Game Theory-Based Approach for Defense against APTs

Juan E. Rubio, Cristina Alcaraz, Javier Lopez
Department of Computer Science, University of Malaga,
Campus de Teatinos s/n, 29071,Malaga, Spain
{rubio,alcaraz,jlm}@lcc.uma.es

March 6, 2022

Abstract

The sophistication of Advanced Persistent Threats (APTs) targeting industrial ecosystems has increased dramatically in recent years. This makes mandatory to develop advanced security services beyond traditional solutions, being Opinion Dynamics one of them. This novel approach proposes a multi-agent collaborative framework that permits to trace an APT throughout its entire life-cycle, as formerly analyzed. In this paper, we introduce TI&TO, a two-player game between an attacker and defender that represents a realistic scenario where both compete for the control of the resources within a modern industrial architecture. By validating this technique using game theory, we demonstrate that Opinion Dynamics consists in an effective first measure to deter and minimize the impact of an APT against the infrastructure in most cases. To achieve this, both attacker and defense models are formalized and an equitable score system is applied, to latter run several simulation test cases with different strategies and network configurations.

Keywords: Opinion Dynamics, Advanced Persistent Threat, Detection, Response, Defense, Game theory.

1 Introduction

There is an evident growth in the number of cyber-security attacks that worldwide companies have to face, which generates a huge economic loss due to the investment performed in terms of cyber-security [2]. This situation becomes more critical when it comes to critical infrastructures (i.e., nuclear plants, electricity grids, transport and manufacturing systems), whose industrial control systems must be kept working under all conditions. Here, we are dealing with SCADA (Supervisory Control and Data Acquisition) systems that have been working in isolation from external networks for decades; nowadays, in turn, they are increasingly integrating novel technologies such as Internet of Things

(IoT) or Cloud Computing to outsource diverse services while cutting costs. As a consequence, a greater effort is needed to keep up with such advancement, as to cope with the newest attack vectors and exploitable vulnerabilities that these systems may pose.

One of the most critical issue in recent years is the Advanced Persistent Threats (APTs), which are sophisticated attacks that are especially tailored to a target infrastructure, perpetrated by a well-resource organization. They are characterized for leveraging zero-day vulnerabilities and employ stealthy techniques that make the threat undetectable for a long period of time within the victim network. Stuxnet was the first reported threat of this nature [6], but many others were detected afterwards, usually months after the attack had been completely executed [7]. On the cyber-security side, just some mechanisms have been proposed to address this issue from a holistic perspective, beyond traditional mechanisms (e.g., firewalls, Intrusion Prevention Systems (IPS), Intrusion Detection Systems (IDS), antivirus) that only represent a punctual protection against APTs in their first stages [21].

Among the novel mechanisms, Opinion Dynamics [15] consists in a multi-agent collaborative system that enables the traceability of the attack throughout its entire life-cycle, by means of a distributed anomaly correlation. In this paper, we propose a theoretical but realistic scenario to prove the effectiveness of that approach under different types of attack model, using concepts supported by the structural controllability field [8] and game theory [14]. For that goal, we develop TI&TO, a two-player game where attacker and defender compete for the control of the resources within a modern industrial architecture. Both players have their own movements and associated scores, according to the behavior of an APT and a detection system based on Opinion Dynamics, respectively. This game is ultimately run in different simulations that aim to show the algorithm capabilities, while also suggesting the optimal configuration of the technique in conjunction with other defense solutions. Therefore, we can summarize our contributions as:

- Formal definition of the TI&TO game, specifying the game board, each player's goal and the score rules.
- Design of an attacker model in form of a set of stages that flexibly represents the phases of an APT, to represent the movements of the attacker, which are subject to a determined score.
- Design of a defender model based on the use of Opinion Dynamics and response techniques (i.e., local detection, redundant links, honeypots) to reduce the impact of the APT within the network, which also implies an associated score in the game.
- Experiments carried out to validate the algorithm and recommend the configuration of the defender that returns the best result.

The remainder of this paper is organized as follows: Section 2 introduces the concept of Opinion Dynamics and highlights other proposals that apply game theory for the detection of cyber-attacks. In Section 3 the game is defined, including the rules as well as the attack and defense models. Then, several simulations are carried out and a discussion is offered in Section 4. Lastly, the conclusions and future work are presented in Section 5.

2 Preliminaries

In this section, the main concepts that are needed to understand the basics of TI&TO are introduced from a theoretical perspective. Firstly, we explain the aspects of the Opinion Dynamics detection and its benefits, to later set the background with respect to game theory.

2.1 Opinion Dynamics

The Opinion Dynamics approach proposes to aggregate the coverage of multiple detection systems that are strategically deployed over an infrastructure, under a common distributed framework that permanently correlates and learns from all the malware patterns and anomalies detected. This way, various detection solutions can be combined at all levels to anticipate the new technology scenarios of Industry 4.0 in terms of security, by easing the traceability of attacks and the application of effective response procedures. Under a theoretical perspective, it was introduced in [15], and then its authors demonstrated the effectiveness of the approach with an enhanced attack model [17] and an improved correlation of events [16]. Concerning a practical applicability, its authors demonstrated its capabilities to detect and monitor attacks in a real industrial testbed [18], and also showed its contributions to the Smart Grid scenario [9], where it can help to prevent against intrusions and blackouts.

As its name suggests, this correlation algorithm conceptually models the opinion fluctuation in a society. It is one of the most popular consensus models to obtain the accurate polarization of opinions in certain population [4]. In our case, if we assume a set of agents (i.e., that monitor each device of the network) which are connected according to a graph $G(V, E)$ (where V is the set of agents and E the intermediate communication links between resources), each one holds a certain opinion (in our case, about the level of anomaly detected in its surroundings) and influences those of the agents who are closer in their posture. Eventually, once this correlation of opinion has been performed among all the individual agents, this 'society' (i.e., the network) is fragmented into different clusters of opinions that correspondingly identify the areas of the network that experience the same degree of anomaly (potentially caused by an attack focused on that zone).

