Dynamic Information Flow Tracking for Detection of Advanced Persistent Threats: A Stochastic Game Approach

Shana Moothedath, Member, IEEE, Dinuka Sahabandu, Joey Allen, Andrew Clark, Member, IEEE, Linda Bushnell, Fellow, IEEE, Wenke Lee, Fellow, IEEE, and Radha Poovendran, Fellow, IEEE

Abstract—Advanced persistent threats (APTs) are stealthy attacks by intelligent adversaries. This article studies the detection of APTs that infiltrate cyber systems and compromise specifically targeted data and/or infrastructures. Dynamic information flow tracking is an information trace-based detection mechanism against APTs that tags suspicious information flows in the system and performs security analysis for unauthorized use of tagged data. In this article, we develop an analytical model for resource-efficient detection of APTs using an information flow tracking game. The game is a nonzero-sum, turn-based, stochastic game with asymmetric information as the defender cannot distinguish whether an incoming flow is malicious or benign. The payoff functions of the game capture the cost for performing security analysis and the rewards and penalties received by the players. We analyze the equilibrium of the game and prove that a Nash equilibrium is given by a solution to the minimum capacity cut set problem on a flow-network derived from the system. The edge capacities of the flow-network are obtained from the cost of performing security analysis. Finally, we implement our algorithm on a real-world dataset for a data exfiltration attack augmented with false-negative and false-positive rates and compute an optimal defender strategy.

Index Terms—Advanced persistent threats (APTs), information flow tracking, minimum-cut problem, stochastic games.

I. INTRODUCTION

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. The intention of an APT is to monitor system activity and continuously mine highly sensitive data, rather than causing damage to the system or organization. APT attacks consist of multiple stages that are initiated by an initial compromise and reconnaissance stage to establish a foothold in the system. Attackers then progress through the system, exploring and planning an attack strategy to obtain the desired data. This is followed by exfiltration of sensitive data, which is continued over a long period of time. Defending against APTs is a challenging task since APTs are specifically designed to evade conventional security mechanisms such as firewalls, antivirus software and intrusion-detection systems (IDSs) that rely on signatures and rules whenever it observes an unauthorized use of tagged data. These heuristic rules are typically defined by practicing experts based on the historical attack knowledge and knowledge about nominal system behavior.

After an APT attack, when the consequences of the attack have been identified, a forensic investigation is conducted using the data from the log files. The purpose of this postmortem investigation is to understand the incident’s root cause, and construct appropriate defense strategies against such APTs and its variants [1]. During the forensic analysis, the system log data are analyzed in an offline setting.

Dynamic information flow tracking (DIFT) [2] is a widely used detection mechanism for offline analysis of APT. DIFT uses the information traces recorded in the system log for performing the security analysis [1]. The key idea behind DIFT is that it tags all suspicious input/data channels and tracks the propagation of the tagged information flows through the system. DIFT generates security analysis using a prespecified set of heuristic security rules whenever it observes an unauthorized use of tagged data. These heuristic rules are typically defined by practicing experts based on the historical attack knowledge and knowledge about nominal system behavior [1].
Although, security rules incorporated in the DIFT mechanism cover a wide range of attacks, these security rules may not be capable of verifying the authenticity of information flows against all possible attacks, resulting in the generation of false negatives (FNs) and false positives (FPs). Incorrectly identifying a benign (nominal user) flow as malicious is an FP and failing to identify a malicious (adversarial) flow is an FN. For instance, while the security rules for buffer overflow protection [3] can be verified accurately, the security rules for web application vulnerabilities [4] cannot be accurately verified. Consequently, attacks that exploit web application may lead to incorrect conclusions by DIFT thereby generating FPs and FNs.

In addition, limited availability of resources for defense along with the performance and memory overhead imposed by the defense mechanism on the system demands a resource-efficient detection technique. An analytical model of DIFT that captures the cost for performing security analysis and the effectiveness of flow-tagging mechanisms will facilitate the tradeoff between the effectiveness of defense and the resource efficiency. Furthermore, such a model would enable the design of optimal security strategies while taking into account the generation of FP and FN.

In this article, we provide an analytical model to enable DIFT to optimally select locations in the system to perform security analysis so as to maximize the probability of detection while minimizing the cost of detection. Our framework is based on the following insights. First, although both APT and DIFT are unaware of the other player’s strategy, the effectiveness of the DIFT depends on the APT’s strategy and the APT’s probability of evading detection depends on the DIFT’s strategy. This strategic interaction motivates a game-theoretic approach. Second, the efficiency of detection also depends on the effectiveness of performing security analysis at different locations in the system, and hence, is determined by rates of FP and FN. Third, the game unfolds at multiple states between the entry points and the exit points of the attack where each state corresponds to the position of the tagged flows in the system. At each state, the defender decides whether to analyze a tagged flow and which one of the tagged flows to analyze (to avoid generation of FP and FN) while the adversary decides which process to transition to, at the cost of spending the defense resources. We formulate a stochastic game model that is played on an information flow graph (IFG). An IFG is a directed graph that expresses the history of a system’s execution in terms of the spatio-temporal relationships between processes and subjects. The contributions of this article are the following.

1) We model the interaction of APT and DIFT as a nonzerosum, two-player, turn-based stochastic game (G) with finite state and action spaces. In the APT versus DIFT game, the state of the game is the nodes of the IFG at which the tagged flows arrive. The location of the tagged flows is known to both players (APT and DIFT), and this makes the state of the game observable. However, each player is unaware of the other player’s strategy. Also, before performing security analysis, DIFT cannot distinguish malicious and benign flows. This results in an asymmetric information structure.

2) We analyze Nash equilibrium (NE) strategy of the game and show that an NE can be obtained from a solution to the minimum capacity cut-set problem on a flow-network constructed from the IFG of the system.

3) We implement our algorithms and results on real-world data obtained for a data exfiltration attack using the refinable attack investigation (RAIN) system [1] augmented with FN and FP rates of the system.

The rest of this article is organized as follows. Section II presents related work. Section III introduces preliminary concepts on IFG, APT attack, and a description of DIFT-based defense mechanism. Section IV describes the formulation of the turn-based stochastic game model. Section V gives the solution concept of the game and the equilibrium analysis results for computing optimal strategies of the players. Section VI illustrates the numerical results using data exfiltration attack dataset collected using RAIN and augmented with FN and FP rates. Finally, Section VII concludes this article.

II. RELATED WORK

Stochastic games model interactions of multiple agents that jointly control the evolution of states of a stochastic dynamical system [5]. Stochastic games have been widely used to model security games [6], economic games [7], and resilience of cyber-physical systems [8], where each player tries to maximize its individual payoff. A brief overview of the existing results in stochastic games is given in [9].

Stochastic games have been used to address system security-related problems. The interaction between malicious attackers and the IDS is modeled using a stochastic game in [10]. A non-zero-sum stochastic game model is given in [11] to model the network security configuration problem in the distributed IDS. Then, a value iteration-based algorithm is proposed in [11] to find a ε-NE for an attacker model where multiple adversaries simultaneously attack a network. A zero-sum stochastic game is formulated in [12] for IDS in a communication or computer network with interdependent nodes and correlated security assets and vulnerabilities. Note that, Hespanha and Prandini [13] considered a finite-horizon game and the authors in [10], [11], and [12] dealt with zero-sum stochastic games for the IDS.

Game-theoretic models for resource-efficient detection of APTs are given in [14], [15], [16], and [17]. DIFT models with fixed trapping nodes are introduced and analyzed in [14] and [15]. Moothedath et al. [16] extended the models in [14] and [15] by considering a DIFT model that selects the trapping nodes in a dynamic manner rather than being fixed and proposed a min-cut-based solution approach. Sahabandu et al. [17] considered the detection of APTs when the attack consists of multiple attackers possibly with different capabilities and analyzed the best responses of the players. Note that the focus in [14], [15], [16], and [17] is resource-efficient detection of APTs and they do not consider the FPs and FNs of the detection mechanism. In other words, the game models in [14], [15], [16], and [17] assume that security analysis performed by DIFT can verify the authenticity of tagged information flows accurately.

A stochastic game model for detecting APTs was recently introduced in the conference version of this work [18]. The proposed method in [18] analyzed a discounted stochastic game and presented a value iteration based algorithm to obtain an ε-NE of the discounted game. Due to the discounted nature of the payoff, the NE analysis of the game in [18] reduces to a nonlinear program (NLP). Solving an NLP is computationally challenging and the convergence of the algorithm in [18] is not guaranteed. This necessitates an alternate solution approach for solving the DIFT versus APT game. In this article, we
solve an average reward (undiscounted) stochastic game and present a min-cut-based approach with guaranteed convergence to compute optimal defense strategies.

In this article, we consider that the trapping nodes are generated dynamically and we extend our previous results on min-cut analysis in [16] in the following aspects. Since there can be a wide range of possible APT attacks, we recognize that the security rules of DIFT may not be capable of verifying the authenticity of the information flows accurately thereby resulting in the generation of $FP$ and $FN$. Consequently, the game model in this article is stochastic unlike in [16]. We introduce a model with multiple information flows out of which one is malicious and the remaining are benign unlike in [16], which dealt with one information flow. Note that, while DIFT knows there is exactly one malicious flow in the system, it is unaware which flow is the malicious flow before performing security analysis. This results in an asymmetric information between players, hence, the game is imperfect information game. In the current game model, although an information flow is concluded as benign at an initial inspection, it can be inspected again later during its propagation. The model in [16] inspects the flow exactly once as $FN$ and $FP$ rates are assumed to be zero. However, DIFT may analyze a flow multiple times and this will result in added resource cost. The game model considered in this article captures the added resource cost incurred from reevaluation of the flows.

III. PRELIMINARIES

In this section, we first describe the graphical representation of the system, denoted as the IFG, and then, present the models of the attacker and the flow tracking-based defense.

A. Information Flow Graph (IFG)

Let $\mathcal{G} = (V_G, E_G)$ represents the IFG of the system. $V_G = \{v_1, \ldots, v_N\}$ consists of the processes (e.g., an instance of a computer program), files, and objects in the system and $E_G \subseteq V_G \times V_G$ represents the information flows (directed) in the system from one node to the other. IFG is a weakly connected graph. IFG-based auditing is heavily desired by large enterprises and government agencies to detect APTs. We perform our game-theoretic analysis on the IFG of the system and use DIFT as the defense mechanism to detect APTs.

B. Attack Model: APTs

APTs are intelligent attacks with specific targets. APTs are stealthy attacks that perform data exfiltration at an ultralow rate to avoid detection. Unlike classical malware, APT campaigns tend to involve multiple hosts, multiple systems, and extend over a long period of time, up to several months [19]. APTs are characterized by their abilities to render existing security mechanisms ineffective. APTs can evade security protection because existing mechanisms lack sufficient visibility into user, program, and operating system activities to ascertain the authenticity of an activity and the provenance of its data. The timeline and key stages of the APT lifecycle are described as follows and presented in Fig. 1.

1) **Reconnaissance:** During the reconnaissance phase, the attacker will try to gain information about the system, such as what nodes are accessible on the system and the security defenses that are being used.

2) **Initial compromise:** During the initial compromise stage, the attacker’s goal is to gain access to an enterprise’s network. In most cases, the attacker achieves this by exploiting a vulnerability or a social engineering trick, such as a phishing email.

3) **Foothold establishment:** Once the attacker has completed the initial compromise, it will establish a persistent presence by opening up a communication channel with their command-and-control server.

4) **Lateral movement:** The attacker will increase its control of the system by moving laterally to new nodes in the system.

5) **Target attainment:** Next, the attacker will try and escalate its privileges, which may be necessary in order to access sensitive information, such as a proprietary source code or customer information.

6) **Attack completion:** The final goal of the attacker is to deconstruct the attack, hopefully in a way to minimize its footprint in order to evade detection. For example, attackers may rely on deleting the system’s log.

