,1\}, & \text{if } s = s_i^j \text{ and } v_i \notin \{\mathcal{D}_j\}, \text{ for } j = 0,1,\ldots M, \\ \emptyset, & \text{otherwise.} \end{cases} \tag{1}$$

Here, the actions "0" and "1" in the first case of Eq. (1) represent $\mathcal{P}_D$ not deciding and deciding to perform a security analysis on an observed tagged flow at the node v_i of the IFG corresponding to the state s_i^j, respectively. Also, v_i is required to be a non-destination node in stage j since we assume that no security analysis can be performed at the target locations. In other words, a destination (target) is acquired by the APT if the adversarial information flow reaches the node corresponding to that destination. For all other states in $\mathcal{S}$, the action set of $\mathcal{P}_D$ is the empty set. This is because security analysis cannot be performed at the entry points and the destination nodes. The reason that security analysis cannot be performed at the entry points is due to the fact that DIFT concludes if a tagged flow is benign or malicious based on the nature of the information flow which in turn depends on the path traversed by the information flow. Thus performing security analysis at the entry points is not useful.

For given pair of nodes $v, v' \in V_{\mathcal{G}} \backslash \{v_0\}$, v' is said to be an out-neighbor of v if there exists an outgoing edge from v to v' in $\mathcal{G}$, i.e., $(v,v') \in E_{\mathcal{G}}$. Let $\mathcal{N}(v)$ denote the out-neighbor set of node v, i.e., $\mathcal{N}(v) = \{v' : (v,v') \in E_{\mathcal{G}}\}$. Next we define the set $\mathcal{A}_A(s)$ in the following.

$$\mathcal{A}_A(s) = \begin{cases} \mathcal{D}_0 \cup \{\varnothing\}, & \text{if } s = s_0, \\ \mathcal{N}(v_i) \cup \{\varnothing\}, & \text{if } s = s_i^j, j = 1,\ldots,M, \\ \emptyset, & \text{otherwise.} \end{cases} \tag{2}$$

The action $\varnothing$ represents $\mathcal{P}_A$ abandoning the attack (dropping out of the game) before getting detected by $\mathcal{P}_D$. The first case in Eq. (2) defines the action set of $\mathcal{P}_A$ at node v_0 in $\mathcal{G}$ as the set of entry points in the system and abandoning the attack. The second case of Eq. (2) defines the action set of $\mathcal{P}_A$ at a node v_i corresponding to a state s_i^j as the set of out-neighbors of node of v_i and abandoning the attack. The last case in Eq. (2) sets the action set of $\mathcal{P}_A$ as the empty set for all the other states.

C. Player policies and information structure

Let $t = 0,1,2,\ldots$ be the decision-making instants in Γ. Define $s(t), d(t)$, and $a(t)$ to designate the state of Γ, $\mathcal{P}_D$'s action, and $\mathcal{P}_A$'s action at the decision time t, respectively. We assume both players in Γ use *stationary policies*, i.e., choice of actions at t only depends on the current state $s(t)$. Notice that the stationary policies require the least amount of memory while implementing the policies in real-world scenarios. We denote the set of all feasible stationary policies of $\mathcal{P}_D$ and $\mathcal{P}_A$ by $\mathbf{P}_D$ and $\mathbf{P}_A$, respectively. Then a stationary policy $p_D \in \mathbf{P}_D$ is a column vector, $p_D = \left[p_D(s)\right]_{s \in \mathcal{S}}$, where $p_D(s) = \left[p_D(s,d)\right]_{d \in \mathcal{A}_D(s)}$. Each entry $p_D(s,d)$ gives the probability of $\mathcal{P}_D$ choosing an action d in a state s. Similarly, we denote a stationary policy of $\mathcal{P}_A$, $p_A \in \mathbf{P}_A$, by $p_A = \left[p_A(s)\right]_{s \in \mathcal{S}}$, where $p_A(s) = \left[p_A(s,a)\right]_{a \in \mathcal{A}_A(s)}$.

Typically, APT attacks are composed of a program code that enforces a well defined deterministic set of commands that directs the information flows from one node to another node in the IFG in order to achieve a malicious goal. Hence, in Γ we define $\mathbf{P}_A$ to be the set of pure policies, i.e., $p_A : (s,a) \to \{1,0\}$, for $p_A \in \mathbf{P}_A$, $s \in S$, and $a \in \mathcal{A}_A(s)$. Conversely, it has been shown that under Stackelberg leader-follower settings, the leader can perform better (in average sense) when using a proper mixed policy compared to the best leader can do with any pure policy [25]. Therefore, we let $\mathbf{P}_D$ to be set of mixed policies, i.e., $p_D : (s,d) \to [1,0]$, for $p_D \in \mathbf{P}_D$, $s \in S$, and $d \in \mathcal{A}_D(s)$.

Recall that in Γ, we assume that $\mathcal{P}_A$ posses the full information on any defense policy $p_D \in \mathbf{P}_D$ chosen by $\mathcal{P}_D$ since we let $\mathcal{P}_D$ to be the leader and $\mathcal{P}_A$ to be the follower. But neither player knows the exact action taken by their opponent player at a decision epoch t. That is, at any $s(t)$, $\mathcal{P}_A$ does not know whether $\mathcal{P}_D$ performs a security analysis and $\mathcal{P}_D$ does not know whether $\mathcal{P}_A$ decides to abandon the attack or make further progression in the system at the current location of the attack indicated by $s(t)$. In fact at time t, $\mathcal{P}_D$ can only observe that a tagged flow is incoming to a location given by

$s(t)$ and is unaware of whether this tagged flow is a benign or an adversarial flow. While making a decision at $s(t)$, $\mathcal{P}_D$ only assumes that if an incoming tagged flow is an adversarial flow, then $\mathcal{P}_A$ will always choose an optimal action corresponding to the defender's policy. Hence, Γ is an *imperfect information game*. Further we assume both players knows each other's type, feasible space of policies and the payoff functions. Thus, Γ is a *complete information game*.

D. State transitions, rewards, and sojourn time

Let $\mathbb{P}(p_D, p_A)$ denote the state transition matrix of Γ induced by the policies $p_D \in \mathbf{P}_D$ and $p_A \in \mathbf{P}_A$. Then $\mathbb{P}(p_D, p_A)$ can be expressed as $\mathbb{P}(p_D, p_A) = \left[p(s'|s, p_D, p_A)\right]_{s,s' \in \mathcal{S}}$, where $p(s'|s, p_D, p_A) = \sum_{d \in \mathcal{A}_D(s)} \sum_{a \in \mathcal{A}_A(s)} p(s'|s,d,a) p_D(s,d) p_A(s,a)$. Here, $p(s'|s,d,a)$ denotes the probability of transitioning from the current state s to a state s' when $\mathcal{P}_D$ and $\mathcal{P}_A$ are choosing their respective actions d and a, respectively, at s. Then for each tuple (s,s',d,a), where $s,s' \in \mathcal{S}$, $d \in \mathcal{A}_D(s)$, and $a \in \mathcal{A}_A(s)$, $p(s'|s,d,a)$ of Γ is defined in Eq. (3). Moreover, in the following, we assume $s = s_i^j$ such that $v_i \notin \mathcal{D}_j$, for $j = 1,2,\ldots,M$, unless otherwise specified.