The formalization of this correlation is the following: Opinion Dynamics is an iterative algorithm which assumes that there is a 1:1 relationship between devices in the control network (modelled with graph $G(V, E)$) and individual agents that are permanently monitoring their security state, so that we have a set of agents $A = \{a_1, a_2, ...a_{|V|}\}$. The opinion of an agent a_i at iteration t

is represented by $x_i(t)$. Initially, prior to execute the Opinion Dynamics correlation, $x_i(0)$ contains the level of anomaly sensed by agent i, which is a float number that ranges from 0 to 1 (being 1 the highest anomaly). On the other hand, the influence between agents (to determine the new opinions in next iterations) is represented by a weight given by each agent a_i to the opinion of each neighbour a_j in A (i.e., there exists an edge (v_i, v_j) in E), which is denoted by w_{ij}. For each agent a_i in A, we have that $\sum_{k=1}^{|V|} w_{ik} = 1$. This way, every agent i also takes its own opinion into account and weighs the influence of surrounding agents depending on the closeness of their opinions, as explained in [17]. Finally, the new opinion of the agent i in the iteration $t+1$ is generated according to the following expression:

$$x_i(t+1) = \sum_{j=1}^{n} w_{ij} x_j(t)$$

Therefore, the correlation of opinion for a given agent is performed as a weighted sum of the other opinions. If this algorithm is executed with enough number of iterations (something trivial due to the lower complexity of calculations), the resulting opinions of all agents can be grouped into clusters with the same anomaly value. Consequently, after the execution of Opinion Dynamics, the more affected areas after an attack will be those that expose a high opinion value.

Due to the flexible characterization of each agent opinion, this algorithm can be conceived as a framework, since multiple detection mechanisms can be orchestrated to analyze each host and its network activity to finally output a single anomaly value to represent $x_i(0)$ in the algorithm. As an example, the original authors suggest the use of anomaly detection mechanisms, vulnerability scanners or Security Information and Event Management systems, as well as ad-hoc machine learning techniques. These systems could be applied in a distributed way, and their outputs would be retrieved by a central correlator that features enough computational capabilities as to execute the Opinion Dynamics algorithm. This centralized entity is put into practice in [18], where other functionalities such as the evolution of anomalies over time and the persistence of resources are also studied.

Here, we leverage game theory to assess the utility of this mechanism when deploying response techniques that use the information provided by Opinion Dynamics in multiple scenarios. For such goal, we base our game on the detection approach and the attacker model presented in the aforementioned publications, to study the optimal configuration of different response procedures that ultimately aim to deter and eradicate the effect of an APT.

2.2 Game Theory: related work

In the context of industrial networks defense, researchers have been extensively exploring the applicability of game theory [14]. In these networks, it is common to cope with many levels of criticality, different network sizes, interconnectivity and access control policies. Therefore, decisions in terms of security frequently fluctuate, which is harder in Industry 4.0 scenarios, where many heterogeneous devices interact with each other and organizations exchange information using the Cloud, Fog Computing or Distributed Ledger Technologies. In this sense, game theory offers the capability of analyzing hundreds of scenarios, thereby enhancing the decision making. At the same time, it also allows to validate the effectiveness of a given technique (e.g., Opinion Dynamics in our case) if we analyze different strategies of use for all the scenarios examined.

Based on the information that each player has, there are different types of games: on the one hand, in a *perfect information* game both players are aware of the actions taken by their adversary at all times; on the other hand, a *complete information* game assumes that every player always knows the strategy and payoffs of the opponent. As explained further in Section 3, the approach presented in this paper (TI&TO) represents a two-player game with imperfect and incomplete information, since no player (i.e., attacker and defender) knows the location of the adversary within the network topology or his/her score. According to a second level of classification, this game can be considered as dynamic and stochastic, as both players take their actions based on the state of the network and being exposed to events that affect them in a probabilistic way.

There are multiple researches in the literature that fall under these classifications. Concerning complete perfect information games, Lye et al. [10] proposes a two-player game that simulates the security of a network composed by four nodes that can be in 18 potential states, on which both players can take up to 3 actions, that are observable at all times by the opponent. With respect to complete imperfect information games, Nguyen et al. [11] propose 'fictitious play (FP)', a game that considers the network security as a sequence of nonzero-sum games were both players cannot make perfect observations of the adversary's previous actions. On the other hand, Patcha et. al [13] propose a incomplete perfect information approach, for the detection of intrusions in mobile ad-hoc networks. Whereas the attacker's objective is to send a malicious message and compromise a target node, the defender tries to detect it using a host-based IDS. Another related work based on imperfect information is [20], where van Dijk et. al propose a simple game where two players compete for the stealthy control of a resource without knowing the actual identity of the owner until a player actually moves.

Many of these solutions have been successfully applied to the detection of threats. However, most of the models are based on either static games or dealing with perfect and complete information, aiming to find an optimal strategy when a steady state of the game is reached (being the Nash equilibrium the most famous one) [14]. In contrast, a real control system faces a dynamic interaction game with incomplete and imperfect information about the attacker, and the proposed models of this category do not specify a realistic scenario with an extensive attack model [20] [1]. This lays the base and inspiration for the design and implementation of our proposed scheme. With TI&TO, we aim to get insight about how to effectively implement and configure a defense strategy based on the use of Opinion Dynamics, under such stochastic conditions.

3 The game: attack and defense models

Once the problematic has been introduced and the Opinion Dynamics has been explained, this section presents TI&TO from a theoretical perspective, prior to execute simulations in Section 4. Firstly, we introduce the board, the rules and the overall objective for both players: attacker and defender. Then, each one is individually addressed and their attack model formalized.

3.1 The board: proposed network architecture

As defined in next subsection, TI&TO focuses on a game where both attacker and defender fight for the control of an infrastructure. The attacker tries to break into the network in a stealthy way by taking over as many nodes as to complete the predefined kill chain of a specific APT. With respect to the defender, he/she must recover those nodes until he/she completely eradicates the threat from the network. Thus, this network infrastructure plays the role of the game board, and must be designed realistically as to represent the topology of a modern industrial ecosystem.

For this reason, the network used in the game embodies cyber-physical resources of different nature, ranging from operational devices (OT) (e.g., sensors/actuators, Programmable Logic Controllers (PLCs), SCADA systems, etc.) to Information Technology (IT) devices from the managerial point of view (e.g., customer-end systems). Following the Opinion Dynamics solution [17], the board will be an infrastructure composed by two sections with the same number of nodes: OT and IT, connected via firewalls to secure the traffic. Let the network be represented with graph $G(V, E)$, so that V refers to the nodes connected with each other based on links contained in the E set. Thus, OT and IT sections are represented with $G(V_{OT}, E_{OT})$ and $G(V_{IT}, E_{IT})$, respectively (having $V = V_{IT} \cup V_{OT}$ and $E = E_{IT} \cup E_{OT}$). Both sections are randomly generated following a different network distribution, which enables us to simulate different infrastructure setups. On the one hand, the IT section follows a small-world network distribution, that models the traditional topology of TCP/IP networks [22]. In turn, $G(V_{OT}, E_{OT})$ is based on a power-law distribution of type $y \propto x^{-\alpha}$, that is commonly used for the modelling of industrial control systems [12].