Let the set of possible entry points of the APT be denoted as $\lambda \subset V_G$. During an attack, the goal of the APT is to capture a subset of nodes of the IFG referred to as destinations and denoted as $D \subset V_G$. Here, $\lambda \cap D = \emptyset$. Once a foothold is established in the system, APT tries to elevate the privileges and proceeds to the destinations through more internal compromises and performs data exfiltration at an ultralow rate. In order to achieve this the APT performs operations in the system to transition through the nodes in $V_G$ and arrive at some node in $D$. In other words, the adversary (APT) selects paths in the IFG and performs transitions along the paths to reach the set $D$ from the set $\lambda$.

The adversary in the DIFT versus APT game has the following properties. We consider an APT attack that generates a single malicious information flow. The malicious flow originates at an entry point of the attack and the objective of the flow is to reach a destination. The state of the game is the set of nodes of the IFG at which the tagged flows arrive. APT observes the tagged flows, and hence, the state of the game is observable to APT. However, APT is unaware of the actions and the strategy of DIFT. Thus, the information structure of APT is asymmetric with respect to that of the defender.

C. Defender Model: DIFT

The objective of the defender is to prevent any adversarial information flow traversing to the destination nodes. DIFT is a dynamic taint analysis-based detection mechanism that consists
of three main components: tag sources, tag propagation rules, and trapping nodes. Trapping nodes are processes and objects (files and network endpoints) in the system that are considered as suspicious sources of information by the system. All data originating from a tag source are labeled or tagged and DIFT tracks the propagation of a tagged data through the system. Propagation of a tagged information flow through the system results in tagging of more information flows based on the tag propagation rules specified by the DIFT. Tag propagation rules are defined based on two kinds of information flows in the system: explicit information flows and implicit information flows.

During the execution of a program in the system, DIFT keeps track of the tagged information flows and generates trapping nodes for any unauthorized use of tagged data that indicate a possible attack. DIFT invokes security analysis at the trapping nodes using the specified security rules to verify the authenticity of the tagged flow, thereby, concluding whether there is an attack or not. These security rules are pre-specified depending on the application running on the system. Note that tagged (suspicious) flows consist of both benign and malicious flows. As the specified security rules may not necessarily cover all possible attacks by APTs, DIFT generates FPs and FNs. An optimal selection of trapping nodes in the IFG is, hence, critical to detect a wide range of attacks in the system with minimum FPs and FNs. Note that while tag sources and tag propagation rules are known to the attacker, the trapping nodes are dynamically generated during the operation of the system, and hence, is unknown to the attacker. Fig. 2 shows a schematic diagram of the DIFT-based detection framework considered in this article.

The defender in the DIFT versus APT game has the following properties. The state of the game, which is the nodes of the IFG at which the tagged flows arrive, is known to DIFT. Thus, DIFT observes the state of the game. However, DIFT cannot distinguish a malicious and a benign flow before performing security analysis. Also, while DIFT knows that there is an APT in the system, DIFT is unaware of the actions and the strategy of APT.

**IV. TURN-BASED APT VERSUS DIFT GAME G**

In this section, we model the interaction of the APT with the system during the different stages of the attack as a dynamic stochastic game between the defender (DIFT) and the adversary (APT). Let $P_D$ be the defender player and $P_A$ be the adversarial player. We consider a nonzero-sum turn-based game. The information structure of the players is asymmetric as the defender does not know if a tagged (suspicious) flow is malicious or benign while the adversarial player knows this information.

The game $G = (S, s_0, \Sigma, \Delta, P)$ unfolds on a finite state space, $S$, with initial state, $s_0$, finite action space of players, $\Sigma$, labeled transitions $\Delta \subseteq S \times \Sigma \times S$, and transition probability matrix $P$.

**A. State Space**

We consider a turn-based game with the state of the game at time $t \in \{0, 1, 2, \ldots \}$ denoted by the random variable $s_t$. Let $T$ denote the time horizon of the game, i.e., the game ends at time $T$ and $s_t = s_T$ for all $t \geq T$. The state space $S$ is partitioned into two sets, $S_A$ and $S_D$, such that $S_A \cap S_D = \emptyset$ and $S_A \cup S_D = S$. The states that belong to the set $S_A$ are referred to as the adversary-controlled states and the states in $S_D$ are referred to as the defender-controlled states. Specifically, $S_A$ is the subset of states at which player $P_D$, where $x \in \{A, D\}$, controls the transitions. The state of the game at time $t$ is the state of the system that corresponds to the position of the tagged (suspicious) flows at time $t$. Let $W$ be the number of tagged flows that arrive into the system at time $t = 1$. In practice (from the analysis of log data), only a small fraction of nodes in the system receive a system call at the same time. Hence, in our model, we assume that $W << N$. We denote the state of the game at time $t$ using the location of the $W$ flows in the system and one additional bit that represents whether that state belong to $S_A$ or $S_D$. Specifically, at time $t$, $s_t = (x, v_{i_1}, \ldots , v_{i_W})$, where $v_{i_k} \in V_F \cup \{\phi, \tau\}$ and $x \in \{A, D\}$. Here, $x = A$ implies that $s_t \in S_A$, and $x = D$ implies that $s_t \in S_D$. Moreover, $v_{i_k} = \phi$ if the $k$th information flow drops out, and $v_{i_k} = \tau$ if the $k$th information flow is trapped by DIFT. At time $t \geq 1$, the state of the game is $s_t = (x, v_{i_1}, \ldots , v_{i_W})$, where $x \in \{A, D\}$ and $v_{i_k} \in V_F \cup \{\phi, \tau\}$. Note that, out of the $W$ flows one is a malicious flow and the remaining $(W - 1)$ are benign flows. For notational convenience, we consider the first flow is malicious (note that, defender does not have this information). While DIFT observes all the $W$ flows, it cannot distinguish malicious and benign flows. APT, on the other hand, knows which flow is the malicious flow. Thus, in the DIFT versus APT game, APT and DIFT have asymmetric information.

Also, corresponding to a state $s_t$ if $v_{i_k} = \phi$, for some $k \in \{1, \ldots , W\}$, then for all time $t' > t$, the state corresponding to the $k$th flow remains $\phi$, i.e., if a flow is dropped at time $t$, it remains dropped throughout the rest of the game.

**B. Action Spaces**

We first construct a directed flow-network $F = (V_F, E_F)$ from the IFG $G$ by introducing a source node $s_F$ with an outgoing edge to all the entry points and a sink node $t_F$ with an incoming edge from all the destination nodes. Here, $V_F = V_G \cup \{s_F, t_F\}$ and $E_F = E_G \cup \{s_F \times \lambda\} \cup \{(D \times t_F)\}$. 

**Definition IV.1:** An attack path in the flow-network $F$ is a simple directed path\(^1\) from $s_F$ to $t_F$. The set of attack paths in $F$ is denoted as $\Omega_D$.

The objective of the APT attack is to capture a destination node in set $D$. To achieve the same, APT chooses transitions along a path from the set $\lambda$ to the set $D$, which translates into an attack path in the flow network $F$. We note that any IFG that contains cycles can be converted into an acyclic IFG without

\(^1\) A directed path is said to be a simple directed path if there are no cycles or loops in the path.
losing any causal relationships between the components given in the original IFG. One such dependency-preserving conversion is node versioning [22]. We consider an acyclic IFG, and hence, all $\omega \in \Omega_D$ are simple directed paths.

Using the construction of the flow network, we denote the initial state of the game at $t = 0$ is denoted as $s_0 = (A, s_1, \ldots, s_t)$, since at $t = 0$ all flows originate at the source node $s_1$. At a state $s_t$, either $\mathcal{P}_D$ or $\mathcal{A}_D$ chooses an action from their respective action sets denoted by $\mathcal{A}_D$ and $\mathcal{A}_A$, depending on whether $s_t \in \mathcal{S}_D$ or $s_t \in \mathcal{S}_A$. If $s_t \in \mathcal{S}_A$, the adversary decides whether to quit the attack by dropping the information flow or to continue the attack. If the adversary decides not to quit the attack, then it selects which neighboring node of the IFG to transition from the current node so as to reach set $\mathcal{D}$. In other words, the adversary either drops the malicious flow or performs a transition along a path in $\Omega_D$, and hence, $\mathcal{A}_A := \{\phi\} \cup \mathcal{V}_G$.

Specifically, at state $s_t = (A, v_{i_1}, \ldots, v_{i_w})$, where $v_{i_k} \in \mathcal{V}_D$ and $v_{i_1} \notin \mathcal{D}$, $\mathcal{A}_{A}(s_t) \in \{\phi\} \cup \mathcal{N}(v_{i_1})$, where $\mathcal{N}(v_{i_1}) = \{v_{i_2} \in \mathcal{V}_D : (v_{i_1}, v_{i_2}) \in E_{G}\}$. On the other hand, if $s_t \in \mathcal{S}_D$, the action of the defender is to decide whether to perform security analysis on an information flow or not. If the defender chooses to perform security analysis, it also selects a flow to analyze. A node at which security analysis is performed is referred to as a trapping node. Consider a state $s_t = (D, v_{i_1}, \ldots, v_{i_w})$ and let defender decides to perform security analysis on the flow at $v_{i_1}$. Then, $v_{i_1}$ is chosen as a trapping node. Thus, $\mathcal{A}_D := \{0\} \cup \mathcal{V}_G$, where 0 represents not analyzing any flow. The defender performs security analysis on at most one flow in a state as there is only malicious flow and also it is not possible to perform analysis on the information flows at the entry points of the attack, $\lambda$. Thus, at state $s_t = (D, v_{i_1}, \ldots, v_{i_w})$, $\mathcal{A}_{D}(s_t) \in \{0\} \cup \{v_{i_k} : v_{i_k} \notin \lambda, k \in \{1, \ldots, w\}\}$. We also note that the information flow chosen by the DIFT to perform security analysis can be either the malicious flow or a benign flow.

A state $s_t$ is said to be an absorbing state if the game terminates at $s_t$ and the action sets of the players are empty, i.e., $\mathcal{A}_A(s_t) = \mathcal{A}_D(s_t) = \emptyset$. In $G$, a state $s_t = (x, v_{i_1}, \ldots, v_{i_w})$, where $x \in \{A, D\}$, is absorbing if one of the three cases holds:

1) $v_{i_1} \in \mathcal{D} \subseteq \mathcal{V}_G$;
2) $v_{i_k} = \tau$ for some $k \in \{1, \ldots, w\}$;
3) $v_{i_k} = \phi$ for all $k \in \{1, \ldots, w\}$.

Case 1) corresponds to adversary reaching a destination, case 2) corresponds to the case where DIFT performed security analysis on an information flow and also concluded it as malicious, i.e., a flow is trapped, and case 3) corresponds to all W flows dropping out.

C. State Transitions

Definition IV.2: Time $t$ is said to be the termination time of the game $G$ if it satisfies one of the following condition: 1) $t = T$ and 2) the state of the game $s_t = (x, v_{i_1}, \ldots, v_{i_w})$, where $x \in \{A, D\}$, is an absorbing state.

We use the notation $W$ to denote the number of tagged information flows that are recorded at the first instant of time, i.e., $t = 1$. The log recording system (RAIN) used in our work uses the system call in [23] to create a timestamp for each audit log. At every instant of the log recording, RAIN records at most one system call for each system component (processes, files). As a result, the W information flows arrive at distinct nodes of the IFG as it is not possible for multiple flows to arrive at one node at the same instant of time. Let the state of the game at $t = 1$ be $s_t = (D, v_{i_1}, \ldots, v_{i_w})$. Here, $\{v_{i_1}, \ldots, v_{i_w}\}$ are the nodes of the IFG at which the flows arrive. The DIFT-based defense mechanism observes the flows. The player $\mathcal{P}_D$ now chooses an action from the set $\mathcal{A}_D(s_t) \subseteq \{0, v_{i_1}, \ldots, v_{i_w}\}$. At state $s_t$, there are the following three possibilities:

1) $\mathcal{P}_D$ does not analyze any flow;
2) $\mathcal{P}_D$ analyzes the malicious flow;
3) $\mathcal{P}_D$ analyzes a benign flow.