$$p(s'|s,d,a) = \begin{cases} 1, & s' = a, d = 0, a \in \mathcal{A}_A(s) \setminus \{\varnothing\} \\ 1, & s' = a, v_i \in \mathcal{D}_0, a \in \mathcal{A}_A(s) \setminus \{\varnothing\} \\ 1, & s' = a, s = s_0, a \in \mathcal{D}_0 \\ 1, & s' = s_0, v_i \notin \mathcal{D}_j : j = 1,2,\ldots,M, a = \varnothing \\ 1, & s' = s_i^{j+1}, v_i \in \mathcal{D}_j : j = 1,2,\ldots,M-1 \\ 1, & s' = s_0, v_i \in \mathcal{D}_M \\ f_n(s), & s' = s_0, d = 1, a \in \mathcal{A}_A(s) \setminus \{\varnothing\} \\ 1 - f_n(s), & s' = a, d = 1, a \in \mathcal{A}_A(s) \setminus \{\varnothing\} \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

Note that, the fifth case of Eq. (3) ensures that at a state that correspond to destination of that stage, i.e., $s = s_i^j$ and $v_i \in \mathcal{D}_j$, the next state is uniquely given by $s' = s_i^{j+1}$. This captures the stage transition of the attack. In other words, the current state of the game being a destination of stage j is equivalent to saying that the next stage of the attack begins at the same destination node. Term $f_n(s) \in [0,1]$ of a state $s = s_i^j$ denotes the probability of $\mathcal{P}_D$ declaring an adversarial flow as a benign flow (false-negatives) after performing the security analysis for a tagged information flow at node v_i in stage j, where $j = 1,\ldots,M$. Thus $f_n(s)$ gives a measure of the accuracy of the security rules employed by $\mathcal{P}_D$ at each node in v_i in identifying adversarial flows in the j^{th} stage of the attack. Note that, as the security rules implemented by DIFT depends on the path traversed by the flow, the stage of the attack also has an effect in deciding the value of $f_n(s)$. In other words, the chances of DIFT incorrectly concluding an adversarial flow as benign is less at a stage j' when compared to stage j, when $j' > j$. Also notice that the structure of Γ allows repeated attacks by $\mathcal{P}_A$. $\mathcal{P}_A$ can continuously attack the system by relaunching the attack: (i) after abandoning a previous attempt, (ii) if it gets detected by $\mathcal{P}_D$ before reaching the final goal, and (iii) after successfully completing the attack by evading detection (cases 4, 6 and 7 in the definition of $p(s'|s,d,a)$ above). This captures the persistent nature of the APT attacks we consider here.

Let $\bar{r}_D(s, p_D, p_A)$ ($\bar{r}_A(s, p_D, p_A)$) denote the expected immediate reward of $\mathcal{P}_D$ ($\mathcal{P}_A$) at a state $s \in \mathcal{S}$ when the players are following their respective policies $p_D \in \mathbf{P}_D$ and $p_A \in \mathbf{P}_A$. Then, $\bar{r}_D(s, p_D, p_A) = \sum_{d \in \mathcal{A}_D(s)} \sum_{a \in \mathcal{A}_A(s)} \bar{r}_D(s,d,a) p_D(s,d) p_A(s,a)$, where $\bar{r}_D(s,d,a) = \sum_{s' \in \mathcal{S}} r_D(s,d,a,s') p(s'|s,d,a)$. Given that the players choose the actions d and a in a state s, the terms $\bar{r}_D(s,d,a)$ and $r_D(s,d,a,s')$ denote the expected immediate reward to $\mathcal{P}_D$ at a state $s \in \mathcal{S}$ and the immediate reward to $\mathcal{P}_D$ when transitioning from state s to s', respectively. Similarly, we can define the terms $\bar{r}_A(s,d,a)$, $r_A(s,d,a,s')$ and the corresponding equations governing the connections between these terms for $\mathcal{P}_A$. In the following, we provide the values assigned to each term $r_D(s,d,a,s')$ and $r_A(s,d,a,s')$ in Γ. We assume $s = s_i^j$ such that $v_i \notin \mathcal{D}_j$, for $j = 1,2,\ldots,M$, unless otherwise specified.

$$r_D(s,d,a,s') = \begin{cases} \beta^j, & d = 0, a \in \mathcal{D}_j, j = 1,2,\ldots,M \\ \sigma^j, & d = 0, a = \varnothing \\ \beta^j + \mathcal{C}_D(s), & d = 1, a \in \mathcal{D}_j, j = 1,2,\ldots,M, \\ & s' = a \\ \mathcal{C}_D(s), & d = 1, a \in \mathcal{A}_A(s) \setminus \{\{\varnothing\} \cup \{\mathcal{D}_j\}\}, \\ & j = 1,\ldots,M, s' \in \mathcal{S} \setminus s_0 \\ \alpha^j + \mathcal{C}_D(s), & d = 1, a \in \mathcal{A}_A(s) \setminus \varnothing, s' = s_0 \\ \sigma^j + \mathcal{C}_D(s), & d = 1, a = \varnothing \\ 0, & \text{otherwise.} \end{cases} \tag{4}$$

Here, $\alpha^j > 0$, $\beta^j < 0$ and $\sigma^j > 0$ terms denote the reward for detecting $\mathcal{P}_A$ before reaching a destination node in $\mathcal{D}_j$, penalty for $\mathcal{P}_A$ evading detection in stage j and reaching a destination node in $\mathcal{D}_j$, and reward for $\mathcal{P}_A$ abandoning the attack in stage j, respectively. These values solely depend on the importance of the set of destination nodes (e.g., sensitivity of data, monetary value) at each stage j to the victim entities. The term $\mathcal{C}_D(s)$ in a state $s \in s_i^j$, for $i = 1,\ldots N$ and $j = 1,\ldots M$, denotes the cost of performing security analysis at node v_i in a stage j, where $j = 1,\ldots M$.

We set $r_A(s,d,a,s') = -r_D(s,d,a,s')$. Hence, Γ has a zero-sum reward structure. This implies that the adversary ($\mathcal{P}_A$) we consider in Γ strategizes not only to reach the target destinations in the system, but also to make the defender incur the maximum cost corresponding to the security analysis. Hence in Γ, we assume $\mathcal{P}_A$ enforces the worst case attack scenario to $\mathcal{P}_D$. Furthermore, APT adversaries maintain stealthiness in the victim systems by often using busy nodes in IFG, i.e., nodes through which most of the benign information flows in the system pass through, and mimicking the behavior of benign flows (e.g., maintaining the rates of malicious information flows within the standard flow bounds) when transferring the adversarial information flows. Such stealthy attack approaches boost $\mathcal{P}_A$'s ability of successfully evading anomaly and signature based intrusion detection systems. Also

note that the performance overhead introduced by DIFT on the system while performing the security analysis, such as processing time and throughput, are proportional to the depth of security analysis employed at the busy nodes. Therefore, typically $f_n(s)$ values associated with the busy nodes are higher compared to the $f_n(s)$ values of other nodes in the IFG. The busyness of a node can be computed empirically by using the total number of tagged flows passing through each node at each stage. Additionally, the values of $f_n(s)$ and $\mathcal{C}_D(s)$ depend on the properties of the processes in the computer system, i.e., nodes of IFG, such as complexity of the arithmetic and logic operations allowed to perform at each node and the resources allocated (e.g., memory, storage, processing power) to each node by the system. Therefore, the zero-sum reward structure considered in Γ also enforces the stealthiness of $\mathcal{P}_A$ by encouraging $\mathcal{P}_A$ to choose high cost nodes when transferring malicious information flows between nodes in the IFG.