Once generated, both sections are connected by means of a set of intermediate firewalls V_{FW}, so that $V = V_{IT} \cup V_{OT} \cup V_{FW}$, in the following way: as for the IT section, we want devices to be able to access the OT section, since they are computationally capable nodes that commonly control the production chain from the corporate network. This means that all nodes in V_{IT} are connected to V_{FW}. However, on the OT side, only SCADA systems and other high-level servers can access external networks, whereas the majority of them are sensors, PLCs and devices with a restricted functionality. Consequently, the connected

$V_{IT} - DS_{IT} - PDS_{IT}$	ψ_1
$V_{OT} - DS_{OT} - PDS_{OT}$	ψ_2
DS_{IT}	ψ_3
DS_{OT}	ψ_4
PDS_{IT}	ψ_5
$PDS_{OT} \cup FW$	ψ_6

Table 1: Map of V to Ψ

nodes will be those that have a maximum connectivity (i.e., dominance in graph theory) within the power-law distribution network of the OT section, given the concepts of structural controllability stated in [3] and [5]. According to these, the Dominating Set (DS) of a graph conforms the subset of nodes ($\mathbf{D_N}$ henceforth, also called 'driver nodes') for which every node not in $\mathbf{D_N}$ is adjacent to at least one member of $\mathbf{D_N}$. On the other hand, if we further restrict this condition, the Power Dominating Set (PDS) of a graph is defined as the subset of nodes for which every edge in E is adjacent to at least one node of the PDS. Therefore, for our concerned network infrastructure, this subset of nodes of the OT section will be connected to the firewalls that also connect to the IT nodes. In our simulations, we consider that the 5% of the total number of nodes in V are firewalls, to restrict the traffic between both sections in a realistic way.

In order to characterize the types of nodes within the architecture and enrich the network model, it is also necessary to define some related concepts that will be useful to understand the game dynamics:

Criticality of nodes. We define the criticality of a resource as the risk subject to that type of device within the organization, and determines the impact of a given threat if the attack is perpetrated at that point. For example, the criticality of a sensor is negligible compared to that of the SCADA system, which implies dramatic consequences on the infrastructure in the event it is disrupted. Likewise, resources in the OT section are also deemed as more critical than the IT ones to ensure the continuity of the production chain. This will be also used by the defender to assess which nodes should be healed in order to minimize the impact of an APT. We formally define this concept taking into account the graph $G(V, E)$ introduced before. Firstly, let $CRIT : V \mapsto \mathbb{R}(0, 1)$ be a function that assigns a criticality degree to all nodes of the network. In order to distinguish which devices present a higher hierarchy within the topology, we leverage the concept of DS and PDS introduced in Section 3.1. At the same time, since the OT section is considered as especially critical, its devices will have to be associated with a higher value. As a result, we define Ψ as an *ordered set of criticality values of size* d, where $\Psi = \psi_1, ..., \psi_d$ and $\psi_i = [0, 1]$, such that $\forall \psi_i, \psi_i < \psi_{i+1}$.

Once Ψ is defined, we can create a model that maps every element of the network (i.e., its nodes) to the elements of Ψ. Such model, where $d = 6$ and $\Psi = \psi_1, \psi_2, \psi_3, \psi_4, \psi_5, \psi_6$ to consider all elements of both network sections (i.e., the OT and IT section, including its nodes and the DS and PDS subsets), is described in Table 1.

Vulnerability of nodes. Besides the criticality, the concept of vulnerability involves the ease of a node to be compromised by the attacker. In this case, we will assume that this value is opposed to the criticality, in the sense that field devices will be commonly equipped with lower security protection measures, whereas high-level systems that control the industrial process will embody advanced security services. Correspondingly, we can define $VULN : V \mapsto \mathbb{R}(0,1)$ as the function that assigns a vulnerability degree to all nodes of the network. In the same way as criticality, Υ is an ordered set that represents the vulnerability of each node type, where $\Upsilon = \upsilon_1, ..., \upsilon_d$ and $\upsilon_i = 1 - \psi_i$. The particular instantiation of these values for the simulations is carried out when the network represented by $G(V, E)$ is created. This is further addressed in Appendix A.

Redundancy of links. In order for the OT subnetwork to be resilient against Denial of Service attacks located on their links, and due to the criticality of its resources, we also consider that this section presents redundancy on its edges. This is a solution that was also proposed in [15] as a response technique to enable the reachability of messages across the network. In our case, with the use of auxiliary edges in E (referred to as E_R, so that $E_R \subset E$), we ensure that the detection algorithm exchanges the opinion among agents even when some links are down as consequence of an APT. This may occur in the game when the attacker attempts the defender to lose track of the anomalies in the affected nodes. This way, all nodes in V_{OT} count on an additional channel that interconnects them with another node, based on the strategy explained in [15]. It is worthy to note that these redundant edges are just logical connections that only serve to transfer the anomaly values between agents.

Altogether, Figure 1 conceptually shows an example of network topology based on these assumptions together with the integration of the Opinion Dynamics correlator. In the diagram, the redundant edges in the OT section are represented with dashed lines.

3.2 Rules and scoring system

We now describe the game dynamics for both players and how each of their movements is measured in quantitative terms. Since the final objective of this research is to assess the effectiveness of the Opinion Dynamics, we aim to analyze the best behavior of the defender for a realistic attack model. Therefore, it becomes necessary to utilize a formal representation of the results while following a fair methodology for both players, which have equivalent costs and rewards assigned to their movements in the game.

We start by defining TI&TO in an informal way. As introduced before, both compete for the control of the game board. The base of the scoring system works as follows: *whereas the attacker earns points as it spreads the threat across the infrastructure, the defender increases the score when those infected nodes are*

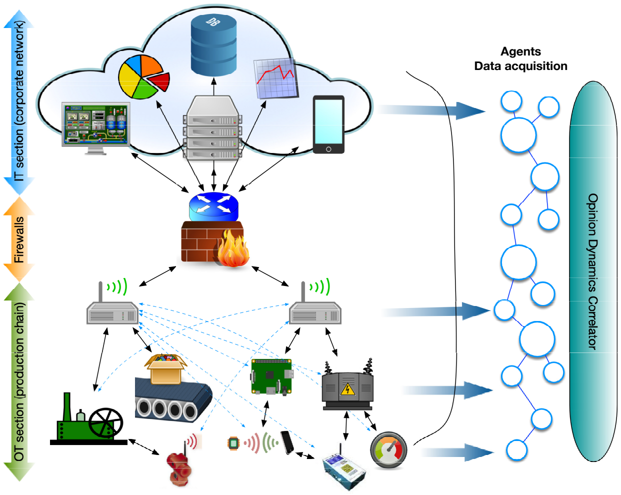

IT section (corporate network)
Firewalls
OT section (production chain)
DB
Agents
Data acquisition
Opinion Dynamics Correlator

Figure 1: Example of network topology used in TI&TO

recovered. However, this is just the number of points scored, which serves as a reference of the throughput achieved by each player. There is a termination condition that regulates who wins a given game: *as for the attacker, the game is over when he/she manages to successfully complete all the phases of the APT kill chain*. Concerning the defender, *the victory is achieved when all nodes infected by the adversary return to their originally uncompromised state*. In the following, we give a formal definition of all the elements involved in TI&TO and the notation used along this manuscript:

Players. There are two players: the attacker and the defender. For simplicity, they are denoted by A and D, respectively.