Based on the action chosen by $\mathcal{P}_D$, i.e., $\mathcal{A}_D(s_t)$, the next possible state of the game $s_{t+1}$ is defined. At $s_{t+1}$, $\mathcal{P}_A$ has two possibilities to choose from: 1) to quit the flow, $\phi$, and 2) to transition to an out-neighbor of node $v_{i_1}$, $\mathcal{N}(v_{i_1})$, i.e., $\mathcal{A}_A(s_{t+1}) \in \{\phi\} \cup \mathcal{N}(v_{i_1})$. Based on the action of $\mathcal{P}_A$ and the distribution of the benign flows in the system, the next state $s_{t+2}$ is arrived. At $t + 2$, $\mathcal{P}_D$ again chooses its action and the game continues in a turn-based fashion. Note that the DIFT will analyze only one information flow at a time $t$ as we consider a single malicious flow in the system.

For case 1), the action of $\mathcal{P}_D$ is $\mathcal{A}_D(s_t) = 0$, i.e., DIFT does not perform security analysis on any of the W information flows. Then, the next state of the game $s_{t+1} = (A, v_{i_1}, \ldots, v_{i_w})$. For case 2), $\mathcal{P}_D$ chooses action $\mathcal{A}_D(s_t) = v_{i_1}$ and the next state of the game is $s_{t+1} = (A, v_{i_1}, \ldots, v_{i_w})$ with probability $FN$ and $s_{t+1} = (A, \tau, v_{i_2}, \ldots, v_{i_w})$ with probability $1 - FN$. For case 3), the player $\mathcal{P}_D$ chooses action $\mathcal{A}_D(s_t) = v_{i_1}, k \neq 1$, i.e., $k \in \{2, \ldots, W\}$. Then, the next state of the game is $s_{t+1} = (A, v_{i_1}, \ldots, v_{i_{k-1}}, \tau, v_{i_{k+1}}, \ldots, v_{i_w})$ with probability $FP$ and $s_{t+1} = (A, v_{i_1}, \ldots, v_{i_w})$ with probability $1 - FP$. Here, $FN$ and $FP$ are the rate of FNs and FPs in the DIFT-architecture, which are empirically computed. The values of $FP$ and $FN$ depend on the security rules and vary across different DIFT-architectures and they determine the transition probabilities $P$ of the stochastic game. In order to reduce FPs and FNs in the system, the security rules must be capable of identifying the behavior and predicting the intent of information flows, i.e., determine whether an unknown flow is indeed malicious, or whether it is benign flow that is exhibiting malware-like behavior [24].

If $t + 1$ is not a terminating time instant, then the adversarial player $\mathcal{A}_A$ chooses action $\mathcal{A}_A(s_{t+1}) \in \{\phi\} \cup \mathcal{N}(v_{i_1})$. The remaining $W - 1$ flows follow the benign flow distribution in the system, which is computed empirically from the nominal system operation and known. Let $\pi_{\mathcal{P}} : v_i \in \mathcal{V}_G \rightarrow [0, 1]^{[\mathcal{N}(v_i) + 1]}$ denote the benign flow distribution. If the action of $\mathcal{P}_D$ is $\phi$, then $s_{t+2} = (D, \phi, v_{i_1}, \ldots, v_{i_w})$, where $v_{i_2}, \ldots, v_{i_w}$ depend on the distribution $\pi_{\mathcal{P}}$. Here, $\{v_{i_2}, \ldots, v_{i_w}\} \subseteq \{\phi\} \cup \mathcal{V}_G$. Note that a benign flow can also drop out. Now $\mathcal{P}_D$ chooses its action and the game continues. To summarize, given $s_t = (D, v_{i_1}, \ldots, v_{i_w})$ (w.p stands for with probability), state transitions and the corresponding transition probabilities, given by $P$, are

$$s_{t+1} = \begin{cases} (A, v_{i_1}, \ldots, v_{i_w}), & \text{w.p1 if } \mathcal{A}_D(s_t) = 0 \\ (A, v_{i_1}, \ldots, v_{i_w}), & \text{w.pFN if } \mathcal{A}_D(s_t) = v_{i_1} \\ (A, \tau, \ldots, v_{i_w}), & \text{w.pFP if } \mathcal{A}_D(s_t) = v_{i_1}, k \neq 1 \\ (A, v_{i_1}, \ldots, v_{i_{k-1}}, \tau, \ldots, v_{i_w}), & \text{w.p1 - FN if } \mathcal{A}_D(s_t) = v_{i_1}, k \neq 1 \\ (A, v_{i_1}, \ldots, v_{i_w}), & \text{w.p1 if } \mathcal{A}_D(s_t) = v_{i_k}, k \neq 1 \\ \end{cases} \quad (1)$$
Fig. 3. Schematic diagram showing the turn-based game $G$. At state $s_t \in S_G$, the defender player $P_D$ chooses an action and the game transitions to the next state $s_{t+1}$. At time $t+1$, the adversarial player $P_A$ observes the current state $s_{t+1} \in S_A$, and then, chooses an action based on which the next state $s_{t+2} \in S_D$ is defined. At $t+2$, $P_D$ again chooses its action and the game continues.

and

$$s_{t+2} = \begin{cases} (D_i, v_{j_1}, \ldots, v_{j_{iW}}), & \text{if } A_A(s_{t+1}) \in N(v_{i_t}) \\ (D, \phi, v_{j_2}, \ldots, v_{j_{jW}}), & \text{if } A_A(s_{t+1}) = \phi. \end{cases}$$ (2)

A schematic diagram that illustrates the transitions in the turn-based game $G$ is given in Fig. 3.

D. Strategies of the Players

A strategy is a rule that each player uses to select actions at every step of the game. We consider mixed (stochastic) and behavioral player strategies. Since the strategy is mixed, at a state, $P_D$ and $P_A$ select an action from the action set $A_D$ and $A_A$, respectively, based on some probability distribution. Furthermore, since the strategy is behavioral, the probability distribution on the action set at a time instant $t$ depends on all the states traversed by the game and the actions taken by the players until time $t$.

Let $d_t$ and $a_t$ be the action of the defender and the attacker, respectively, at time $t$. We denote the information available to the players $P_D$ and $P_A$ at time $t \leq T$ by $Y_t$ and $Z_t$, respectively. Then,

$$Y_t := \{s_0, d_0, s_1, d_1, \ldots, s_{t-1}, d_{t-1}, s_t\}$$
$$Z_t := \{s_0, a_0, s_1, a_1, \ldots, s_{t-1}, a_{t-1}, s_t\}. \quad (3)$$

We denote the set of all possible outcomes for $Y_t$ and $Z_t$ at time $t$ using $Y_t^*$ and $Z_t^*$, respectively. Let the set of all pure behavioral strategies of $P_D$ and $P_A$ for game $G$ be $P_D^\ast$ and $P_A^\ast$, respectively. Define $\Delta_P(D_D)(\Delta_P(A))$ as the simplex of $P_D$ ($P_A$), i.e., the set of probability distributions over $P_D$ ($P_A$). A mixed behavioral strategy of player $P_D$ is a mapping given by $\gamma_D^* \rightarrow \Delta_P(D_D)$, for $t \geq 0$, where $\Delta_P(D_D)$ denote the set of all mixed behavioral strategies for $P_D$. Similarly, a mixed behavioral strategy of player $P_A$ is given by the mapping $\gamma_A^* \rightarrow \Delta_P(A)$, for $t \geq 0$, where $\Delta_P(A)$ denote the set of all mixed behavioral strategies for $P_A$. Furthermore, let the set of all mixed policies of $P_D$ and $P_A$ for game $G$ be $P_D^\ast$ and $P_A^\ast$, respectively. Then, a policy $p_D \in P_D^\ast$ and $p_A \in P_A^\ast$ are said to be pure or deterministic policy if all the entries of the vectors $p_D$ and $p_A$, respectively, belong to the set $\{0, 1\}$.

E. Payoffs to the Players

The payoff functions of $P_D$ and $P_A$ are denoted by $U_D$ and $U_A$, respectively. $U_D$ consists of three components:

1) resource cost $C_D(v_i) < 0$ for performing security analysis of a flow at a node $v_i \in G$;

2) penalty $\beta_D < 0$ for adversary reaching a destination node in $D$;

3) reward $\alpha_D > 0$ for detecting the adversary. Similarly, $U_A$ consists of the following two components:

1) a reward $\beta_A > 0$ for reaching a destination node in the set $D$;

2) penalty $\alpha_A < 0$ for getting detected by the defender.

The reward and penalty parameters and the resource cost of DIFT and APT do not necessarily induce a zero-sum scenario where the payoff that is gained by one player is lost by the other. Therefore, we considered a nonzero sum payoff structure for the DIFT versus APT game.

Let the payoff of the player $P_x$ at an absorbing state $s_t$ be denoted as $c_x(s_t)$, where $x \in \{A, D\}$. Also, let the payoff of $P_D$ at a nonabsorbing defense-controlled state $s_t$ with action $d_t$ be $r^D(s_t, d_t)$, and the payoff of $P_A$ at a nonabsorbing adversary-controlled state $s_t$ with action $a_t$ be $r^A(s_t, a_t)$. At each state in the game, $s_t$ at time $t$, where $t < T$ and $s_t$ is a nonabsorbing state, player chooses its action ($d_t$ for $s_t \in S_D$ and $a_t$ for $s_t \in S_A$) and receives payoff $r^D(s_t, d_t)$ and $r^A(s_t, a_t)$, respectively, and the game transitions to a next state $s_{t+1}$. This is continued until they reach an absorbing state and incur $c^A(s_t)$ and $c^D(s_t)$, respectively, or the game arrives at the horizon, i.e., $t = T$. Then

$$r^A(s_t, a_t) = 0 \text{ for all } s_t, a_t. \quad (4)$$

$$c^A(s_t) = \begin{cases} \alpha_A, & s_t \in S_A, s_t = (D, \tau, \ldots, v_{iW}) \\ \beta_A, & s_t \in S_A, s_t = (A, v_{i_1}, \ldots, \tau, \ldots, v_{iW}) \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

$$c^D(s_t) = \begin{cases} \beta_D, & s_t \in S_D, s_t = (D, v_{i_1}, \ldots, v_{iW}), v_{i_k} \in D \\ \alpha_D, & s_t \in S_D, s_t = (A, \tau, \ldots, v_{iW}) \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

$$r^D(s_t, d_t) = \begin{cases} 0, & s_t \in S_D, d_t = 0 \\ C_D(v_{i_k}), & s_t \in S_D, d_t = v_{i_k}, k \in \{1, \ldots, W\}. \end{cases} \quad (7)$$

As the initial state of the game is $s_0$, for a strategy pair $(p_D, p_A)$, the expected payoffs of the players are

$$U_A(p_D, p_A) = \mathbb{E}_{s_0,p_A,p_D} \left[ \sum_{t=0}^{T} (R_t^A) \right] \quad (8)$$

$$U_D(p_D, p_A) = \mathbb{E}_{s_0,p_A,p_D} \left[ \sum_{t=0}^{T} (R_t^D) \right] \quad (9)$$

where $\mathbb{E}_{s_0,p_A,p_D}$ denotes the expectation with respect to $s_0, p_A$, and $p_D$ and from (4)–(7)

$$R_t^x = \begin{cases} r^D(s_t, d_t), & x = D, s_t \in S_D, t < T \\ r^A(s_t, a_t), & x = A, s_t \in S_A, t < T \\ c^D(s_t), & s_t \in S_D, t = T \\ c^A(s_t), & s_t \in S_A, t = T. \end{cases} \quad (10)$$

We incorporate the idea of maximizing the probability of detection and minimizing the probability of adversary evading detection using the terms $p_T$ and $p_A$, respectively. Furthermore, we capture the FP s generated in the system using the term $p_{FP}$. Note that, generation of FP implies that the defender failed to capture the adversary and concluded a benign flow.
is malicious. This means that the adversary will attain the target as the malicious flow evaded detection. Given a set of strategies \( (p_D, p_A) \), where \( p_D \in P_D \) and \( p_A \in P_A \), and benign distribution \( \pi_B \), the payoff functions of the players, i.e., (8) and (9), can be rewritten using (4)–(7) and (10) as

\[
U_D(p_D, p_A) = p_D \alpha_D + (p_R + p_{FP}) \beta_D \\
+ \sum_{s \in S_D} \sum_{v_i \in s} (p_D(v_i) \mathcal{C}_D(v_i)) 
\]

(11)

\[
U_A(p_D, p_A) = (p_D \alpha_A + (p_R + p_{FP}) \beta_A). 
\]

(12)

Here, \( p_D \) is the cumulative probability that the adversarial flow is detected by the defender. The term \( p_D \alpha_D \) captures the reward received by DIFT for detecting the attack and the term \( p_D \alpha_A \) captures the penalty incurred by APT for getting detected. Also, \( p_R \) denotes the cumulative probability that the adversarial flow reaches a destination and \( p_{FP} \) denotes the cumulative probability that a benign flow is concluded as malicious (i.e., trapped). The term \( p_R \alpha_D + p_{FP} \beta_A \) captures the total resource cost for all \( v \) and \( s \). The term \( p_D(v_i) \mathcal{C}_D(v_i) \) captures the reward received by the APT for not getting detected. Recall that \( p_D(v_i) \) denotes the probability with which DIFT selects node \( v_i \) as a security check point (trap). The term \( \sum_{s \in S_D} \sum_{v_i \in s} (p_D(v_i) \mathcal{C}_D(v_i)) \) captures the total resource cost associated with DIFT for performing security analysis. Note that \( p_D \) and \( p_R \) are functions of \( p_D \) and \( \pi_B \). Our focus is to compute a limiting average equilibrium of the non-zero-sum stochastic game.