Let the expected time spent in a state transition at a state s to all other states when the player policies p_D and p_A are in effect be $\bar{\tau}(s,p_D,p_A) := \sum_{s'\in\mathcal{S}} \tau(s,d,a,s')p(s'|s,a,d)p_D(s,d)p_A(s,a)$. The term $\tau(s,d,a,s')$, referred to as *sojourn time*, gives the time spent in a state transition from a state s to s' when $\mathcal{P}_D$ and $\mathcal{P}_A$ are using actions $d \in \mathcal{A}_D(s)$ and $a \in \mathcal{A}_A(s)$, respectively. The governing sojourn time distributions, $\tau(s,d,a,s')$, when $d = 0$ can be extracted from the differences between timestamps of two consecutive events recorded in the system logs. Estimates for $\tau(s,d,a,s')$ when $d = 1$ depends on the security rules implemented by DIFT at s and the amount of resources, such as memory, storage and processing power, allocated by the system at each node in the IFG. Hence, heuristic approaches that involve simulating multiple instances of DIFT for each pair $(s,s') \in \mathcal{S}$ and for all $a \in \mathcal{A}_A(s)$ are required to obtain estimates of $\tau(s,d,a,s')$, when $d = 1$.

V. Solution Concept

In this section, we present the solution concept used to find optimal policies of $\mathcal{P}_D$ and $\mathcal{P}_A$ in Γ. First we introduce the payoff evaluation criteria in Γ. Then we introduce the concepts of best response and Stackelberg equilibrium in games.

A. Limiting average reward criteria

The persistent nature of APTs causes $\mathcal{P}_A$ to strategize in farseeing manner, considering all the rewards and penalties that $\mathcal{P}_A$ can incur in far future. Conversely, a patient defender, $\mathcal{P}_D$, can take advantage of a such patient APT adversary, $\mathcal{P}_A$, as it allows $\mathcal{P}_D$ to efficiently use the resources across time to successfully detect $\mathcal{P}_A$ compared to an impatient defender. In the process of making decisions, an impatient defender forces itself to use resources at his disposal to detect $\mathcal{P}_A$ only considering the short term rewards and penalties that can be incurred. But in cyber security, a defender being more patient will allow the adversary to cause enough damage to the system before getting detected. Therefore, to capture the persistent nature of $\mathcal{P}_A$ and the trade-off of $\mathcal{P}_D$ between being patient to be more resource efficient and being impatient to detect $\mathcal{P}_A$ quickly in order to avoid further damage to the system, we use infinite horizon limiting average payoff evaluation criteria.

Definition V.1. *Let $U(s_0,p_D,p_A)$ denote the limiting average payoff of Γ when the initial state of Γ is set to s_0 and the players are following the policies $p_D \in \mathbf{P}_D$ and $p_A \in \mathbf{P}_A$. Then we define $U(s_0,p_D,p_A)$ as follows.*

$$U(s_0,p_D,p_A) = \limsup_{T\to\infty} \frac{\sum_{t=0}^{T} \bar{r}_D(s(t),p_D,p_A)}{\sum_{t=0}^{T} \bar{\tau}(s(t),p_D,p_A)}, \tag{5}$$

where $s(t) \in \mathcal{S}$ denote the state at time t with $s(0) = s_0$.

Using the definition in Eq. (5), we define the limiting average payoff of the players $\mathcal{P}_D$ and $\mathcal{P}_A$ as $U_D(s_0,p_D,p_A) = U(s_0,p_D,p_A)$ and $U_A(s_0,p_D,p_A) = -U(s_0,p_D,p_A)$, respectively. Notice that the expected rewards and the expected sojourn times integrated in the payoff structure characterize the trade-off between resource efficiency vs. quickest detection.

Next we formally define the followers response to the leader's policy in Stackelberg settings using $U(s_0,p_D,p_A)$.

B. Best response of $\mathcal{P}_A$

Recall that in Stackelberg game settings the leader $(\mathcal{P}_D)$ first commits to a policy $(p_D \in \mathbf{P}_D)$. Then the follower $(\mathcal{P}_A)$ observes the leader's policy $(p_D \in \mathbf{P}_D)$ and chooses a policy $(p_A \in \mathbf{P}_A)$ that maximizes the follower's payoff function $(U_A(s_0,p_D,p_A))$. We call such a policy of $\mathcal{P}_A$ a best response of $\mathcal{P}_A$ and it is defined as follows.

Definition V.2. *Let $BR(p_D)$ denote the best response of $\mathcal{P}_A$. Then for a given defender's policy $p_D \in \mathbf{P}_D$,*

$$BR(p_D) = \arg\max_{p_A\in\mathbf{P}_A} U_A(s_0,p_D,p_A) = \arg\min_{p_A\in\mathbf{P}_A} U(s_0,p_D,p_A). \tag{6}$$

Next we introduce the equilibrium concept we employed to analyze Γ using the definition of $BR(p_D)$.

C. Stackelberg equilibrium

Stackelberg equilibrium provides a natural framework to analyze the equilibrium in leader-follower settings. The following defines the policies and payoff of Γ under Stackelberg equilibrium.

Definition V.3. *A policy pair $(p_D^*,p_A^*) \in (\mathbf{P}_D,\mathbf{P}_A)$ in Γ is said to form a Stackelberg equilibrium if,*

$$U(s_0,p_D,BR(p_D)) \leqslant U(s_0,p_D^*,p_A^*) \text{ for all } (p_D,p_A) \in (\mathbf{P}_D,\mathbf{P}_A),$$

where $p_A^* = BR(p_D^*)$. Then the payoff of Γ at Stackelberg equilibrium is given by,

$$U^*(s_0,p_D^*,p_A^*) = \max_{p_D\in\mathbf{P}_D} U(s_0,p_D,BR(p_D)). \tag{7}$$

VI. An Algorithm to Solve for Stackelberg Equilibrium

In this section, we provide a reinforcement learning-based algorithm to compute optimal player policies $p_D^* \in \mathbf{P}_D$ and $p_A^* \in \mathbf{P}_A$. The outline of our approach is as follows. We first

prove that an algorithm that tracks Stackelberg equilibrium at each state $s \in \mathcal{S}$ is sufficient to compute Stackelberg equilibrium of Γ. It has been shown in [24] that such an algorithm exists for computing Stackelberg strategies in Markov games. The following theorem extends the result given in Theorem 1 of [24] for discounted Markov games to the case of limiting average semi-Markov games considered in this paper.