Time. In our approach, time is split into discrete ticks for the interest of the analysis. The game begins at time $t = 0$ and continues indefinitely as $t \to \infty$. At a given t, A and then D has a turn to play. They act sequentially adopting a Stackelberg game [19], where the attacker is the leader and the defender acts depending on the resulting state of the board.

Movement. It is performed by A or D and changes the board at time t according to their respective attack and defense models. In brief, both players take actions to either take over healthy nodes of the network (in the case of the attacker) or heal a compromised node (by the defender). Therefore, every movement can alter the state of a node. It is denoted by $M^p(t)$

Node State. It is a time-dependent variable $N = N(t)$ that determines whether a node in V is compromised (i.e., the attacker has reached it) or remains safe from the APT. For a given node i (belonging to the IT or OT section), $N_i(t)$ is equals to one if it is compromised at time t, and zero otherwise. We assume that $\forall v \in V, N_v(0) = 0$.

Reward. Every movement performed by A or D generates a reward depending on the ultimate goal that both of them chase, which determines the score. In this case, A receives one point when a new node is compromised, whereas D obtains the same reward once a previously compromised node has been successfully recovered. A reward for a player p at a time t is denoted by $R^p(t)$.

Cost. Besides a reward, every movement also implies a cost C for the player. This represents the fact that the attacker can exploit vulnerabilities that in turn may cause its detection, while the defender may stop the production chain to recover the security state of a critical resource. It is formalized with $C^p(t)$.

Utility. It is the total number of points scored by a player p at time t. It is calculated as the reward minus the cost of the movement made by p, which is denoted by $U^p(t)$. The overall goal for both players is to maximize the utility as $t \to \infty$, until the game is over.

Strategy. We define a strategy S for a player p as the sequence of movements $M(t)$ along time for a given instance of game, represented by $S^p = \{M^p(0), M^p(1), ..., M^p(t)\}$. As explained later on, this strategy changes as the game evolves: whereas the attacker seeks vulnerable nodes throughout the network while avoiding its detection, the defender follows an adaptive strategy based on the last movement of A (more specifically, on the new state of the affected nodes).

Whereas we consider the utility as a reference for the performance of both players in a given game instance, we define three different termination states:

(TS_1) **Attacker wins.** It is reached when he/she successfully completes all the movements of the strategy S^A, where $S^A = \{M^A(0), M^A(1), ..., M^A(n)\}$. We assume there exists at least one last node v that is compromised, so that $N_v(n) = 1$.

(TS_2) **Defender wins.** It is accomplished when the defender manages to heal all nodes and hence eradicate the effect of the attacker over the entire network, before the succession of movements in S^A are completed. In other words, for a given attacker strategy $S^A = \{M^A(0), M^A(1), ..., M^A(n)\}$, there exists $t' < n$ such that for all $v \in V, N_v(t') = 0$.

(TS_3) **Draw.** For the interest of the analysis, we define an additional third termination condition that occurs when the attacker completes the strategy $S^A = \{M^A(0), M^A(1), ..., M^A(n)\}$ but the defender also performs a last movement that ultimately heals all nodes. In this case, we have that for all $v \in V, N_v(n) = 0$. Even though this may be considered as an attacker win (since he/she succeeds in the disruption of resources), the defender still finds the trace to the threat in the end, which shows the accuracy of the detection technique going after the infection.

With this, the dynamics of the game and the basic rules have been presented. However, we have to describe the precise specification of the players' movements. Whereas the intruder puts into practice a set of individual attack stages that represent an APT (i.e., a strategy of n movements), the defender leverages the Opinion Dynamics algorithm to flexibly adapt to the threat propagation over the network. In both cases, they can apply different actions to change the state of nodes and obtain a score based on different conditions.

3.3 Attacker model: succession of APT stages

As introduced before, we aim to find a formal representation of an APT for the attacker model. In TI&TO, the same authors' methodology in [17] will be used. After an extensive review of the most important APTs reported in recent years, it is possible to specify one of these threats as a finite succession of attack stages perpetrated against an industrial control network defined by the graph $G(V, E)$, so that $attackStages = \{attack\ stage_1, attack\ stage_2, ..., attack\ stage_n\}$. This way, each attack stage corresponds to a different movement performed by the attacker. In the following, we describe the different types of stages and explain their effect on the game board. Then, the reward and cost generated for this player are calculated. Lastly, the strategy creation is explained:

- ***initialIntrusion***$_{(IT,OT,FW)}$. After a phase of reconnaissance, the attacker breaks into the network through a 'patient zero' $v_0 \in V$, that can be a node from the IT or OT section. It is the first movement of the attacker ($M^A(0)$), so that $N_{v_0}(0) = 1$.

- ***LateralMovement***$_{(IT,OT,FW)}$. Once a node v_i has been compromised, the adversary chooses a FW (if it is accessible), IT, or OT node v_j from the set $neighbours(v_i)$ (i.e., those nodes for which there exists one edge $e = (v_i, v_j)$ such that $e \in E$). For the election of the node to take over, we assume that the attacker scans the network in the seek for the most vulnerable device (according to the $VULN$ function). We assume A can compromise a node that has been previously healed by the defender, but its $VULN$ value is then reduced by half.

- ***LinkRemoval***. Once the attacker has perpetrated a lateral movement from v_i towards v_j, that communication channel can be disrupted to decoy the defender (and hence avoid the Opinion Dynamics detection). As a result, the defender cannot exchange the opinion of the agents assigned to v_i and v_j, since no anomaly information is transferred through that link, as explained in the next Section.

- ***Exfiltration of information and Destruction***. It represents the final movement of the attacker. The adversary destroys the node that has been previously compromised, after possibly extracting information that is sent to an external Command&Control network.

Each of these movements results in a different cost and reward for the attacker, who determines his or her utility after each turn of the game, so that the score can be compared with the defender. As for the reward, and aiming to hold the symmetry between both players, they will receive one point every time they gain control of a given node that previously belonged to the adversary. For the attacker, it means that there exists one node $v \in V$ at a time t such that $N_v(t-1) = 0$ and $N_v(t) = 1$ after $M^A(t)$, resulting in $R^A(t) = 1$. For simplicity, we consider that all stages have the same reward.