V. COMPUTATION OF THE OPTIMAL STRATEGY

A. Solution Concept

In this subsection, we present the solution concept of the game.

**Definition VI1:** Let \( p_A : \mathbb{Z}^* \rightarrow [0, 1]^{\mathcal{A}_A} \) denote a strategy of the adversary. Also let \( p_D : \mathbb{Z}^* \rightarrow [0, 1]^{\mathcal{A}_D} \) denote a strategy of the defender, i.e., the probability of performing security analysis at a state. Then, the best response of the defender is given by

\[
\text{BR}(p_A) = \arg \max_{p_D \in P_D} \{ U_D(p_D, p_A) \}. 
\]

Similarly, the best responses of the adversary are given by

\[
\text{BR}(p_D) = \arg \max_{p_A \in P_A} \{ U_D(p_D, p_A) \}. 
\]

The best response of the defender is the set of defense strategies that maximize the payoff of the defender for a given adversarial strategy and known benign flow distribution. The best response of the adversary is a set of transition strategies, for a given defender strategy and known benign flow distribution, that maximizes the probability of adversary reaching a destination node without getting detected.

**Definition VI2:** A pair of strategies \( (p_D, p_A) \) is said to be an NE if

\[
p_D \in \text{BR}(p_A) \text{ and } p_A \in \text{BR}(p_D). 
\]

An NE is a pair of strategies such that no player can benefit through unilateral deviation, i.e., by changing the strategy, while the other player keeps the strategy unchanged. We prove existence of an NE for the APT versus DIFT game in Proposition V.5.

**Lemma V.3:** Consider the APT versus DIFT game \( G = \{ \mathbb{S}, \mathbb{S}_0, \Sigma, \Delta, P \} \). Let \( N \) be the number of nodes in the IFG and \( W \) be the number of flows that arrive into the system at time \( t = 1 \). Then, \( |S| = O(N^W) \).

**Proof:** We prove the result by showing that \( |S| = O(N^W) \) and \( |S_D| = O(N^W) \). Consider an arbitrary state \( s = (x, v_1, \ldots, v_W) \in \mathbb{S} \setminus \mathbb{S}_0 \), where \( x \in \{ A, D \} \). Here, \( v_i \in V_G \cup \{ \phi, \tau \} \) for all \( k \in \{ 1, \ldots, W \} \). All the W tagged flows arrive at distinct nodes in the set \( V_G \) of the IFG as a process or file in the system receive exactly one information flow at a particular time. Therefore, for a state \( s = (x, v_1, \ldots, v_W) \), where \( v_i \in V_G \) for all \( k \in \{ 1, \ldots, W \} \), \( i_a \neq i_b \), for \( a, b \in \{ 1, \ldots, W \} \). Also, the defender performs security analysis on one tagged flow at a time, and hence, there is no state \( s = (x, v_1, \ldots, v_W) \) with \( v_i = v_k = \tau \) for \( a \neq b \). Repetition of \( v_i \)’s is possible only for states with \( \phi \). Hence, \( s \) belongs to one of the two types: 1) \{ \( s = (x, v_1, \ldots, v_W) : v_k \in V_G \cup \{ \tau \} \) and \( 2) \{ s = (x, v_1, \ldots, v_W) : v_k \in V_G \cup \{ \phi \} \) such that \( v_i = \phi \) for at least some \( k \in \{ 1, \ldots, W \} \).

Case 1) corresponds to selecting \( W \) items from a set of \( N \) + 1 items without any repetitions. The cardinality of this set is

\[
(N+1) \cdot \frac{(N+2) \cdots (N+W-1)}{(N+1) \cdots (N+W-2)} = O(N^W). 
\]

Case 2) corresponds to states with one or more \( \phi \) entries. Note that, here repetitions are allowed only for \( \phi \). The cardinality of this set is

\[
(N+1)^2 \frac{(N+2) \cdots (N+W-1)}{(N+1) \cdots (N+W-2)} + (N+1)^W = O(N^{W-2}). 
\]

In addition, there is an initial state \( s_0 \). From (1) and (2), the number of possible cases for \( s \) is \( O(N^W) \). Since \( s = (x, v_1, \ldots, v_W) \), where \( x \in \{ A, D \} \), we get \( |S| = O(N^W) \) and \( |S_D| = O(N^W) \). As \( S = S_A \cup S_D \), we get \( |S| = O(N^W) \).

It is shown that there exists an NE for zero-sum discounted stochastic games [25]. However, the existence of NE for non-zero-sum undiscounted stochastic games is open when the time horizon is infinite, i.e., \( T = \infty \). In the following result, we prove the existence of the NE for the game \( G \). The existence is shown by proving that time horizon is indeed finite for \( G \), and then, invoking the existence of NE for finite horizon undiscounted games.

**Proposition V.4:** Let \( T \) denote the termination time of the APT versus DIFT game. The APT versus DIFT game, \( G \), terminates in at most \( 2(N-1) \) steps, i.e., \( T \leq 2(N-1) \).

**Proof:** The action set \( A_D \) of the adversary player (APT) is to choose a transition along an attack path of the flow-network \( F \) (see Definition IV.1) or to drop out at some point. Let \( \Omega_D \) denote the set of attack paths in \( F \). Consider an arbitrary attack path \( \hat{\omega} \in \Omega_D \). Recall that \( \hat{\omega} \) is a simple path, as an APT due to its stealthy behavior will not traverse in cycles through the system. Hence, the length of \( \hat{\omega} \) is at most \( N-1 \), as \( \hat{\omega} \) is a simple directed path. Thus, in any run of the game, the adversary can take at most \( N-1 \) transitions. As \( G \) is a turn-based game, the game terminates in at most \( 2(N-1) \) steps.

**Proposition V.5:** There exists an NE for the APT versus DIFT game \( G \).

**Proof:** It is shown in [13] that there exists an NE for a non-zero-sum stochastic game with asymmetric information structure, under stochastic behavioral strategies, when the time horizon is finite. Proposition V.4 proves that the APT versus DIFT game will terminate in \( 2N \) number of steps. Furthermore, the behavioral strategy space is a subset of the strategy space \( P_A \times P_D \). Hence, by the result in [13], the proof follows.

B. Solution Approach

In this section, we compute an NE of the DIFT versus APT game \( G \). We first derive the BRs of the players separately, and...
then, use this derivation to construct an NE of the overall game. Our approach is based on a minimum capacity cut formulation on a flow network constructed from the information flow graph of the system. Consider the flow-network $\mathcal{F} = \{V_F, E_F\}$, where $V_F = \{V_s \cup \{s_r, t_r\}\}$ and $E_F = E_q \cup \{s_r \times \lambda \} \cup \{D \times t_r\}$. Then, a cut of $\mathcal{F}$ is defined as follows.

**Definition V6.** In a flow network $\mathcal{F}$ with vertex set $V_F$ and directed edge set $E_F$, the cut induced by $\hat{S} \subset V_s$ is a subset of edges $\kappa(\hat{S}) \subseteq E_F$ such that for every $(u, v) \in \kappa(\hat{S}), |\{u, v\} \cap \hat{S}| = 1$. Furthermore, given edge capacity vector $c_F : E_F \to \mathbb{R}_+$, the capacity of a cut $\kappa(\hat{S})$, is defined as the sum of the capacities of the edges in the cut, i.e., $c_F(\kappa(\hat{S})) = \sum_{e \in \kappa(\hat{S})} c_F(e)$.

The (source-sink)-min-cut problem aims to find a cut $\kappa(\hat{S}^*)$ of $\hat{S}^* \subset V_s$ such that $c_F(\kappa(\hat{S}^*)) \leq c_F(\kappa(\hat{S}))$ for any cut $\kappa(\hat{S})$ of $\hat{S} \subseteq V_s$ satisfying $s_r \in \hat{S}$ and $t_r \notin \hat{S}$. The (source-sink)-min-cut problem is well studied and there exist algorithms to find a min-cut in polynomial time in $|V_F|$ [26]. In our approach to compute NE, which is detailed later in the section, we find a min-cut through nodes of the flow network instead of the edges. Hence, we transform the node version of the min-cut problem to an equivalent edge version. For this, we introduce an edge correspondence to each node in $\mathcal{F}$ except the source and sink nodes. The transformed flow network is denoted as $\tilde{\mathcal{F}} = \{\tilde{V}_F, \tilde{E}_F\}$, where $\tilde{V}_F = V_F \setminus \{s_r\}$ and $\tilde{E}_F = E_F \setminus \{s_r \times \lambda\} \cup E_q$.

$\tilde{E}_F = \{(v_i, v_j) : (v_i, v_j) \in E_q\}$, $\tilde{E}_F = \{(v_i, v_j) : (v_i, v_j) \in E_q\}$, $\tilde{E}_F = \{(v_i, v_j) : (v_i, v_j) \in E_q\}$, $\tilde{E}_F = \{(v_i, v_j) : (v_i, v_j) \in E_q\}$, $\tilde{E}_F = \{(v_i, v_j) : (v_i, v_j) \in E_q\}$.

The capacity vector associated with the edges in $\tilde{\mathcal{F}}$ is given by

$$c_{\tilde{F}}(e) = \begin{cases} C_D(v_i), & e \in E_q' \setminus \hat{S}^* \\ 0, & \text{otherwise}. \end{cases}$$

Note that $E_q'$ is a cut of $\tilde{\mathcal{F}}$ and $\sum_{e \in E_q'} c_{\tilde{F}}(e) < \infty$. Hence, any minimum capacity cut in $\tilde{\mathcal{F}}$ corresponds to edges from the set $E_q'$ as the capacity of the remaining edges is $\infty$ as shown in (13). Furthermore, the capacity of an edge in set $(v_i, v_j) \in E_q'$ corresponds to the cost of conducting the security analysis at node $v_i$. Thus, a minimum capacity cut in $\tilde{\mathcal{F}}$ corresponds to a node cut set of the IFG with minimum total cost of performing security analysis.

Let $\kappa(\hat{S}^*)$ denotes an optimal solution to the (source-sink)-min-cut problem on $\tilde{\mathcal{F}}$. Then, $\kappa(\hat{S}^*) \subseteq E_q'$ and $c_{\tilde{F}}(\kappa(\hat{S}^*)) < \infty$. The nodes corresponding to the min-cut is

$$\hat{S}^* := \{v_i : (v_i, v_i') \in \kappa(\hat{S}^*) \}.$$  

In the APT versus DIFT game, the aim of the defender is to optimally select trapping nodes in the IFG, i.e., nodes of IFG to perform security analysis, such that no adversarial flow reaches some node in $D$. In other words, defender ensures that all adversarial flows that originate in node $s_t$ gets detected before reaching node $t_r$. In order to ensure security, the defender must select at least one node in all possible paths from $s_t$ to $t_r$ as a trapping node. In the equilibrium analysis of the game, we prove that an optimal strategy of the defender is indeed to select the min-cut nodes of the flow network as trapping nodes. The objective of the adversary is to optimally choose the transitions in such a way that the probability of reaching $t_r$ is maximum. The adversary, hence, plans its transitions to select an attack path with least probability of detection.