Theorem VI.1. *Let $U^*(p_D^*, p_A^*) = \left[U^*(s, p_D^*, p_A^*)\right]_{s\in\mathcal{S}}$ be the limiting average equilibrium payoff vector of Γ under Stackelberg equilibrium player policies p_D^* and p_A^*. Then the following statements are equivalent.*

1) *Policy pair (p_D^*, p_A^*) forms a Stackelberg equilibrium point in Γ with equilibrium payoff $U^*(p_D^*, p_A^*)$.*
2) *For each $s \in \mathcal{S}$, the policy pair $(p_D^*(s), p_A^*(s))$ forms a Stackelberg equilibrium point in the static $|\mathcal{A}_D(s)| \times |\mathcal{A}_A(s)|$ matrix game $Q(s) = \left[Q(s,d,a)\right]_{d\in\mathcal{A}_D(s), a\in\mathcal{A}_A(s)}$ with Stackelberg equilibrium payoff $U^*(s, p_D^*, p_A^*) := \max_{p_D\in\mathbf{P}_D} U(s, p_D, BR(p_D))$, where each entry of $Q(s)$ is defined as*

$$Q(s,d,a) = \sum_{s'\in\mathcal{S}} p(s'|s,d,a)\,[\,r_D(s,d,a,s') - \rho\,\tau(s,d,a,s') + U^*(s', p_D^*, p_A^*)\big], \tag{8}$$

where $\rho = U^*(s_0, p_D^*, p_A^*)$ and $s_0 \in \mathcal{S}$ is the initial state of Γ. Furthermore, $|\mathcal{A}_D(s)|$ and $|\mathcal{A}_A(s)|$ represent the cardinality of the sets $\mathcal{A}_D(s)$ and $\mathcal{A}_A(s)$.

Proof of Theorem VI.1 is presented in the Appendix.

Motivated by the single agent average reward Q-learning algorithm presented in [26] for semi-Markov decision processes and using Theorem VI.1, we present a two time-scale Q-learning-based algorithm that tracks Stackelberg equilibrium of Γ. Moreover, our algorithm generalizes the algorithm presented in [26] to the case of two player zero-sum semi-Markov game under Stackelberg equilibrium settings. The core update equations of our proposed algorithm is given below. For $s, s' \in \mathcal{S}$, $d \in \mathcal{A}_D(s)$, $a \in \mathcal{A}_A(s)$, $p_D \in \mathbf{P}_D$, $p_A \in \mathbf{P}_A$, and iterates of the algorithm $t = 0, 1, \ldots,$

$$Q^{t+1}(s,d,a) = (1-\gamma_1(m(t,s,d,a)))Q^t(s,d,a) + \gamma_1(m(t,s,d,a))\,[r_D(s,d,a,s') - \rho^t\tau(s,d,a,s') + U^t(s', p_A, p_D)], \tag{9}$$

$$\rho^{t+1} = (1-\gamma_2(t))\rho^t + \gamma_2(t)\Big[\frac{T(t)\rho^t + r_D(s,d,a,s')}{T(t+1)}\Big]. \tag{10}$$

The term $m(t,s,d,a)$ denotes the total number of times action pair (d,a) is tried in a state s till iteration t. Functions $\gamma_1(m(t,s,d,a))$ and $\gamma_2(t)$ represent the learning rates. $T(t)$ and $U^t(s', p_A(s), p_D(s))$ denote the sum of the time spent in all the states visited till iteration t and the limiting average payoff of the Stackelberg matrix game $Q^t(s)$, respectively. Notice that Eq. (9) is the iterative version of Eq. (8) and Eq. (10) updates the limiting average payoff of Γ, ρ^t.

Next we present proof of convergence of the iterates given in Eq. (9) and Eq. (10). The proof of the convergence result uses Borkar's Lemma [27] presented in the following proposition.

Proposition VI.2 (Borkar's Lemma)**.** *Let $\{x^t\} \in \mathbb{R}^n$ and $\{y^t\} \in \mathbb{R}^l$ be two sequences with $t = 0, 1, \ldots$ iterates*

$$x^{t+1} = x^t + \delta_1(t)[F(x^t, y^1) + W_1^t], \tag{11}$$

$$y^{t+1} = y^t + \delta_2(t)[G(x^t, y^1) + W_2^t]. \tag{12}$$

Let $\lambda : y \to x$. Then, given the iterates in Eqs. (11) *and* (12) *are bounded, $\{(x^t, y^t)\}$ converges to $(\lambda(y^*), y^*)$ with probability one under the following conditions.*

1) *The maps $F : \mathbb{R}^{n+l} \to \mathbb{R}^n$ and $G : \mathbb{R}^{n+l} \to \mathbb{R}^l$ are Lipschitz.*
2) *$\sum_{t=0}^{\infty} \delta_1(t) = \sum_{t=0}^{\infty} \delta_2(t) = \infty$, $\sum_{t=0}^{\infty} \delta_1(t)^2 = \sum_{t=0}^{\infty} \delta_2(t)^2 < \infty$, and $\limsup_{t\to\infty} \frac{\delta_2(t)}{\delta_1(t)} = 0$.*
3) *W_1^t and W_2^t are two sequences of random variables (noise) satisfying, $\sum_{t=0}^{\infty} \delta_1(t) W_1^t$, $\sum_{t=0}^{\infty} \delta_2(t) W_2^t < \infty$, almost surely.*
4) *For all $y \in \mathbb{R}^l$, the ODE $\frac{dx}{dt} = F(x(t), y)$ has an asymptotically stable critical point $\lambda(y)$ such that function λ is Lipschitz.*
5) *The ODE $\frac{dy}{dt} = G(\lambda(y(t)), y(t))$ has a global asymptotically stable critical point.*

We show a relation between updating equations given in Eq. (9) and Eq. (10) and the two sequences presented in Eq. (11) and Eq. (12) of Proposition VI.2.

Define two mappings F_1 and F_2 associated with Eq. (9) as

$$F_1(Q^t(s,d,a)) = \sum_{s'\in\mathcal{S}} P(s'|s,a,d)\Big[r(s,a,d,s') - \rho^t\tau(s,a,d,s') + U^t(s', p_A(s), p_D(s))\Big], \tag{13}$$

$$F_2(Q^t(s,d,a)) = r(s,a,d,s') - \rho^t\tau(s,a,d,s') + U^t(s', p_A, p_D). \tag{14}$$

Also define mappings G_1 and G_2 associated with Eq. (10),

$$G_1(\rho^t) = \sum_{s'\in\mathcal{S}} P(s'|s,a,d)\Big[\frac{r_D(s,a,d,s') + T(k)\rho^k}{T(k+1)}\Big], \tag{15}$$

$$G_2(\rho^t) = P(s'|s,a,d)\Big[\frac{r_D(s,a,d,s') + T(k)\rho^k}{T(k+1)}\Big]. \tag{16}$$

Eq. (14) and Eq. (16) are the Robbins-Monro stochastic approximation [28] versions of Eq. (13) and Eq. (15), respectively. The following equations express the update equations given in Eq. (9) and Eq. (10) in the form of Eq. (11) and Eq. (12) using the maps F_1, F_2, G_1, and G_2.

$$Q^{t+1}(s,d,a) = Q^t(s,d,a) + \gamma_1(m(t,s,d,a))[F(Q^t(s,d,a)) + W_1^t], \tag{17}$$

$$\rho^{t+1} = \rho^t + \gamma_2(t)[G(\rho^t) + W_2^t], \tag{18}$$

where $F(Q^t(s,d,a)) = F_1(Q^t(s,d,a)) - Q^k(s,d,a)$, $W_1^t = F_2(Q^t(s,d,a)) - F_1(Q^t(s,d,a))$, $G(\rho^t) = G_1(\rho^t) - \rho^t$, and $W_2^t = G_2(\rho^t) - G_1(\rho^t)$.