With respect to the cost of every attack stage, we have to recall the Opinion Dynamics algorithm in relationship with the defender goals. We assume all the network resources are monitored by anomaly detection mechanisms, outputs of which are retrieved by a Opinion Dynamics correlation system. This allows the defender to potentially trace the movement of the attacker along the network, since the different attack stages will generate various security alerts that increase the probability of detection, so it can be conceived as a cost. In [17], authors propose a taxonomy of detection probabilities in form of an ordered set associated with each attack stage. Following the same procedure, here we define Θ as the ordered set of detection probabilities, where $\Theta = \{\theta_1, ..., \theta_n\}$ and $\theta_i = [0, 1]$, such that $\forall \theta_i, \theta_i < \theta_{i+1}$. This model, which is illustrated in Table 2, maps every attack stage to the elements of Θ to represent their cost. The precise election of this taxonomy and quantitative instantiation of the θ values is further explained in Appendix A.

As for the strategy applied for the attacker in TI&TO, S^A will vary depending on the state of surroundings nodes that are vulnerable at every time t of the game. The precise behavior to define the chain of attack stages is the following: S^A always starts with an ***initialIntrusion***, which is randomly chosen from the IT or OT section (hence representing multiple kinds of APTs[7]). Then, A attempts to make a ***LateralMovement***$_{FW}$ movement to compromise a firewall. This movement is straightforward on the IT section as every node is connected to them. However, in case of the OT section, the attacker needs to escalate over the hierarchy of nodes until reaching a PDS node and then the firewall, as explained in Section 3.1. Once there, A penetrates the other section, where we assume he/she must complete a minimum succession of $\sigma = 3$ ***LateralMovements*** (choosing the most vulnerable nodes) before finally executing the ***Destruction*** of a resource. In that case, the game terminates complying

$initialIntrusion(v_0)$	θ_3
$*LateralMovement_{IT,FW}(v_i \to v_j), neighbours(v_i)$	$\theta_4 \to \theta_2, \theta_1$
$*LateralMovement_{OT}(v_i \to v_j), neighbours(v_i)$	$\theta_5 \to \theta_2, \theta_1$
$*LinkRemoval_{(}v_i \to v_j)$	$\theta_5 \to \theta_5$
$destruction(v_i)$	θ_6

Table 2: Map of *attackStages* to Θ

with TS_1 or TS_3, depending on the movements of D. In this sense, the defender can prevent this chain from completing if he/she detects the attacker and successfully eradicates the infection from all nodes (complying with TS_2). In order for the attacker to avoid that situation, a ***LinkRemoval*** can be executed. In TI&TO, D makes this movement when the defender manages to heal $\beta = 3$ nodes in a row, which represents the situation where D is close behind the attacker on the board, as explained in the next Section.

This procedure to define the attacker strategy as the game evolves is formalized in Algorithm 1. Note that the attacker can always follow this chain of stages as long as he/she posses at least one node. In case one is healed, another node is chosen and the APT continues. Otherwise, if the defender manages to heal all victim nodes, the game ends complying with TS_2 or TS_3.

Algorithm 1 Attacker strategy creation

```
output: $S^A$ representing the attacker strategy
local: Graph $G(V, E)$ representing the network, where $V = V_{IT} \cup V_{OT} \cup V_{FW}$, $gameState = 0$ representing initial game state

$S^A \leftarrow \{\}$, $Victims \leftarrow \{\}$, $numSteps \leftarrow 0$
$attackedNode \leftarrow$ random node in $V_{IT} \cup V_{OT}$
$S^A \leftarrow S^A \cup initialIntrusion(attackedNode), Victims \leftarrow Victims \cup attackedNode$
while $gameState == 0$ do
    if defender healed $\beta$ nodes in a row  and  $numSteps < \sigma$ then
        $S^A \leftarrow S^A \cup LinkRemoval$
    else if $attackedNode$ is in first section attacked then
        $S^A \leftarrow S^A \cup LateralMovement_{FW}(nextAttackedNode)$
        $Victims \leftarrow Victims \cup nextAttackedNode$
        $attackedNode \leftarrow nextAttackedNode$
    else if $attackedNode$ is in second section attacked  and  $numSteps < \sigma$ then
        $S^A \leftarrow S^A \cup LateralMovement_{(IT,OT)}(nextAttackedNode)$
        $Victims \leftarrow Victims \cup nextAttackedNode$
        $attackedNode \leftarrow nextAttackedNode$, $numSteps \leftarrow numSteps + 1$
    else
        $S^A \leftarrow S^A \cup Destruction(attackedNode)$, $gameState \leftarrow TS_1$
    end if

    if defender healed $attackedNode$ then
        $Victims \leftarrow Victims \setminus attackedNode$, $numSteps \leftarrow 0$
        if $Victims$ is empty then
            if $gameState == TS_1$ then $gameState == TS_3$
            else
                $gameState \leftarrow TS_2$
            end if
        else
            $attackedNode \leftarrow$ random node in $Victims$
        end if
    end if
end while
```

3.4 Defender model: detection and response

As discussed before, the ultimate goal of this paper is the analysis of the Opinion Dynamics technique against the effects of a realistically-defined APT. As such, we assume that the set of movements that the defender can leverage is summarized in the execution of the algorithm at every turn of the game, followed by an optional node reparation, as described in Section 3.2. Therefore, the defender adopts a dynamic behavior which allows us to analyze the effectiveness of different protection strategies.

We start with the basics. As mentioned in Section 3.2, the defender aims to locate the attacker position across the whole network, keeping track of the anomalies suffered and their persistence over each area of the network as the game evolves. This is enabled by the Opinion Dynamics traceability, as proposed in [17]. Thus, the status of the network is checked by the defender at each turn: then, the most affected node is selected and, based on the severity of the anomaly, he/she finally decides to heal the node. Depending on the accuracy of this action, the defender receives a determined utility. This process, which is henceforth referred to as 'reparation', is described in Algorithm 2. It is repeated successively in each turn of the defender, until all compromised nodes are repaired, complying with the defender-win condition (so that the complexity of the defensive approach is linear) or the attacker completes its set of attack stages. There are some aspects to point out here: firstly, the defender can decide whether to repair the most affected node or stay idle during each turn, which depends on a predefined threshold. Namely, if the opinion given by the agent that monitors that node surpasses it, then the defender opts to heal it. After executing the experiments, and since Opinion Dynamics is calculated as a sum of weighted sum of opinions, this threshold is set to 0.5, which returns the best outcome for the defender.