The following result proves that an NE of game $G$ is represented by the nodes corresponding to a minimum capacity cut in the flow-network $\mathcal{F}$. Note that, the solution to the min-cut problem may not be unique. Consequently, there may exist multiple NE for the game $G$.

**Theorem V7.** Every NE $(\pi_A, \pi_D)$ of the APT versus DIFT game $G$ satisfies the following properties.

1. The defender’s strategy $\pi_D$ selects all the nodes in $\hat{S}^*$ as trapping nodes, where $\hat{S}^*$ is a set of min-cut nodes of the flow network $\mathcal{F}$.
2. The adversary’s strategy $\pi_A$ chooses transitions such that each attack path passes through exactly one node in the set $\hat{S}^*$.

We prove Theorem V7 by invoking the property that at NE, every player plays a best response against the other players simultaneously. We first present the proof the best response results, Lemmas V8, V10, and V11, and then, present the proof of Theorem V7.

**Lemma V8.** Let $\Omega_D$ be the set of all attack paths in $\mathcal{F}$. Consider a defender strategy $\pi_D$ in which only the min-cut nodes $\hat{S}^*$ of the flow-network $\mathcal{F}$ are chosen as trapping nodes with nonzero probability. Then, the best response of the adversary, $BR(\pi_D)$, is to choose the transitions in such a way that all attack paths that have nonzero probability under $BR(\pi_D)$ pass through exactly one node in $\hat{S}^*$.

**Proof:** Consider the payoff function of the adversary, $U_A(\pi_D, \pi_A) = (\pi_D \alpha_A + (\pi_r + \pi_{FF}) \beta_A)$. Here, $\pi_r$ and $\pi_{FF}$ are functions of $\pi_D$ and $\pi_A$. However, $\pi_{FF}$ depends only on $\pi_D$. Thus, for a given $\pi_D$, $\pi_{FF}$, the probabilities $\pi_r$ and $\pi_{FF}$ vary depending on $\pi_A$, however, $\pi_{FF}$ is a constant. We prove the result using a contradiction argument. Suppose the best response of $\pi_A$, is a strategy $\pi_A'$ such that there exists a path $\hat{\omega} \in \Omega_D$ that passes through two nodes, say $v_i, v_j \in \hat{S}^* \subseteq V_s$, and $\pi_A(\hat{\omega}) \neq 0$, where $\pi_A(\hat{\omega})$ is the probability of attack path $\hat{\omega}$ under $\pi_A'$. Note that $\hat{S}^*$ is a min-cut and $v_i, v_j \in \hat{S}^*$. Furthermore, $C_D(v_i) \neq 0$ and $C_D(v_j) \neq 0$. Thus, there exists at least one directed path, say $\hat{\omega}'$, from $v_i$ to $v_j$ that does not pass through any node in $\hat{S}^* \setminus \{v_i\}$. Similarly, there exists at least one directed path, say $\hat{\omega}''$, from $v_j$ to $v_i$ that does not pass through any node in $\hat{S}^* \setminus \{v_j\}$. As per the given defender strategy $\pi_D$, the only trapping node in $\hat{\omega}', \hat{\omega}''$ is $v_i, v_j$, respectively. Thus, there exists a strategy $\pi_A$ such that all paths in $\Omega_D$ with nonzero probability under $\pi_D$ pass through exactly one node in $\hat{S}^*$ and $U_A(\pi_D, \pi_A) > U_A(\pi_D, \pi_A')$ (since $\pi_{FF}$ remains same and $\pi_r$ is higher and $\pi_{FF}$ is lower under $\pi_A'$ when compared to the values under strategy $\pi_A$). This contradicts the assumption that $\pi_A'$ is a best response and completes the proof. ■

Recall that $\Omega_D$ is the set of all source-to-sink paths in the flow-network $\mathcal{F}$ (see Definition IV.1). Note that, any path in $\mathcal{F}$ that does not belong to $\Omega_D$ is not a valid attack, as the attacker cannot reach a destination node. Every attack path in $\Omega_D$, depending on the defender strategy and the transition structure of the game, induces a set of paths in the state space graph of the APT versus DIFT game. Let $\Omega$ be the set of all paths induced by $\Omega_D$. In other words, $\Omega$ is the set of all possible state transition paths in the state space graph corresponding to all attack paths in $\mathcal{F}$. Theorem V7 shows that the NE of the game is such that the nodes corresponding to the minimum capacity cut in the flow-network $\mathcal{F}$ are selected as trapping nodes by the defender. This ensures that any adversarial flow that originates in node $s_t$ gets detected before reaching node $t_r$. In order to ensure security, the defender must select at least one node in all possible paths from $s_t$ to $t_r$ as a trapping node. The equilibrium analysis of the game, we prove that an optimal strategy of the defender is indeed to select the min-cut nodes of the flow network as trapping nodes. The objective of the adversary is to optimally choose the transitions in such a way that the probability of reaching $t_r$ is maximum. The adversary, hence, plans its transitions to select an attack path with least probability of detection.
**Definition V.9:** Let the set of paths induced in the state space $S$ by the attack paths $\Omega_\mathcal{D}$ in $\mathcal{F}$ be denoted as $\Omega$. Then, the probability of selecting a path $\omega \in \Omega$, denoted by $\pi(\omega)$, is $\pi(\omega) = \pi_\omega(\omega)\pi_\mathcal{D}(\omega)$, where $\pi_\mathcal{D}(\omega)$ is the probability with which an adversary chooses $\omega$ (product of the adversary transition probabilities along path $\omega$) and $\pi_\omega(\omega)$ is the probability of benign flows in $\omega$ under distribution $\pi_\mathcal{P}$.

The following result proves that, under certain conditions, for a given adversary strategy, the best response of the defender is to select one node in every attack path as a trapping node.

**Lemma V.10:** Let $\Omega_\mathcal{D}$ be the set of attack paths in $\mathcal{F}$, $\Omega$ be the set of paths in $S$ induced by $\Omega_\mathcal{D}$, and $p_\mathcal{D}$ be a given strategy of $\mathcal{P}_\mathcal{D}$. For $\omega \in \Omega$, let $\omega(A)$ denote the set of nodes in $\omega$ corresponding to the adversarial (malicious) flow and $\omega(B)$ denote the set of nodes in $\omega$ corresponding to the benign flows. Also, let $p(\omega)$ denote the probability of detecting the adversary along path $\omega$, i.e., $p(\omega) = [1 - \prod_{i \in \omega(A)} (1 - p_D(v_i))] (1 - FN)$. Also, let $f(\omega)$ denote the probability of trapping a benign flow (FP), i.e., $f(\omega) = [1 - \prod_{i \in \omega(B)} (1 - p_D(v_i))] FP$. If the defender’s strategy satisfies the following two conditions:

1. $p(\omega) = p(\omega')$ for all $\omega, \omega' \in \Omega$;
2. $f(\omega) = f(\omega')$ for all $\omega, \omega' \in \Omega$;

then the best response of the defender, $BR(p_\mathcal{D})$, is to select with nonzero probability exactly one node in every $\omega \in \Omega_\mathcal{D}$ as a trapping node.

**Proof:** Let $\pi(\omega)$ denote the probability of a path $\omega \in \Omega$ under the given strategy $p_\mathcal{D}$ and benign distribution $\pi_\mathcal{P}$. For a path $\omega$ in the state space $S$, $\omega(A)$ denotes the set of nodes in $\omega$ corresponding to the adversarial (malicious) flow and $\omega(B)$ denotes the set of nodes in $\omega$ corresponding to the benign flows. Let $p(\omega)$ denote the probability of detecting the adversary along path $\omega$, i.e., $p(\omega) = [1 - \prod_{i \in \omega(A)} (1 - p_D(v_i))] (1 - FN)$. Also, let $f(\omega)$ denote the probability of trapping a benign flow, i.e., FP. Then $f(\omega) = [1 - \prod_{i \in \omega(B)} (1 - p_D(v_i))] FP$. The defender’s payoff is

$$U_D(p_\mathcal{D}, p_\mathcal{P}) = \sum_{\omega \in \Omega} \pi(\omega) [p(\omega)\alpha_D + (1 - p(\omega) + f(\omega))\beta_D]$$

$$+ \sum_{\omega \in \Omega} \left( \sum_{v_i \in \omega} p_D(v_i)C_D(v_i) \right).$$

Given $p(\omega)$’s and $f(\omega)$’s are equal for all $\omega \in \Omega$. Thus,

$$U_D(p_\mathcal{D}, p_\mathcal{P}) = [p(\omega)\alpha_D + (1 + f(\omega))\beta_D] \left( \sum_{\omega \in \Omega} \pi(\omega) \right)$$

$$+ \sum_{\omega \in \Omega} \left( \sum_{v_i \in \omega} p_D(v_i)C_D(v_i) \right)$$

$$= [p(\omega)\alpha_D + (1 - p(\omega) + f(\omega))\beta_D]$$

$$+ \sum_{\omega \in \Omega} \left( \sum_{v_i \in \omega} p_D(v_i)C_D(v_i) \right).$$

The aforementioned equation holds as $\sum_{\omega \in \Omega} \pi(\omega) = 1$. Consider a defender strategy $p_\mathcal{D}$, in which exactly one node in every $\omega \in \Omega_\mathcal{D}$ is chosen as the trapping node. Assume that the defender strategy is modified to $p_\mathcal{D}'$, such that more than one node in some path are chosen as trapping nodes. This variation updates the probabilities of nodes in a set of paths in $\Omega$. Note that, due to the constraints on $p(\omega)$ and $f(\omega)$ [conditions (1) and (2)], the defender’s probabilities (strategy) at two nodes in a path are dependent. Hence, for all paths $\omega \in \Omega$, whose probabilities are modified in $p_\mathcal{D}'$, $\sum_{v_i \in \omega} p_D(v_i)C_D(v_i) < \sum_{v_i \in \omega} p_D(v_i)C_D(v_i)$. This holds as the probability in the single node case is less than the sum of the probabilities of more than one node case as the events are dependent and the sum of the values of $C_D(\cdot) < 0$ are also minimum (since min-cut). This implies

$$\sum_{\omega \in \Omega} \left( \sum_{v_i \in \omega} p_D(v_i)C_D(v_i) \right) < \sum_{\omega \in \Omega} \left( \sum_{v_i \in \omega} p_D(v_i)C_D(v_i) \right).$$

From (15) and by conditions 1) and 2), we get $U_D(p_\mathcal{D}', p_\mathcal{P}) < U_D(p_\mathcal{D}, p_\mathcal{P})$. Therefore, $p_\mathcal{D}'$ is not a best response for the defender and no best response of the defender has more than one node chosen as trapping node, if $p(\omega)$’s and $f(\omega)$’s are equal for all $\omega \in \Omega$.

The following result proves that if Lemma V.10 holds, then for a given adversary strategy, the best response of the defender is to select the min-cut nodes of $\mathcal{F}$ as trapping nodes.

**Lemma V.11:** Let $\Omega_\mathcal{D}$ be the set of attack paths in $\mathcal{F}$. Assume that the defender’s strategy satisfies conditions 1) and 2) in Lemma V.10 for a given adversary strategy $p_\mathcal{D}$. Then, the best response of the defender, $BR(p_\mathcal{D})$, is to select with nonzero probability a min-cut node set of the flow-network $\mathcal{F}$ as the trapping nodes.