We present an algorithm to compute Stackelberg equilibrium policies in Γ in Algorithm VI.1.

Algorithm VI.1 Algorithm to compute Stackelberg equilibrium of Γ

Input: State space (**S**), transition structure ($\mathcal{P}$), rewards (r_D), sojourn times (τ), number of iterations ($I >> 0$)
Output: Stackelberg equilibrium policies, $(p_D^\star, p_A^\star) \leftarrow (p_A^I, p_D^I)$ and limiting average payoff value $\rho^\star$
Initialization: $t \leftarrow 0$, $Q^0 \leftarrow 0$, $p_D^0 \leftarrow \mathbf{P}_D$, $p_A^0 \leftarrow \mathbf{P}_A$, $s = s_0$
while $t \leqslant T$ **do**
    $t \leftarrow t+1$
    Draw d^t from $p_D^t(s)$ and a^t from $p_A^t(s)$
    Observe immediate reward $r_D(s,d^t,a^t,s')$ and sojourn time $\tau(s,d^t,a^t,s')$
    Reveal the next state s' according to $\mathcal{P}$
    Solve Stackelberg matrix game $Q^t(s')$ to find $U^t(s',p_A,p_D)$ and update player policies $p_D^t(s')$ and $p_A^t(s')$
    Update $Q^t(s,d,a)$ and ρ^t:

$$Q^{t+1}(s,d,a) = (1-\gamma_1(m(t,s,d,a)))Q^t(s,d,a) + \gamma_1(m(t,s,d,a))[r_D(s,d,a,s') - \rho^t\tau(s,d,a,s') + U^t(s',p_A,p_D)]$$

$$\rho^{t+1} = (1-\gamma_2(t))\rho^t + \gamma_2(t)\Big[\frac{T(t)\rho^t + r_D(s,d,a,s')}{T(t+1)}\Big]$$

    Update the state of Γ: $s \leftarrow s'$
end while

Theorem VI.3 provides the convergence guarantee of Algorithm VI.1.

Theorem VI.3. *All $Q^t(s,d,a)$ and ρ^t sequences in Eq.* (9) *and Eq.* (10) *converge and at the convergence, policy pair (p_D,p_A) forms a Stackelberg equilibrium in Γ, given that the sequences $Q^t(s,d,a)$ and ρ^t are bounded under the conditions in Proposition VI.2 and under the following assumptions.*

1) *A unique learning rate $\gamma_1(m(t,s,d,a))$ in Eq.* (9) *is decayed based on the number of times an action pair $(d,a) \in (\mathcal{A}_D(s),\mathcal{A}_A(s))$ is tried at a state $s \in \mathcal{S}$. Also a unique learning rate $\gamma_2(t)$ in Eq.* (10) *is decayed based on the number of times ρ is updated.*
2) *The Markov chain induced by the $\mathbb{P}(p_D,p_A)$ for any $p_D \in \mathbf{P}_D$ and $p_A \in \mathbf{P}_A$ contains only a single recurrent class (unichain Markov structure) and there exists at least one $s \in S$ that will be common to all the Markov chains induced by $\mathbb{P}(p_D,p_A)$.*

Proof of Theorem VI.3 uses boundedness of the iterates $Q^t(s,d,a)$ in Eq. (9) which we prove in Lemma VI.5. First, we define the weighted sup-norm of a vector with respect to another vector and then present the lemma.

Definition VI.4. *Let $||u||_\varepsilon$ denote the weighted sup-norm of a vector u with respect to the vecor $\varepsilon \in \mathbb{R}^n$. Then,*

$$||u||_\varepsilon = \max_{k=1,\dots,n} \frac{|u(k)|}{\varepsilon(k)},$$

where $|u(k)|$ represent the absolute value of the k^{th} entry of vector u.

Lemma VI.5. *The iterates $Q^t(s,d,a)$ in Eq.* (9) *are bounded under the conditions* 2) *and* 3) *in Proposition VI.2 and assumption* 2) *given in Theorem VI.3.*

Proof of Lemma VI.5 is given in the Appendix. Now we present the proof of Theorem VI.3.

Proof of Theorem VI.3: Boundedness of the iterates $Q^t(s,d,a)$ are proved in Lemma VI.5. Limiting average cost, ρ, is finite due to finiteness of rewards $r_D(s,d,a,s')$, sojourn times $\tau(s,d,a,s')$, and finite state space in Γ [29]. Hence ρ iterates are bounded.

Observing both maps F and G are linear maps in $Q^t(s,d,a)$ and ρ^t, respectively (see Eq. (9) and Eq. (10)). Therefore it follows that F and G are Lipschitz (Condition 1) in Proposition VI.2.

We set $\gamma_1(m(t,s,d,a)) = \dfrac{1}{m(t,s,d,a)}$ and $\gamma_2(t) = \dfrac{1}{t}$ in Eq. (9) and Eq. (10) to satisfy assumption 1) in Theorem VI.3. Notice that $\delta_1(t) = \gamma_1(m(t,s,d,a)) = \dfrac{1}{m(t,s,d,a)}$ and $\delta_2(t) = \gamma_2(t) = \dfrac{1}{t}$ in Eq. (17) and Eq. (18) satisfies the condition 2) (standard step size assumptions for stochastic approximation algorithms) in Proposition VI.2. □

VII. Simulations

In this section, we provide simulation results of Algorithm VI.1. Simulations were performed on an IFG extracted from a log data set related to an attack, *ScreenGrab*, conducted by US DARPA red-team during an evaluation of RAIN system [6] for the transparent computing program. The resulting IFG is shown in Figure 1. During *ScreenGrab* attack, the adversary aims to get access to the ScreenGrab process (see node v_{12} in Figure 1) to capture the screenshot of the victim's desktop. For simulation purposes, we consider *ScreenGrab* attack as a two stage attack and select IFG nodes v_1 and v_{10} as the entry point and the destination of stage 1, respectively.

State space of Γ induced by IFG in Figure 1 is shown in Figure 2. IFG nodes v_6 and v_7 are being excluded in the state space (Figure 2) as they do not have a path to the destination of first attack stage (v_{10}). The nodes s_{10}^1 and s_{10}^2 in Figure 2 shows the transitioning between attack stages. We calculate fraction of flows through each node from the log data and multiply the resulting distribution by a negative constant to estimate the $\mathcal{C}_D$ values. Rewards and sojourn times are appropriately chosen for the simulation purposes. Figure 3 shows the convergence of the Stackelberg limiting average payoff value of Γ (ρ) in Algorithm VI.1 within a span of 1000 iterations. Figure 4 (a) illustrates the corresponding p_D at the Stackelberg equilibrium.