On the other hand, the reward is one as long as the defender succeeds at healing a node that was in fact compromised; otherwise, the reward is zero. With respect to the cost, it is equivalent to the criticality of the node that is healed (regulated with the $CRIT$ function), in such a way that high-level resources are subject to a potential stop in the production chain and usually need a greater effort in terms of security.

The reparation procedure is the main movement of the defender. However, this reparation strategy can also be influenced by three different configurations:

- ***Local Opinion Dynamics.*** In practice, a global correlation of the Opinion Dynamics agents in a synchronous way may not be feasible in a real industrial environment. Concretely, we aim to demonstrate that the execution of the aforementioned correlation, but considering a subset of nodes of the original network, is effective enough for the defender. Let $G'(V', E')$ be the subgraph of $G(V, E)$ so that $V' \subset V$ and $E' \subset E$. This subgraph is built including a $candidateNode$ and all its child nodes within graph G located at a distance of certain number of hops (in our tests, a distance of one or two hops will be used). The graph G' is used for the computation of the Opinion Dynamics, as usually performed in the original approach. The

Algorithm 2 Reparation of nodes at time t

```
output: $U^D(t)$ representing the utility
local: Graph $G(V, E)$ representing the network, where $V = V_{IT} \cup V_{OT} \cup V_{FW}$
input: $X$ representing the opinion vector of the network agents

$candidateNode \leftarrow$ node in $V$ with maximum $x(t)$
$OldNodeState \leftarrow N_{candidateNode}(t), healThreshold \leftarrow 0.5$
if $x_{candidateNode} > healThreshold$ then
    RepairNode($candidateNode$)
end if
if $OldNodeState == 1$ then
    $N_{candidateNode}(t) \leftarrow 0, R^D(t) \leftarrow 1$
else
    $N_{candidateNode}(t) \leftarrow 0, R^D(t) \leftarrow 0$
end if
$C^D(t) \leftarrow CRIT(candidateNode)$, $U^D(t) \leftarrow R^D(t) - C^D(t)$
```

first election of *candidateNode* is established after $M^A(0)$, considering the highest anomaly measured by the agents over the network. Afterwards, the defender is able to locally compute the correlation and heal nodes in subsequent movements. Thus, at every turn, the *candidateNode* is updated to the node in V' with the greatest opinion, which implies moving the Opinion Dynamics detection zone.

- ***Redundancy of links.*** In Section 3.3 section, the link removal stage was introduced, that allows the attacker to potentially remove links from the topology that make the defender lose track of the threat position, by fooling the local Opinion Dynamics. At this point, we must recall the subset of redundant links $E_R \subset E$ introduced in Section 3.1. These channels will be used by the defender whenever the attacker destroys a link in E, so that opinions will be transmitted using those links only in that case. Despite this may seem as an advantage for the defender, those links can randomly cover pairs of nodes that may not be affected by a link removal. Additionally, the disruption of a link from v_i to v_j in E' does not make v_j inaccessible for the local Opinion Dynamics at all times, since there could be a third node v_k covered by the defender that has another connection $(v_k, v_j) \in E'$.

- ***Honeypots***. For the interest of the analysis, the defender lastly features the possibility of establishing honeypots. It implies modifying the network from the beginning to assign the role of honeypot to specific nodes, which will be randomly chosen in the simulations. These are used as a bait to lure the attacker to compromise them by exposing a higher degree of vulnerability (which was regulated with the $VULN$ function). If the attacker attempts to compromise it, then a higher anomaly will be generated by that agent, which would help the defender to rapidly find the position of the threat, eradicate the threat at a given turn t and hence update the area of the local Opinion Dynamics detection. For our tests, 5% of

Player	Movements	Reward	Cost
Attacker	Initial Intrusion	1	θ_3
	Lateral Movement ($v_i \rightarrow v_j$)	1	θ_4 or θ_5 + θ_1*$\lvert neighbours(v_i)\rvert$
	Link Removal ($v_i \rightarrow v_j$)	1	$2 * \theta_5$
	Destruction (v_i)	1	θ_6
Defender	Node reparation (v_i)	1	$CRIT(v_i)$

Table 3: Summary of movements leveraged by attacker and defender

the total number of nodes have been considered as honeypots, which is a minimal value to show the effectiveness of this response technique.

Table 3 summarizes the set of movements eligible for each player, indicating their reward and cost. Note that the game approach itself is validated from a theoretical point of view in Appendix C. In the following, we run simulations with different configurations for the defender to assess the Opinion Dynamics detection technique.

4 Experimental simulations and discussions

Once both attacker and defender have been described, this section presents the results of playing games under different parameters of TI&TO. As explained, the aim of these experiments is to find the best strategy for the defender given an APT perpetrated by attacker.

In specific, four test cases of games are conducted to assess incremental configurations for the defender' strategy: (1) a local Opinion Dynamics detection around 1 hop of distance from the observed node; (2) local detection with 2 hops of distance: (3) the addition of redundant edges in V_{OT}; and (4) the integration of honeypots within the topology. On the other hand, the attacker follows the model explained in Section 3.3. Each test case is composed by 10 sets of 100 games, where each set is based on a new generated board, following the network architecture introduced in Section 3.1. At the same time, different sizes of network are considered in each test case: 100, 200 and 500 nodes. The instantiation values for their criticality and vulnerability are presented in Appendix A.

For each board and game set, the percentage of victories achieved by each player (in addition to the ratio of draws) is calculated. These are shown in form of a boxplot, where each box represents the quartiles for each player given the different configurations of size in each case. Different conclusions can be drawn from these simulations, which are discussed in the following.

Test Case 1: local Op. Dynamics with 1 hop, no redundancy, no honeypots. In this case (Figure 2), the attacker clearly experiences a high rate of victories as he/she easily escapes from the defender detection, which

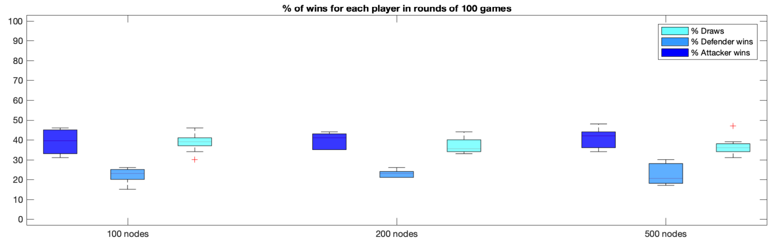


Figure 2: Test-case 1: Percentage of victories and draws

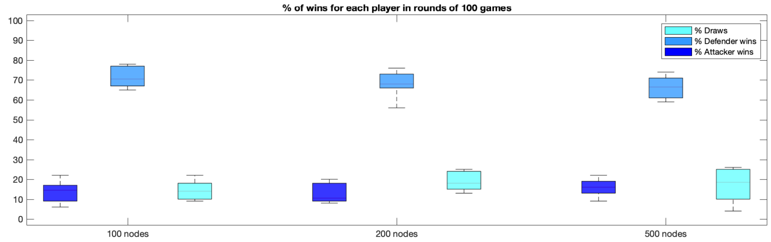


Figure 3: Test-case 2: Percentage of victories and draws

only encompasses one hop of distance from the affected node. Therefore, the best-case scenario for D occurs when he/she just manages to follow the infection until it is eradicated in the last turn, resulting in a draw.