**Proof:** Under conditions (1) and (2) in Lemma V.10, the best response of the defender is to select one node in every attack path as trapping node. Note that all attack paths in $\mathcal{F}$ pass through some node in the min-cut node set $\hat{S}$. By selecting the nodes in $\hat{S}$ as trapping nodes, all possible attack paths have some nonzero probability of getting detected. We prove the result using a contradiction argument. Suppose that the best response of the defender is not to select the nodes in $\hat{S}$ as trapping nodes. Then, there exists a subset of nodes $\hat{S} \subseteq V_\mathcal{D}$ such that $\sum_{v_i \in \hat{S}} C_D(v_i) < \sum_{v_i \in \hat{S}} C_D(v_i)$. Furthermore, all possible attack paths in $\mathcal{F}$ pass through some node in $\hat{S}$. Then, $\hat{S}$ is a (source-sink)-cut set and $\kappa(\hat{S}) := \{ (v_i, v_j) : v_i, v_j \in \hat{S} \}$. Then, $\kappa(\hat{S})$ is a cut set and $c_f(\kappa(\hat{S})) < c_f(\kappa(\hat{S}'))$. This contradicts the fact that $\kappa(\hat{S}')$ is an optional solution to the (source-sink)-min-cut problem. Hence, under Lemma V.10, the best response of the defender is to select the nodes in $\hat{S}$ as trapping nodes.

Using Lemmas V.8, V.10, and V.11, we present the proof of Theorem V.7 as follows.

**Proof of Theorem V.7:** Consider a defender’s strategy that selects the min-cut nodes as the trapping nodes. Then, by Lemma V.8, the best response of the adversary is to select transitions in such a way that all attack paths with nonzero probability pass through exactly one node in the min-cut. Furthermore, let $\Omega_\mathcal{P}$ be the set of attack paths in $\mathcal{F}$ and $\Omega$ be the set of paths in $S$ induced by $\Omega_\mathcal{D}$. Lemma V.11 concludes that the best response of the defender is to select the min-cut nodes of $\mathcal{F}$ as trapping nodes, provided the probability of detecting the adversary and the probability of trapping a benign flow are equal for all $\omega \in \Omega$. This implies that, if NE strategy pair satisfy the conditions that $p(\omega)$’s and $f(\omega)$’s are equal for all $\omega \in \Omega$, then the defender’s
strategy at NE is to select the min-cut nodes as the trapping nodes and the adversary’s strategy is to choose an attack path such that it passes through exactly one node in the min-cut node set. By Proposition V.5, there exists an NE for G. Consequently, if $p(\omega)$’s and $f(\omega)$’s are equal at NE, the proof follows.

Let $(p_A, p_D)$ be an NE of G. Consider a unilateral deviation in the strategy of the adversary. Let $\pi(\omega)$’s for $\omega \in \Omega$ are modified due to change in transition probabilities of the adversary such that the updated probabilities of the attack paths are $\pi(\omega_i) + \epsilon_i$, for $i = 1, \ldots, |\Omega|$. Here, $\epsilon_i$’s can take positive values, negative values, or zero such that $\sum_{\omega \in \Omega} \epsilon_i = 0$. Consider two arbitrary paths, say $\omega_1 \in \Omega$ and $\omega_2 \in \Omega$, such that a unilateral change in the adversary strategy changes $\pi(\omega_1)$ and $\pi(\omega_2)$ and the probabilities of the other paths remain unchanged. Without loss of generality, assume that $\pi(\omega_1)$ increases by $\epsilon$, while $\pi(\omega_2)$ decreases by $\epsilon$ and all other $\pi(\omega)$’s remain the same. As $(p_A, p_D)$ is an NE, $(\pi(\omega_1) + \epsilon)(p(\omega_1)(\alpha_A - \beta_A) + \beta_A + f(\omega_1)\beta_A) + (\pi(\omega_2) - \epsilon)(p(\omega_2)(\alpha_A - \beta_A) + \beta_A + f(\omega_2)\beta_A) \leq \pi(\omega_1)p(\omega_1)(\alpha_A - \beta_A) + \beta_A + f(\omega_1)\beta_A) + \pi(\omega_2)p(\omega_2)(\alpha_A - \beta_A) + \beta_A + f(\omega_2)\beta_A).$

Thus, 

$$ (p(\omega_1) - p(\omega_2))(\alpha_A - \beta_A) + (f(\omega_1) - f(\omega_2))\beta_A \leq 0. \tag{17} $$

Now consider another unilateral deviation of the adversary strategy such that $\pi(\omega_1)$ decreases by $\epsilon$, while $\pi(\omega_2)$ increases by $\epsilon$ and all other $\pi(\omega)$’s remain the same. This gives 

$$ (p(\omega_1) - p(\omega_2))(\alpha_A - \beta_A) + (f(\omega_1) - f(\omega_2))\beta_A \geq 0. \tag{18} $$

Equations (17) and (18) imply 

$$ (p(\omega_1) - p(\omega_2))(\alpha_A - \beta_A) + (f(\omega_1) - f(\omega_2))\beta_A = 0. \tag{19} $$

Note that $(\alpha_A - \beta_A) < 0$ and $\beta_A > 0$. Therefore, there are the following three possible cases where (19) holds:

1. $(p(\omega_1) - p(\omega_2)) > 0$ and $(f(\omega_1) - f(\omega_2)) > 0$;
2. $(p(\omega_1) - p(\omega_2)) < 0$ and $(f(\omega_1) - f(\omega_2)) < 0$;
3. $(p(\omega_1) - p(\omega_2)) = 0$ and $(f(\omega_1) - f(\omega_2)) = 0$.

Consider case (1). For a path $\omega$ in the state space, let $\omega(A)$ denote the set of nodes in $\omega$ corresponding to the adversarial flow and $\omega(B)$ denote the set of nodes in $\omega$ corresponding to the benign flows. Rewriting $p(\omega_1) - p(\omega_2) > 0$, we get 

$$ \left[ 1 - \prod_{v_i \in \omega(A)} (1 - p_D(v_i)) \right] \left[ 1 - \prod_{v_j \in \omega(B)} (1 - p_D(v_j)) \right] (1 - FN) > 0. \tag{20} $$

Similarly rewriting $(f(\omega_1) - f(\omega_2)) > 0$, we get 

$$ \left[ 1 - \prod_{v_i \in \omega_1(B)} (1 - p_D(v_i)) \right] \left[ 1 - \prod_{v_j \in \omega_1(B)} (1 - p_D(v_j)) \right] FP > 0. \tag{21} $$

As $0 < FP < 1$ and $0 < FN < 1$, (20) and (21) imply 

$$ \prod_{v_i \in \omega_2(A)} (1 - p_D(v_i)) - \prod_{v_j \in \omega_2(A)} (1 - p_D(v_j)) > 0 \tag{22} $$

The aforementioned equations imply

$$ \prod_{v_i \in \omega_2(A)} (1 - p_D(v_i)) > \prod_{v_j \in \omega_2(A)} (1 - p_D(v_j)) \tag{24} $$

and

$$ \prod_{v_i \in \omega_2(B)} (1 - p_D(v_i)) > \prod_{v_j \in \omega_2(B)} (1 - p_D(v_j)). \tag{25} $$

Note that at every state in $s \in S$ with $s = \{v_1, \ldots, v_k\}$, $0 \leq \sum_{v_k \in s} p_D(v_k) \leq 1$. Thus, (24) and (25) cannot be together satisfied. Thus, case (1) does not hold at an NE. Following the similar arguments, one can show that case (2) also does not hold at an NE. This implies $(p(\omega_1) - p(\omega_2)) = 0$ and $(f(\omega_1) - f(\omega_2)) = 0$.

Since $\omega_1$ and $\omega_2$ are arbitrary, one can show that for the general case 

$$ \sum_{i=1}^{[\Omega]} \epsilon_i p(\omega_i) = 0 \text{ and } \sum_{i=1}^{[\Omega]} \epsilon_i f(\omega_i) = 0. \tag{26} $$

The aforementioned equation should hold for all possible values of $\epsilon_i$’s satisfying $\sum_{i=1}^{[\Omega]} \epsilon_i = 0$. This gives $p(\omega_1) = p(\omega_2)$ and $f(\omega_1) = f(\omega_2)$ for all $i, j \in \{1, \ldots, [\Omega]\}$, at NE. This completes the proof.

Theorem V.7 concludes that the set of all solutions to the min-cut problem characterizes the set of NE of game G. Two key properties of an equilibrium strategy pair is that defender chooses the min-cut nodes as trapping nodes and indeed with equal probability, which is proved in Lemma V.12.

**Lemma V.12:** Consider the APT versus DIFT game G and let $(p_A, p_D)$ be an NE of G. Then, under $p_D$, all the min-cut nodes of the flow network have equal probability of selecting as a trapping node.

**Proof:** At NE, all the paths in the state space have equal probability of getting detected, i.e., $p(\omega)$’s are same for all $\omega \in \Omega$ (by Theorem V.7). For a path $\omega \in \Omega$ with $\omega(A)$ denoting the set of nodes corresponding to the adversarial flow, we know 

$$ p(\omega) = \left[ 1 - \prod_{v_i \in \omega(A)} (1 - p_D(v_i)) \right] (1 - FN) \text{ for all } \omega \in \Omega. \tag{27} $$

Also, there is exactly one node corresponding to an attack path that has nonzero probability of selecting as a trapping node (Theorem V.7). Hence, in set $\omega(A)$ exactly one node, say $v_i \in \omega(A)$, has nonzero value of $p_D$. Equation (27), hence, implies that at NE all the min-cut nodes of the flow network have equal probability of selecting as a trapping node.

As a consequence of Theorem V.7 and Lemma V.12, the following corollary holds.

**Corollary V.13:** Let $\hat{S}^*$ be a set of min-cut nodes of the flow-network. Then, at an NE of the game G, the defender’s strategy is to choose all the nodes in $\hat{S}^*$ as trapping nodes with equal probability and the adversary’s strategy is to choose transitions in such a way that each attack path passes through exactly one node in $\hat{S}^*$.

**Proof:** By Theorem V.7 and Lemma V.12, the defender’s strategy at NE is to select min-cut nodes as trapping nodes with
equal probability. We know adversary’s payoff

$$U_a(p_D, p_A) = \sum_{\omega \in \Omega} \pi(\omega) \left[ p(\omega)\alpha_A + (1 - p(\omega) + f(\omega))\beta_A \right]$$

and at NE $p(\omega)$’s and $f(\omega)$’s are equal for all $\omega \in \Omega$. Thus, the adversary’s optimal strategy is to select any path (or set of paths if mixed) that passes through exactly one min-cut node.

Using Theorem V.7 and Corollary V.13, we present an approach in the following to compute an equilibrium strategy pair of $G$. First, we solve the node version of the (source-sink)-min-cut problem on $F$. Let an optimal solution be $\kappa(S^*)$ and the corresponding vertex set be $S^* = \{\tilde{v}_1, \ldots, \tilde{v}_r\}$, where $\hat{S}^* := \{v_i : (v_i, v'_i) \in \kappa(S^*)\}$. By Theorem V.7, at NE, the defender only selects the nodes in $\hat{S}^*$ as trapping nodes, with equal probability, and the adversary chooses transitions such that each attack path passes through only one node in $\hat{S}^*$. Therefore, the attack paths chosen by the adversary is characterized by the nodes in $\hat{S}^*$. The set of paths in the state space $S$ can also be characterized by the nodes in $\hat{S}^*$. In other words, the set of nodes in the set $\Omega$ can be grouped such that each group corresponds to a set of paths in $\Omega$ in which attack path (not necessarily benign flows) passes through exactly one node in $\hat{S}^*$, i.e., at least one among the $W$ tagged flows passes through exactly one min-cut node. We denote the set of paths in $\Omega$ that correspond to node $v_i \in \hat{S}^*$ as $\Omega(v_i)$. By Corollary V.13, any distribution over paths in $\Omega(v_i)$, for $i \in \{1, \ldots, r\}$ is an optimal strategy for $P_A$.

**Remark V.14:** At NE, all attack paths in $F$ with nonzero probability pass through exactly one node in the min-cut node set $\hat{S}^* = \{\tilde{v}_1, \ldots, \tilde{v}_r\}$. Furthermore, the probabilities of selecting them as trapping nodes are also equal (since $p(\omega)$’s are equal). Thus, without loss of generality, one can characterize the action space of the adversary as the set of attack paths in $F$ that pass through $\hat{S}^*$, disjoint with respect to nodes in $\hat{S}^*$, and the adversary strategies over this set of paths.