Figure 4 compares the two cases where protecting stage 1 destination is more beneficial compared to protecting stage 2 destination (Figure 4 (a)) and vice-versa (Figure 4 (b)). The results suggest that, at the equilibrium, $\mathcal{P}_D$ assign more weight in performing security analysis at stage 1 nodes when first case is considered and vice-versa. Figure 5 compares the

equilibrium policy of defender, $p_D^\star$ under low (Figure 5 (a)) and high (Figure 5 (b)) $\mathcal{C}_D$ values. Results show $\mathcal{P}_D$ takes more conservative approach in performing security analysis when high security analysis costs are involved.

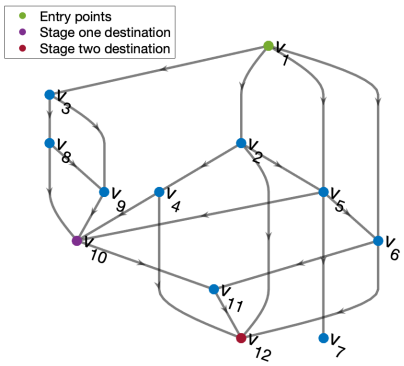

Figure 1: IFG of *ScreenGrab* attack. Nodes v_1 and v_{10} have been chosen as the entry point and the destination of first attack stage, respectively. Destination node of the second stage, v_{12}, represents the goal of the attack, *ScreenGrab* process.

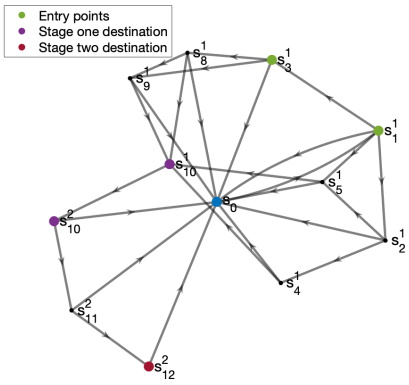

Figure 2: State space of Γ induced by the IFG in Figure 1. Attack progressing from first stage to second stage is captured by the states s_{10}^1 and s_{10}^2 (purple colored nodes). Transitions from all the states excluding the state s_{10}^2 to state s_0 are due to action $\varnothing$ of $\mathcal{P}_A$ and/or action $\{1\}$ of $\mathcal{P}_D$.

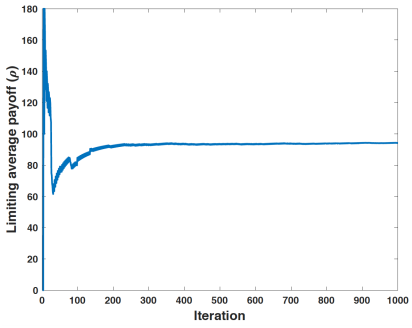

Figure 3: Convergence of the limiting average payoff value, ρ in Algorithm VI.1.

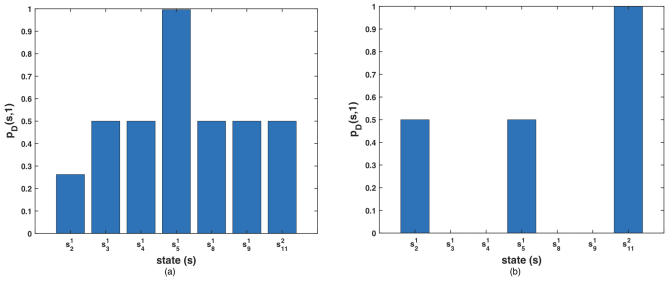

Figure 4: Figures show the Stackelberg equilibrium policy of $\mathcal{P}_D$ (i.e., $p_D \in \mathbf{P}_D$) computed using Algorithm VI.1. The figure on the left considers a case where $\alpha^1 > \alpha^2$, $\beta^1 < \beta^2$, and $\sigma^1 > \sigma^2$. On the contrary figure on the right considers the case of $\alpha^1 < \alpha^2$, $\beta^1 > \beta^2$, and $\sigma^1 < \sigma^2$. In both cases same $\mathcal{C}_D$ values, τ values and $\mathbb{P}$.

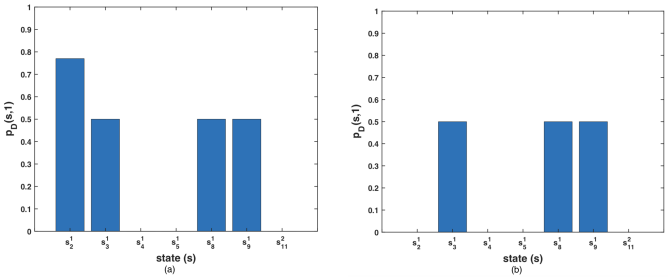

Figure 5: Figures show the Stackelberg equilibrium policy of $\mathcal{P}_D$ (i.e., $p_D \in \mathbf{P}_D$) computed using Algorithm VI.1. Let $\mathcal{C}_D$ be the security cost values used in the left figure. Then figure on the right considers a case where $\mathcal{C}_D$ values are scaled by 10. All other parameters in Γ are same for both cases.

VIII. Conclusions

In this paper, we presented a game theoretic model that captures the system level interactions between APTs and DIFT for enabling a resource efficient mechanism that can effectively detect and prevent threats imposed by APTs. We modeled the interaction between APT and DIFT as a two-player, zero-sum, Stackelberg semi-Markov game. Our game model addresses the trade-off between resource efficiency and quickest detection of APTs. An infinite-horizon, limiting average payoff criteria is used to analyze the game. The game consists of multiple stages where each stage corresponds to a stage in the attack. The transition probabilities of the game, depend on the effectiveness of the defense mechanism (DIFT) such as false-negative rates, are assumed to be unknown. We presented a two-time scale Q-learning algorithm that converge to a Stackelberg equilibrium of the DIFT vs. APT game. The convergence of the algorithm is proved utilizing the structure of the Markov chain induced by the policies of the players. In order to validate the proposed model and algorithm, we conducted simulations on *ScreenGrab* attack data obtained from Refinable Attack INvestigation (RAIN) [6] framework. In future, we plan to extend our model to have a nonzero-sum game reward structure and analyze the

underlying Stackelberg equilibrium. It will be also interesting to extend the current attacker model to a multi-attacker case and study the equilibrium of the Stackelberg semi-Markov game with one leader and multiple followers.

REFERENCES

[1] J. Jang-Jaccard and S. Nepal, "A survey of emerging threats in cybersecurity," *Journal of Computer and System Sciences*, vol. 80, no. 5, pp. 973–993, 2014.

[2] B. Watkins, "The impact of cyber attacks on the private sector," pp. 1–11, 2014.

[3] G. Brogi and V. V. T. Tong, "TerminAPTor: Highlighting advanced persistent threats through information flow tracking," *IFIP International Conference on New Technologies, Mobility and Security*, pp. 1–5, 2016.