Test Case 2: local Op. Dynamics with 2 hops, no redundancy, no honeypots. With the introduction of more nodes covered by the local detection (whose number is approximately squared with respect to Test case 1), the percentage of defender wins increases significantly, which shows the importance of applying Opinion Dynamics on a wide area, as shown in Figure 3. However, the number of attacker victories and draws still remain moderate, since the defender has not sufficient accuracy as to keep track of A when the removal of links is performed and the detection is eluded.

Test Case 3: local Op. Dynamics with 2 hops, redundancy, no honeypots. The implementation of more defensive aids results in a higher number of wins for the defender (Figure 4). Here, the redundancy makes D able to trace most of the attacker movements, including when that player wants to get rid of the detection, which is more evident in smaller networks. And yet, the defender must successfully heal all the compromised nodes across the network that may continue the attack and be far away from the current detection focus, which still returns a mild number of attacker victories and draws.

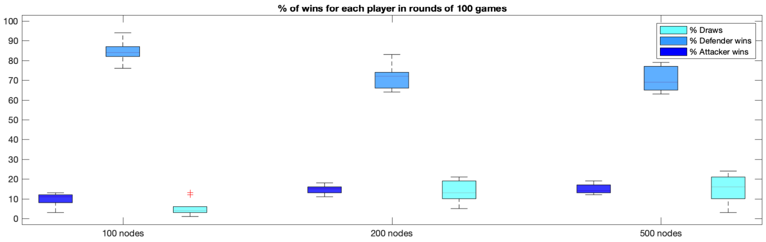


Figure 4: Test-case 3: Percentage of victories and draws

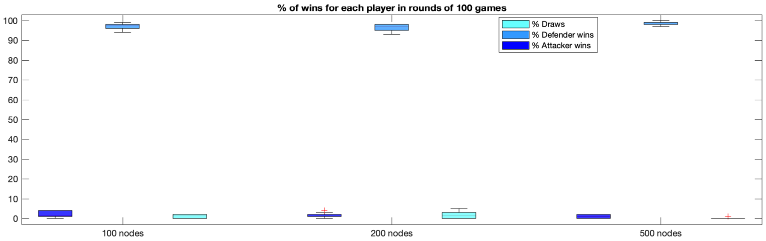


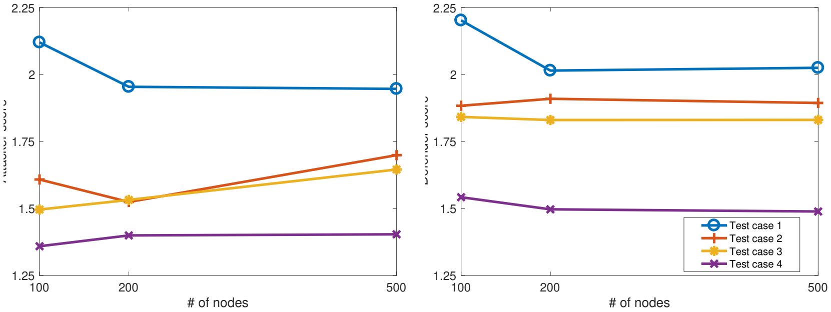
Figure 5: Test-case 4: Percentage of victories and draws

Test Case 4: local Op. Dynamics with 2 hops, redundancy, honeypots. Lastly, the addition of honeypots are a secure way for the defender to ensure the highest number of victories, as shown in Figure 5. The presence of these devices triggers severe anomalies when the attacker tries to compromise then. They are sensed by the defender to rapidly locate the current affected node, as long as D covers a wide area that contains the position of the attacker at that time. This situation is illustrated through an example of game instance in Appendix B.

In general, we can deduce that solely by implementing Opinion Dynamics, the defender can benefit from its detection to reduce the impact of the attacker over the network. The protection improves with the introduction of additional measures such as redundancy or honeypots, and the same results are obtained for different sizes of network.

We can also draw some analysis on the overall score in these test cases: Figure 6 plots the average score of the defender and attacker for the four test cases presented before. At a glance, we can see how D shows a superior throughput in all cases, and a slightly higher score when using low-size networks, since he/she experiences greater accuracy in the reparation of nodes. Also, the score decreases as test cases implement additional defense measures: on the one hand, the attacker generates more anomalies (and hence more costs) due to the link removal attacks in the attempt to dodge the detection. On the other hand, the defender has more candidates to heal due to the increased number of anomalies, and does not always have a high accuracy in choosing them.

To sum up, by means of game theory we have demonstrated that local Opin-



(a) Average of attacker score

(b) Average of defender score

Figure 6: Percentage of victories for each player in each test case

ion Dynamics is still valid for catching the compromised nodes of the attacker when it is applied with a minimally wide detection area (i.e., two hops of distance from the observed node) and it is paired with effective response techniques (i.e., where honeypots pose an effective measure) that precisely make use of the provided detection information.

5 Conclusions

The increasing impact of APTs on modern critical infrastructures demands the development of advanced detection techniques. Opinion Dynamics paves the way towards the effective traceability of sophisticated attacks, as described in this paper. We have leveraged game theory through the design of TI&TO, a two-player game based on a realistic attack and defense model that serves as test-bench for the deployment of response procedures that make use of the information provided by the Opinion Dynamics solution. Based on the execution of multiple games under different configurations, we have extracted guidelines for the correct parametrization of Opinion Dynamics, while we validate the accuracy of the detection technique. Our ongoing work is currently revolving around the reproduction of these test cases on a real environment and the design of an enhanced multi-player game definition that also comprises more than one threat taking place simultaneously. The precise analysis of the optimal parameters for the defender approach (e.g., number of honeypots or thresholds for the Opinion Dynamics detection) will be also carried out.

Acknowledgments

This work has been partially supported by the EU H2020-SU-ICT-03-2018 Project No. 830929 CyberSec4Europe (cybersec4europe.eu), the EU H2020-MSCA-RISE-2017 Project No. 777996 (SealedGRID), and by a 2019 Leonardo

Grant for Researchers and Cultural Creators of the BBVA Foundation. Likewise, the work of the first author has been partially financed by the Spanish Ministry of Education under the FPU program (FPU15/03213).