In the following theorem, we derive a closed-form expression for the optimal defender strategy for the game $G$.

**Theorem V.15:** Consider the APT versus DIFT game $G$. At NE, the defender selects all the min-cut nodes as trapping nodes with probability $1 / \min\{W, r\}$, where $W$ and $r$, respectively, denote the number of tagged flows at a time in the system and the cardinality of the min-cut nodes.

**Proof:** Consider a nonterminal state $s \in S$, where $s = \{v_1, \ldots, v_w\}$. The action set of $P_D$ at $s$ is $A_D(s) = \{0\} \cup \{v_1, \ldots, v_w\} \subseteq V_G$. Any defender strategy must satisfy for all $s \in S$, $p_D(0) + \sum_{v_i=1}^w p_D(v_i) = 1$ if $v_i \neq \phi$. By Corollary V.13, at NE, $P_D$ only selects min-cut nodes as trapping nodes. Let $\hat{S}^* = \{\tilde{v}_1, \ldots, \tilde{v}_r\}$ be the solution obtained for the min-cut problem. Thus, the constraint on $p_D$ (that it should add up to 1 at all states) boils down to the following. For all states $s = \{v_1, \ldots, v_w\} \subseteq \{\tilde{v}_1, \ldots, \tilde{v}_r\}$

$$p_D(0) + \sum_{v_{i_j}, v_{i_j} \in \{\tilde{v}_1, \ldots, \tilde{v}_r\}} p_D(v_{i_j}) = 1. \quad (28)$$

Moreover, we know all min-cut nodes are chosen as trapping nodes with equal probability (see Corollary V.13), i.e., $p_D(\tilde{v}_1) = \ldots = p_D(\tilde{v}_r) := \theta$. For all states $s = \{v_{i_1}, \ldots, v_{i_w}\} \subseteq \{\tilde{v}_1, \ldots, \tilde{v}_r\}$

$$p_D(0) + \sum_{v_{i_j}, v_{i_j} \in \{\tilde{v}_1, \ldots, \tilde{v}_r\}} p_D(v_{i_j}) = 1. \quad (29)$$

For all states $s \in S$ with $s = \{v_{i_1}, \ldots, v_{i_w}\} \subseteq \{\tilde{v}_1, \ldots, \tilde{v}_r\}$

$$\sum_{j \in \{1, \ldots, W\}} 1 \leq \min\{W, r\}. \quad (30)$$

The aforementioned equation must hold for all states, and hence, the maximum value that $\theta$ can take is when $p_D(0) = 0$ and $\sum_{v_{i_j}, v_{i_j} \in \{\tilde{v}_1, \ldots, \tilde{v}_r\}} 1 = \min\{W, r\}$ at a state $s \in S$. Thus, $\theta = \frac{1}{\min\{W, r\}}$.

As the defender’s and adversary’s action sets are characterized by the min-cut nodes, the defender’s payoff is

$$U_D(p_D, p_A) = \sum_{i=1}^r \sum_{\omega \in \Omega(v_i)} \pi(\omega) \left[ p(\omega)\alpha_D + (1 - p(\omega) + f(\omega))\beta_D \right] + \sum_{i=1}^r \sum_{\omega \in \Omega(v_i)} \left( \sum_{v_{i_j} \in \omega} p_D(v_{i_j})C_D(v_{i_j}) \right) \quad (31)$$

The aforementioned equation (31) holds from Lemma V.12 by substituting $p(\omega) = [1 - \prod_{i \in \mathcal{V}(A)} (1 - p_D(\tilde{v}_i))] (1 - FN) = (1 - (1 - \theta)) (1 - FN) = \theta (1 - FN)$. Here, 0 $\leq \theta (1 - FN) \leq 1$ and 0 $\leq f(\omega) \leq 1$. Also, $(\alpha_D - \beta_D) >> \beta_D$ with $(\alpha_D - \beta_D) >> 1$ and $\beta_D < 1$. Also note that any distribution, $\pi(\omega)$, over paths in $\Omega(v_i)$, for $i \in \{1, \ldots, r\}$, is optimal for adversary (see Corollary V.13). Thus, (31) is maximum for the maximum possible $\theta$. Hence, at NE, the defender selects a set of min-cut nodes as trapping nodes with probability $1 / \min\{W, r\}$.

In this work, we provide a characterization for the equilibrium of the DIFT versus APT game using the min-cut analysis. Min-cut of a flow-network need not necessarily be unique. For an application whose IFG results in a flow network with multiple min-cut sets, equilibrium of the DIFT versus APT game will be nonunique. The game considers a strategic adversary and the defense strategy computed as equilibrium solution to the game provides a selection of the trapping nodes that maximizes the probability of detection while minimizing the cost of security analysis. During a real attack scenario, the adversary may choose an attack path that is different from the equilibrium of the game. We note that, by the definition of Nash equilibrium, such a strategy will not improve the payoff of the adversary. Moreover, for a defender that is unaware of the actions of the adversary, implementing a solution to the game as the trapping nodes maximizes the probability of detection.
VI. NUMERICAL STUDY

We validate our theoretical results using real-world attack data obtained using RAIN [1] for a data exfiltration attack. We use system log data collected during DARPA transparent computing (DARPA-TC) red team engagement that evaluated the RAIN log recording system. A brief description of the dataset used and the steps involved in the construction of the IFG for that attack is given as follows.

The goal of the attack is to exfiltrate sensitive information from the system. The attack uses stolen credentials to exfiltrate sensitive files from the targeted machine. The attack exfiltrates sensitive system files using scp. The resulting IFG of the data consists of 299 nodes and 6398 edges. Fig. 4 shows a portion of the IFG, obtained after performing a pruning technique, that captures all possible attack paths in the IFG. The details of the type of the nodes in the IFG in Fig. 4 is presented in Table I. Entry points of the attack are identified as /bin/bash and /usr/sbin/sshd (nodes 1 and 5 in Fig. 4). Destination node of the attack is /etc/passwd (node 3 in Fig. 4).

First, we solve the min-cut problem on the flow-network constructed from the IFG of the data. Resulting min-cut nodes, /usr/bin/sudo and /run/consolekit/data, are indicated in red color in Fig. 4. The min-cut nodes are chosen as trapping nodes with probability \( \theta = \frac{1}{\min\{W, r\}} \). W and \( r \) denote the number of tagged flows at a time in the system and the cardinality of the obtained solution to the min-cut problem, respectively. Here, \( W = 3 \) and \( r = 2 \). Thus, \( \theta = 0.5 \).

Then, we simulate an attack in the IFG and perform security analysis to evaluate the performance of the DIFT model. We conduct two experiments that are detailed as follow. Each payoff value represented in Figs. 5 and 6 are obtained after averaging over 1000 trials. We have identified 13 distinct attack paths related to the adversary in the underlying IFG and uniformly picked one attack path in each trial. Each of these attack paths consist only one min-cut node (see Theorem V.7). Furthermore, we set \( \alpha_a = -1000, \beta_a = 1000, \alpha_b = 1000, \) and \( \beta_b = -1000 \). \( C_{\theta}(v_i) \) values are set to be proportional to the number of information flows passing through each node, \( v_i \) for \( i \in \{1, \ldots, N\} \), in the IFG throughout the logging period.

A. Case Study 1

In this experiment setup, we vary the probability of selecting min-cut nodes as trapping nodes. Recall that any probability greater than \( \theta \) is not a valid defender’s strategy (see Theorem V.15). Hence, for comparing the performance, we select probabilities \( \theta \leq \frac{1}{\min\{W, r\}} \). Note that at NE, \( \theta = 0.5 \). Fig. 5(a) and (b) plots the payoff values for both players averaged over 1000 trials for different values of FP and FN. Fig. 5(a) shows that defender performs better when following the NE strategy, i.e., \( \theta = 0.5 \), which validates the theoretical analysis.

B. Case Study 2

In this case study, we compare the performance of the DIFT by varying the locations of trapping nodes. In provenance-based analysis, various heuristics have been proposed for selecting the trapping nodes [27], [28]. In [27], a heuristic that assigns weights to the nodes of the IFG is proposed. In this case study, we compare the performance of the defense strategy obtained as solution to the DIFT versus APT game with two cases. For the first case, we assign nonzero weights to the cut nodes of the IFG, since cut nodes are the smallest set of nodes through which all the flows pass through, and assign zero weights to all noncut nodes. For the second case, we select the noncut nodes as the trapping nodes, since all information flows will pass through the noncut nodes, by assigning nonzero weights to all noncut nodes and assign zero weights to all cut nodes. Fig. 6(a) and (b) shows that the expected payoff of the defender takes higher values when the min-cut nodes are chosen as the trapping nodes. Also note that choosing a cut is better than selecting set of nodes that are not a cut.

C. Case Study 3

In this case study, we compare the performance of the game theoretic approach with a heuristic rule-based method. For comparing the two approaches, we use the knowledge of the actual attack path as the ground truth. The attack path for the data
Parameters chosen are: $\alpha_A = -1000$, $\beta_A = 1000$, $\alpha_D = 1000$, and $\beta_A = -1000$. The resource cost for the nodes are chosen such that $C_D(v_i)$ is proportional to the total number of information flows at node $v_i$ for the whole logging period, where $i \in \{1, \ldots, N\}$. (a) and (b) shows the payoffs of the defender and adversary, respectively, for different values of FPs and FNs when min-cut nodes are chosen as trapping nodes. The probability of selecting a min-cut node as trapping node is varied from $\theta = 1/\min\{W/r\} = 1/\min\{3, 2\} = 0.5$, $\theta = 0.4$, and $\theta = 0.2$ for all the experiments. Also, each case is averaged over 1000 runs.

Parameters chosen are: $\alpha_A = -1000$, $\beta_A = 1000$, $\alpha_D = 1000$, and $\beta_A = -1000$. The resource cost for the nodes are chosen such that $C_D(v_i)$ is proportional to the total number of information flows at node $v_i$ for the whole logging period, where $i \in \{1, \ldots, N\}$. (a) and (b) shows the payoffs of the defender and adversary, respectively, for different values of FPs and FNs when 1) min-cut nodes are chosen as trapping nodes, 2) cut nodes (not min-cut), and 3) nodes that are not a cut. Also, each case is averaged over 1000 runs.

### Table II

| Node $v_i$ | 2 | 16 | 17 |
|------------|---|----|----|
| $q(v_i)$   | 0.005705 | 0.167162 | 0.016177 |

Exfiltration attack is $1 \rightarrow 15 \rightarrow 5 \rightarrow 16 \rightarrow 17 \rightarrow 4 \rightarrow 2 \rightarrow 3$ in the IFG given in Fig. 4. The heuristic rules used for identifying the trapping nodes are presented in Table III. Using these rules, for the data exfiltration attack, we identified the set of nodes $\{2, 6, 12, 13, 16, 17\}$ as the trapping nodes. Solution to the DIFT versus APT game returns the min-cut node set $\{2, 11\}$ as the trapping nodes. In this case study, we set $W = 1$, and consequently, $\theta = 1$. The parameters of the game are set as $\alpha_D = 1$, $\beta_D = -1$, $FN = 0.2$, and $FP = 0$.