[4] B. Bencsáth, G. Pék, L. Buttyán, and M. Felegyhazi, "The cousins of Stuxnet: Duqu, Flame, and Gauss," *Future Internet*, vol. 4, no. 4, pp. 971–1003, 2012.

[5] J. Newsome and D. Song, "Dynamic taint analysis: Automatic detection, analysis, and signature generation of exploit attacks on commodity software," *Network and Distributed Systems Security Symposium*, 2005.

[6] Y. Ji, S. Lee, E. Downing, W. Wang, M. Fazzini, T. Kim, A. Orso, and W. Lee, "RAIN: Refinable attack investigation with on-demand inter-process information flow tracking," *ACM SIGSAC Conference on Computer and Communications Security*, pp. 377–390, 2017.

[7] M. Dalton, H. Kannan, and C. Kozyrakis, "Real-world buffer overflow protection for userspace and kernelspace." *USENIX Security Symposium*, pp. 395–410, 2008.

[8] M. Dalton, C. Kozyrakis, and N. Zeldovich, "Nemesis: Preventing authentication & access control vulnerabilities in web applications," *USENIX Security Symposium*, pp. 267–282, 2009.

[9] P. Jacobs, "Dwell time is down, APTs are on the rise, and other cyberattack trends you need to know," Jul 2019. [Online]. Available: https://governmenttechnologyinsider.com/dwell-time-is-down-apts-are-on-the-rise-and-other-cyberattack-trends-you-need-to-know/#.Xa4Odfd7myI

[10] S. Moothedath, D. Sahabandu, J. Allen, A. Clark, L. Bushnell, W. Lee, and R. Poovendran, "A game theoretic approach for dynamic information flow tracking to detect multi-stage advanced persistent threats," *ArXiv e-prints*, p. arXiv:1811.05622, Nov. 2018.

[11] S. Moothedath, D. Sahabandu, A. Clark, S. Lee, W. Lee, and R. Poovendran, "Multi-stage dynamic information flow tracking game," *Conference on Decision and Game Theory for Security, Lecture Notes in Computer Science*, vol. 11199, pp. 80–101, 2018.

[12] D. Sahabandu, B. Xiao, A. Clark, S. Lee, W. Lee, and R. Poovendran, "DIFT games: Dynamic information flow tracking games for advanced persistent threats," *IEEE Conference on Decision and Control (CDC)*, pp. 1136–1143, 2018.

[13] D. Sahabandu, S. Moothedath, A. Clark, J. Allen, L. Bushnell, W. Lee, and R. Poovendran, "A game theoretic approach for dynamic information flow tracking with conditional branching," in *American Control Conference (ACC)*, 2019.

[14] J. Pita, M. Jain, J. Marecki, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus, "Deployed armor protection: the application of a game theoretic model for security at the los angeles international airport," in *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track*. International Foundation for Autonomous Agents and Multiagent Systems, 2008, pp. 125–132.

[15] F. Fang, T. H. Nguyen, R. Pickles, W. Y. Lam, G. R. Clements, B. An, A. Singh, M. Tambe, and A. Lemieux, "Deploying paws: Field optimization of the protection assistant for wildlife security," in *Twenty-Eighth IAAI Conference*, 2016.

[16] M. Brown, A. Sinha, A. Schlenker, and M. Tambe, "One size does not fit all: A game-theoretic approach for dynamically and effectively screening for threats," in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[17] Y. Vorobeychik, B. An, M. Tambe, and S. Singh, "Computing solutions in infinite-horizon discounted adversarial patrolling games," in *Twenty-Fourth International Conference on Automated Planning and Scheduling*, 2014.

[18] X. Feng, Z. Zheng, P. Mohapatra, and D. Cansever, "A stackelberg game and markov modeling of moving target defense," in *International Conference on Decision and Game Theory for Security*. Springer, 2017, pp. 315–335.

[19] L. Huang and Q. Zhu, "Adaptive strategic cyber defense for advanced persistent threats in critical infrastructure networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 46, no. 2, pp. 52–56, 2019.

[20] M. O. Sayin, H. Hosseini, R. Poovendran, and T. Başar, "A game theoretical framework for inter-process adversarial intervention detection," *International Conference on Decision and Game Theory for Security*, pp. 486–507, 2018.

[21] C. Szepesvári and M. L. Littman, "A unified analysis of value-function-based reinforcement-learning algorithms," *Neural computation*, vol. 11, no. 8, pp. 2017–2060, 1999.

[22] H. Prasad, P. LA, and S. Bhatnagar, "Two-timescale algorithms for learning Nash equilibria in general-sum stochastic games," *International Conference on Autonomous Agents and Multiagent Systems*, pp. 1371–1379, 2015.

[23] J. Hu and M. P. Wellman, "Nash Q-learning for general-sum stochastic games," *Journal of Machine Learning Research*, vol. 4, pp. 1039–1069, 2003.

[24] V. Könönen, "Asymmetric multiagent reinforcement learning," *Web Intelligence and Agent Systems: An international journal*, vol. 2, no. 2, pp. 105–121, 2004.

[25] T. Basar *et al.*, "Lecture notes on non-cooperative game theory," *Game Theory Module of the Graduate Program in Network Mathematics*, pp. 3–6, 2010.

[26] A. Gosavi, "Reinforcement learning for long-run average cost," *European Journal of Operational Research*, vol. 155, no. 3, pp. 654–674, 2004.

[27] V. S. Borkar, "Stochastic approximation with two time scales," *Systems & Control Letters*, vol. 29, no. 5, pp. 291–294, 1997.

[28] H. Robbins and S. Monro, "A stochastic approximation method," *The annals of mathematical statistics*, pp. 400–407, 1951.

[29] A. K. Lal and S. Sinha, "Zero-sum two-person semi-markov games," *Journal of applied probability*, vol. 29, no. 1, pp. 56–72, 1992.

[30] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-dynamic programming*. Athena Scientific Belmont, MA, 1996, vol. 5.

APPENDIX

Proof of Theorem VI.1: We first assume 2) holds and prove 1). Since 2) holds, for each $s \in \mathcal{S}$ we have $p_D(s)Q(s)p_A(s) = p_D(s)Q(s)BR(p_D(s)) \leqslant p_D^*(s)Q(s)BR(p_D^*(s)) = p_D^*(s)Q(s)p_A^*(s) = U^*(s,p_D^*,p_A^*)$. The equality holds only when $p_D(s) = p_D^*(s)$. This immediately suggests $U^*(p_D^*,p_A^*) \geqslant U^*(p_D,p_A)$ and proves the first statement.

Here we assume 1) holds and then prove 2). Let $\bar{s} \in \mathcal{S}$ be an arbitrary initial state of Γ. Then consider a p_D, where $p_D(\bar{s}) \neq p_D^*(\bar{s})$ and $p_D(s) = p_D^*(s)$ for each $s \in \mathcal{S}\backslash\bar{s}$. The Bellman equation associated with the state $\bar{s}$ is [29]:

$$\arg\max_{p_D(\bar{s})\in\mathbf{P}_D(\bar{s})} \bar{r}_D(\bar{s},p_D(\bar{s}),BR(p_D(\bar{s}))) - \rho\,\bar{\tau}(\bar{s},p_D(\bar{s}),BR(p_D(\bar{s}))) + \sum_{s'\in\mathcal{S}} p(s'|s,d,a)\left[U^*(s',p_D,p_A)\right].$$

But we know from 1) that (p_D^*,p_A^*) forms a Stackelberg equilibrium. Therefore, $p_D^*(\bar{s})$ and $p_A^*(\bar{s}) = BR(p_D^*(\bar{s}))$ will maximize the aforementioned expression with the maximum value of $U^*(\bar{s},p_D^*,p_A^*)$. Now since $\bar{s}$ can be any arbitrary state, it immediately implies the second statement. □

Proof of Lemma VI.5: Let $\hat{F}_1(Q^t(s,d,a))$ ($\hat{F}_1(Q^t)$ for brevity) is defined as, $\hat{F}_1(Q^t) = \sum_{s'\in\mathcal{S}} P(s'|s,a,d)\Big[r(s,a,d,s') +$

$$U^t(s', p_A(s), p_D(s))\Big]$$
$$= \sum_{s'\in\mathcal{S}} P(s'|s,a,d)\Big[r(s,a,d,s') + \max_{p_D\in\mathbf{P}_D} U(s,p_D,BR(p_D))\Big]$$
$$= \sum_{s'\in\mathcal{S}} P(s'|s,a,d)\Big[r(s,a,d,s') + \max_{p_D\in\mathbf{P}_D} p_D(s')Q^t(s')BR(p_D(s'))\Big].$$

Then from Eq. (13) we have,

$$F_1(Q^t(s,d,a)) = \hat{F}_1(Q^t(s,d,a)) - \rho^t \sum_{s'\in\mathcal{S}} P(s'|s,a,d)\tau(s,a,d,s').$$

By triangle inequality, $|F_1(Q^t(s,d,a))| \leqslant$

$$|\hat{F}_1(Q^t(s,d,a))| + |\rho^t \sum_{s'\in\mathcal{S}} P(s'|s,a,d)\tau(s,a,d,s')|. \qquad (19)$$

Next we show the map $\hat{F}_1(Q^t)$ is a contraction with respect to some weighted sup-norm under the assumption 2) of Theorem VI.3.

$$\begin{aligned}
&|\hat{F}_1(Q_1^t) - \hat{F}_1(Q_2^t)| \\
&= \Big|\sum_{s'\in\mathcal{S}} P(s'|s,a,d)\Big[\max_{p_D\in\mathbf{P}_D} p_D(s')Q_1^t(s')BR(p_D(s')) \\
&\qquad - \max_{p_D\in\mathbf{P}_D} p_D(s')Q_2^t(s')BR(p_D(s'))\Big]\Big| \\
&\leqslant \sum_{s'\in\mathcal{S}} P(s'|s,a,d) \max_{p_D\in\mathbf{P}_D}\Big|\Big[p_D(s')Q_1^t(s')BR(p_D(s')) \\
&\qquad - p_D(s')Q_2^t(s')BR(p_D(s'))\Big]\Big| \\
&= \sum_{s'\in\mathcal{S}} P(s'|s,a,d) \max_{p_D(s')\in\mathbf{P}_D(s')} \sum_{d'\in\mathcal{A}_D(s')} p_D(s',d') \\
&\qquad |Q_1^t(s',d',BR(p_D(s')) - Q_2^t(s',d',BR(p_D(s'))| \\
&\leqslant \sum_{s'\in\mathcal{S}} P(s'|s,a,d) \max_{k_2} |Q_1^t(k_2) - Q_2^t(k_2)|, \qquad (20)
\end{aligned}$$

where $k_2 = 1,\ldots,n_2$ represents the indexing of (s,d',a',s') tuples, $s,s'\in\mathcal{S}$, $d\in\mathcal{A}_D(s)$, $a\in\mathcal{A}_A(s)$, $d'\in\mathcal{A}_D(s')$ and $a'\in BR(d')\subseteq\mathcal{A}_A(s')$.

Note that, if we can substitute the term $\sum_{s'\in\mathcal{S}} P(s'|s,a,d)$ in Eq. (20) by an appropriate vector, then we can relate Eq. (20) to its weighted sup-norm (see Definition VI.4). Therefore, consider a semi-Markov decision process (SMDP) with the same probability structure as in Γ and a minimizing player whose action set is given by $(d,a)\in\mathcal{A}_D(s)\times\mathcal{A}_A(s)$. Let $r_D(s,d,a,s') = -1$ and $\tau(s,d,a,s') = 1$ for all instances of (s,d,a,s'). When stationary strategies are considered, the following equation holds for the aforementioned SMDP under assumption 2) given in Theorem VI.3 and Proposition 2.2 in [30].

$$\begin{aligned}
\bar{Q}(k_1) &= -1 + \min_{k_1} \sum_{s'\in\mathcal{S}} p(s'|k_1)\bar{Q}(k_2) \\
&\leqslant -1 + \sum_{s'\in\mathcal{S}} p(s'|k_1)\bar{Q}(k_2) \leqslant -1, \qquad (21)
\end{aligned}$$

where $k_1 = 1,\ldots,n_1$ represents the indexing of tuples, (s,d,a,s'). Then define a vector ε such that $\varepsilon(k_1) = -\bar{Q}(k_1)$. The following holds for ε from Eq. (21).

$$\sum_{s'\in\mathcal{S}} p(s'|k_1)\varepsilon(k_2) \leqslant \varepsilon(k_1) - 1 \leqslant \hat{\varepsilon}\varepsilon(k_1), \qquad (22)$$

where $\hat{\varepsilon} = \max_{k_1} \frac{\varepsilon(k_1)-1}{\varepsilon(k_1)} \leqslant 1$.

Using the vector ε and weighted sup-norm (see Definition VI.4) in Eq. (20), we observe

$$|\hat{F}_1(Q_1^t) - \hat{F}_1(Q_2^t)| \leqslant \sum_{s'\in\mathcal{S}} P(s'|k_2)\varepsilon(k_2)||Q_1^t - Q_2^t||_\varepsilon,$$

Since $\sum_{s'\in\mathcal{S}} p(s'|k_1)\varepsilon(k_2) \leqslant \hat{\varepsilon}\varepsilon(k_1)$ by Eq. (22),

$$\begin{aligned}
|\hat{F}_1(Q_1^t) - \hat{F}_1(Q_2^t)| &\leq \hat{\varepsilon}\varepsilon(k_1)||Q_1^t - Q_2^t||_\varepsilon \\
||\hat{F}_1(Q_1^t) - \hat{F}_1(Q_2^t)||_\varepsilon &\leq \hat{\varepsilon}||Q_1^t - Q_2^t||_\varepsilon, \qquad (23)
\end{aligned}$$

where $\hat{\varepsilon} < 1$.

This proves the map $\hat{F}_1(Q^t)$ is a contraction under weighted sup-norm with respect to the vector ε given in assumption 2) of Theorem VI.3. This completes the proof. □