Resource cost for performing security analysis at a node $v_i$ is given by

$$C_D(v_i) = cq(v_i)$$ (32)

where $c$ is the cost parameter for performing security analysis, and $q(v_i)$ is the fraction of information flows passing through node $v_i$. The resource cost for conducting security analysis is more for a node with higher fraction of flows, i.e., a busy node. Fig. 7 presents a plot showing the variation of the DIFT’s payoff with respect to $c$. For the data exfiltration attack considered and
TABLE III
HEURISTIC RULES USED FOR IDENTIFYING TRAPPING NODES IN CASE STUDY 3

| System call    | Performed on file | Description                                                                 |
|----------------|-------------------|------------------------------------------------------------------------------|
| read           | /dev/mem          | Potentially anomalous actions on sensitive files/folder                      |
| read           | /etc/passwd       |                                                                              |
| read           | /etc/shadow       |                                                                              |
| read           | *.ssh/d_rsa       | Potentially anomalous actions on sensitive SSH files                         |
| read           | *.default/key3.db |                                                                              |
| read           | *.default/logins.json |                                                            |
| read           | *.default/signons.sqlite |                                                          |
| write          | /dev/mem          |                                                                              |
| write          | /etc/*            |                                                                              |
| write          | /usr/local/*      |                                                                              |
| write          | /usr/local/bin/*  |                                                                              |
| write          | /var/spool/cron/crontabs/root |                                                          |
| write          | /bin/*            | Potentially anomalous actions on sensitive SSH files                         |
| write          | /boot/*           |                                                                              |
| write          | /var/*            |                                                                              |
| write          | /usr/bin/*        |                                                                              |
| write          | /dev/*            |                                                                              |
| write          | /etc/security/*   |                                                                              |
| write          | /usr/spool/*      |                                                                              |
| write          | /usr/etc/*        |                                                                              |
| write          | /usr/kvns/*       |                                                                              |
| write          | /usr/*            |                                                                              |
| write          | /usr/lib/*        |                                                                              |
| write          | *.default/key3.db |                                                                              |
| write          | *.default/logins.json |                                                          |
| write          | *.default/signons.sqlite |                                                          |
| write          | /etc/hosts        | Important DNS information                                                   |
| write          | *.default/extensions | Stores extensions for Firefox                                     |
| open           | *libpam*          | Used for authentication                                                     |
| exec           | bin/ip            | Used for gathering information about network                                |
| exec           | /bin/ifconfig     |                                                                              |
| exec           | /bin/netstat      |                                                                              |
| exec           | /bin/chmod        | Changes permissions of files/folders                                        |
| exec           | /bin/chown        | Changes ownership of files/folders                                          |

for the selection of the trapping nodes \{2, 6, 12, 13, 16, 17\}, heuristic rule-based method will conduct security analysis on the flow at nodes 2, 16, and 17. On the other hand, the game theoretic modeling will conduct security analysis on the flow at node 2. Table II presents the fraction of the information flows passing through nodes 2, 16, and 17. Fig. 7 shows that the heuristic rule-based method performs better for lower values of \(c\). This is expected since when cost for conducting security analysis is negligible, it is better to analyze the flow at multiple locations, as it increases the probability of detection. As the cost parameter increases, the game theoretic solution outperforms the heuristic. The performance of the heuristic decreases drastically for higher values of \(c\) as heuristic conducts security analysis at node 16, which is a busier node, and hence, incurs high resource cost. The game, on the other hand, considers the resource cost while computing the min-cut, and hence, chooses node 2 for conducting the security analysis. This validates that the game theoretic approach provides a tradeoff between the effectiveness of detection and the cost of detection.

VII. CONCLUSION

This article provided a game-theoretic model for the detection of APTs using DIFT. The interaction of APT and DIFT is modeled as a stochastic game in which the defender dynamically chooses locations (trapping nodes) to perform security analysis and the adversary chooses transitions along attack paths in order to maximize the probability of reaching the desired target. Before performing security analysis, the defender is unable to distinguish whether a tagged information flow at a node in the IFG is malicious or not, and this results in the information asymmetry of the players. Furthermore, the defender is not accurate in its detection process, which results in the generation of FPs and FNs. We modeled the strategic interactions between DIFT and APT as a nonzero-sum, turn-based stochastic game. The game captures the information asymmetry among the players along with FPs and FNs of DIFT. Then, we provided an approach to compute NE of the game, using a minimum capacity cut-set formulation. Finally, we implemented our algorithm on real-world data for a data exfiltration attack obtained using the RAIN system. Extending the game model to address a multiattacker case with multiple malicious flows is a part of future work.

ACKNOWLEDGMENT

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the NSF or its federal agency and industry partners.

REFERENCES

[1] Y. Ji et al., “RAIN: Refinable attack investigation with on-demand inter-process information flow tracking,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., 2017, pp. 377–390.
[2] J. Newsome and D. Song, “Dynamic taint analysis: Automatic detection, analysis, and signature generation of exploit attacks on commodity software,” in Proc. Netw. Distrib. Syst. Secur. Symp., 2005, pp. 3–4.
[3] M. Dalton, H. Kannan, and C. Kozyrakis, “Real-world buffer overflow protection for userspace and kernelspace,” in Proc. USENIX Secur. Symp., 2008, pp. 395–410.
[4] M. Dalton, C. Kozyrakis, and N. Zeldovich, “Nemesis: Preventing authentication & access control vulnerabilities in web applications,” in Proc. USENIX Secur. Symp., 2009, pp. 267–282.
[5] L. S. Shapley, “Stochastic games,” Proc. Nat. Acad. Sci., vol. 39, no. 10, pp. 1095–1100, 1953.
[6] K.-W. Lye and J. M. Wing, “Game strategies in network security,” Int. J. Inf. Secur., vol. 4, no. 1/2, pp. 71–86, 2005.
[7] R. Amir, “Stochastic games in economics and related fields: An overview,” in Proc. Stochastic Games Appl., 2003, pp. 455–470.
[8] Q. Zhu and T. Başar, “Robust and resilient control design for cyber-physical systems with an application to power systems,” in Proc. IEEE Decis. Control Eur. Control Conf., 2011, pp. 4066–4071.
[9] A. Jäskiewicz and A. S. Nowak, “Non-zero-sum stochastic games,” in Handbook of Dynamic Game Theory. Berlin, Germany: Springer, 2016, pp. 1–64.
[10] T. Alpcan and T. Başar, “An intrusion detection game with limited observations,” in Proc. Int. Symp. Dyn. Games Appl., vol. 26, 2006.
[11] Q. Zhu, H. Tembine, and T. Başar, “Network security configurations: A non-zero-sum stochastic game approach,” inProc. Amer. Control Conf., 2010, pp. 1059–1064.
[12] K. C. Nguyen, T. Alpcan, and T. Başar, “Stochastic games for security in networks with interdependent nodes,” in Proc. Int. Conf. Game Theory Netw., 2009, pp. 697–703.
Shana Moothedath (Member, IEEE) received the B.Tech. and M.Tech. degrees in electrical and electronics engineering from Kerala University, Thiruvananthapuram, India, in 2011 and 2014, respectively, and Ph.D. degree in electrical engineering from the Indian Institute of Technology Bombay, Mumbai, India, in 2018.

Her research interests include network security analysis, structural analysis of large-scale control systems, and applications of systems theory to complex networks.

Dinuka Sahabandu received the B.S. and M.S. degrees in electrical engineering from Washington State University, Pullman, WA, USA, in 2013 and 2016, respectively, and the Ph.D. degree in electrical engineering from the University of Washington, Seattle, WA, USA, in 2023.

He is currently a Postdoctoral Scholar with the Department of Electrical and Computer Engineering, University of Washington. His research interests include game theory for network security, security and privacy of machine learning algorithms, and the control of multiagent systems.

Joey Allen received the B.S. and M.S. degrees in computer engineering from the University of Tennessee, Knoxville, TN, USA, in 2014 and 2016, respectively, and the Ph.D. degree in computer science from the Georgia Institute of Technology, Atlanta, GA, USA, in 2023.

He is currently a Senior Staff Researcher with Palo Alto Networks, Santa Clara, CA, USA. His research interests include forensic analysis, mobile security, and information tracking.

Andrew Clark (Member, IEEE) received the B.S. degree in electrical engineering and the M.S. degree in mathematics from the University of Michigan - Ann Arbor, Ann Arbor, MI, USA, in 2007 and 2008, respectively, and the Ph.D. degree in electrical engineering from the Network Security Lab (NSL), Department of Electrical and Computer Engineering, University of Washington - Seattle, Seattle, WA, USA, in 2014.

He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Worcester Polytechnic Institute, Worcester, MA, USA. His research interests include control and security of complex networks, submodular optimization, control-theoretic modeling of network security threats, and deception-based network defense mechanisms.

Dr. Clark was the recipient of the IEEE/IFIP William C. Carter award in 2010, the WiOpt Best Paper award in 2012, the WiOpt Student Best Paper award in 2014, and an IEEE GameSec Outstanding Paper award in 2016, and was a finalist for the IEEE CDC 2012 Best Student-Paper Award and IEEE/ACM ICCPS Best Paper in 2016 and 2018. He was also the recipient of the University of Washington Center for Information Assurance and Cybersecurity (CIAC) Distinguished Research Award in 2012 and Distinguished Dissertation Award in 2014, and an NSF CRII award in 2016. He holds a patent in privacy-preserving constant-time identification of RFID.

Linda Bushnell (Fellow, IEEE) received the B.S. and M.S. degrees in electrical engineering from the University of California - Berkeley, Berkeley, CA, USA, in 1989 and 1994, respectively, and the M.B.A degree in technology management from the University of Washington Foster School of Business, Seattle, WA, USA, in 2010.

She is currently a Research Professor with the Department of Electrical and Computer Engineering, University of Washington - Seattle, Seattle. Her research interests include networked control systems and secure control.

Dr. Bushnell is a Fellow of the IEEE for contributions to networked control systems. She is a Fellow of IFAC for contributions to the analysis and design of networked control systems. She was the recipient of the US Army Superior Civilian Service Award, NSF ADVANCE Fellowship, and IEEE Control Systems Society Distinguished Member Award. She has also been a member of the IEEE Control Systems Society since 1990. She is currently the Treasurer of the American Automatic Control Council, Member of the Technical Board for the International Federation on Automatic Control, an Associate Editor for Automatica, and an Associate Editor for IEEE TRANSACTIONS ON CONTROL OF NETWORK SYSTEMS.
Wenke Lee (Fellow, IEEE) received the Ph.D. degree in computer science from Columbia University, New York, NY, USA, in 1999.

He is currently a Professor of computer science with the College of Computing, Georgia Institute of Technology, Atlanta, GA, USA. He also serves as the Director of the Georgia Tech Information Security Center (GTISC). He works in systems and network security. His current research projects are in the areas of botnet detection and attribution, malware analysis, virtual machine monitoring, mobile phone security, and detection and mitigation of information manipulation on the Internet, with funding from National Science Foundation (NSF), Department of Homeland Security (DHS), Department of Defense (DoD), and the industry. He co-founded Damballa, Inc., a spin-off from his lab that focuses on botnet detection and mitigation, in 2006. He has authored and coauthored more than 100 articles with more than 40 of them cited more than 100 times.

Radha Poovendran (Fellow, IEEE) received the B.S. degree in electrical engineering from the Indian Institute of Technology-Bombay, Mumbai, India, 1988, the M.S. degree in electrical and computer engineering from the University of Michigan - Ann Arbor, Ann Arbor, MI, USA, in 1992, and the Ph.D. degree in electrical and computer engineering from the University of Maryland - College Park, College Park, MD, USA, in 1999.

He is a Professor with the Department of Electrical and Computer Engineering, University of Washington (UW) - Seattle, Seattle, WA, USA. He served as the Chair of the Electrical and Computer Engineering Department, UW for five years starting January 2015. He is the Director of the Network Security Lab (NSL), UW. He is the Associate Director of Research with the UW Center for Excellence in Information Assurance Research and Education. His research interests include the areas of wireless and sensor network security, control and security of cyber-physical systems, adversarial modeling, smart connected communities, control security, games security, and information theoretic security in the context of wireless mobile networks.

Dr. Poovendran is a Fellow of the IEEE for his contributions to security in cyber-physical systems. He was the recipient of the Rising Star Award in 1999, National Science Foundation CAREER in 2001, Army Research Office Young Investigator Program (ARO YIP) in 2002, Office of Naval Research Young Investigator Program (ONR YIP) in 2004, and Presidential Early Career Awards for Scientists and Engineers (PECASE) in 2005 for his research contributions to multiuser wireless security. He was also the recipient of the Outstanding Teaching Award and Outstanding Research Advisor Award from University of Washington Electrical Engineering (UW EE) (2002), Graduate Mentor Award from Office of the Chancellor at University of California - San Diego (2006), and the University of Maryland ECE Distinguished Alumni Award (2016). He was a coauthor of award-winning papers including IEEE/IFIP William C. Carter Award Paper in 2010 and WiOpt Best Paper Award in 2012.