References

[1] T. Alpcan and T. Basar. A game theoretic analysis of intrusion detection in access control systems. In *2004 43rd IEEE Conference on Decision and Control (CDC)(IEEE Cat. No. 04CH37601)*, volume 2, pages 1568–1573. IEEE, 2004.

[2] K. L. I. CERT. Threat landscape for industrial automation systems. H2 2018. https://ics-cert.kaspersky.com/reports/2019/03/27/threat-landscape-for-industrial-automation-systems-h2-2018/, 2019. Last accessed on September 2019.

[3] T. W. Haynes, S. M. Hedetniemi, S. T. Hedetniemi, and M. A. Henning. Domination in graphs applied to electric power networks. *SIAM Journal on Discrete Mathematics*, 15(4):519–529, 2002.

[4] R. Hegselmann, U. Krause, et al. Opinion dynamics and bounded confidence models, analysis, and simulation. *Journal of artificial societies and social simulation*, 5(3), 2002.

[5] J. Kneis, D. Mölle, S. Richter, and P. Rossmanith. Parameterized power domination complexity. *Information Processing Letters*, 98(4):145–149, 2006.

[6] R. Langner. Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security & Privacy*, 9(3):49–51, 2011.

[7] A. Lemay, J. Calvet, F. Menet, and J. M. Fernandez. Survey of publicly available reports on advanced persistent threat actors. *Computers & Security*, 72:26–59, 2018.

[8] C.-T. Lin. Structural controllability. *IEEE Transactions on Automatic Control*, 19(3):201–208, 1974.

[9] J. Lopez, J. E. Rubio, and C. Alcaraz. A resilient architecture for the smart grid. *IEEE Transactions on Industrial Informatics*, 14:3745–3753, 2018.

[10] K.-w. Lye and J. M. Wing. Game strategies in network security. *International Journal of Information Security*, 4(1-2):71–86, 2005.

[11] K. C. Nguyen, T. Alpcan, and T. Basar. Security games with incomplete information. In *2009 IEEE International Conference on Communications*, pages 1–6. IEEE, 2009.

[12] G. A. Pagani and M. Aiello. The power grid as a complex network: a survey. *Physica A: Statistical Mechanics and its Applications*, 392(11):2688–2700, 2013.

[13] A. Patcha and J.-M. Park. A game theoretic approach to modeling intrusion detection in mobile ad hoc networks. In *Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004.*, pages 280–284. IEEE, 2004.

[14] S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, and Q. Wu. A survey of game theory as applied to network security. In *2010 43rd Hawaii International Conference on System Sciences*, pages 1–10. IEEE, 2010.

[15] J. E. Rubio, C. Alcaraz, and J. Lopez. Preventing advanced persistent threats in complex control networks. In *European Symposium on Research in Computer Security*, volume 10493, pages 402–418. 22nd European Symposium on Research in Computer Security (ESORICS 2017), 09/2017 2017.

[16] J. E. Rubio, M. Manulis, C. Alcaraz, and J. Lopez. Enhancing security and dependability of industrial networks with opinion dynamics. In *European Symposium on Research in Computer Security (ESORICS2019)*, volume 11736, pages 263–280, 2019.

[17] J. E. Rubio, R. Roman, C. Alcaraz, and Y. Zhang. Tracking advanced persistent threats in critical infrastructures through opinion dynamics. In *European Symposium on Research in Computer Security*, volume 11098, pages 555–574, Barcelona, Spain, 08/2018 2018. Springer, Springer.

[18] J. E. Rubio, R. Roman, C. Alcaraz, and Y. Zhang. Tracking apts in industrial ecosystems: A proof of concept. *Journal of Computer Security*, 27:521–546, 09/2019 2019.

[19] M. Simaan and J. B. Cruz. On the stackelberg strategy in nonzero-sum games. *Journal of Optimization Theory and Applications*, 11(5):533–555, 1973.

[20] M. Van Dijk, A. Juels, A. Oprea, and R. L. Rivest. Flipit: The game of "stealthy takeover". *Journal of Cryptology*, 26(4):655–713, 2013.

[21] N. Virvilis and D. Gritzalis. The big four-what we did wrong in advanced persistent threat detection? In *2013 International Conference on Availability, Reliability and Security*, pages 248–254. IEEE, 2013.

[22] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world'networks. *nature*, 393(6684):440, 1998.

A Instantiation of Ψ, Υ and Θ values

In Section 3.3 we have presented an ordered set of probabilities Θ that are mapped to the different attack stages to represent the cost that every movement of the attacker implies, which is summarized in Table 2. There are multiple reasons behing this mapping, that are summarized as follows:

1. We assign the lowest level of detection probability (θ_1) only to the devices in the neighbourhood of the affected node in a lateral movement, since some discovery queries will normally raise subtle network alerts.

2. The second lowest probability of detection (θ_2) is linked to the elements that are the target of a lateral movement, because these connections usually leverage stealthy techniques to go unnoticed.

3. An initial intrusion causes a mild detection probability θ_3, since the attacker either makes use of zero-day vulnerabilities or social engineering techniques, which is a crucial stage for the attacker to be successful at breaking into the network through the ‘patient zero’.

4. θ_4 and θ_5 are assigned to devices (from the IT and OT section, respectively) causing the delivery of malware to establish a connection to an uncompromised node in a lateral movement. In specific, since the heterogeneity of traffic is lower and the criticality of the resources in that segment is greater, anomalies are likely to be detected when compared to the IT section. On the other hand, θ_5 is also assigned to the involved nodes in a link removal stage, since it is an evident anomaly sensed by both agents.

5. The highest probability of detection (θ_6) is assigned to the last stage of the APT, as it usually causes major disruption in the functionality of a device or the attacker manages to connect to an external network to exfiltrate information, which is easily detected.

Considering a realistic scenario and according to the methodology explained in [17], we have assigned values for this ordered set and also for Ψ and Υ sets, which regulate the criticality and vulnerability of resources in our simulations. This instantiation of values is shown in Table 4. For the interest of realism and to represent a certain level of randomness in the accuracy of the detection mechanisms that every agent embodies, these values will also include a random deviation in the experiments, with a maximum value of ±0.1.

i	1	2	3	4	5	6
ψ_i	0.2	0.3	0.4	0.5	0.6	0.8
υ_i	0.8	0.7	0.6	0.5	0.4	0.2
θ_i	0.1	0.3	0.4	0.5	0.6	0.9

Table 4: Instances of the Ψ, Υ, Θ ordered sets used in the simulations

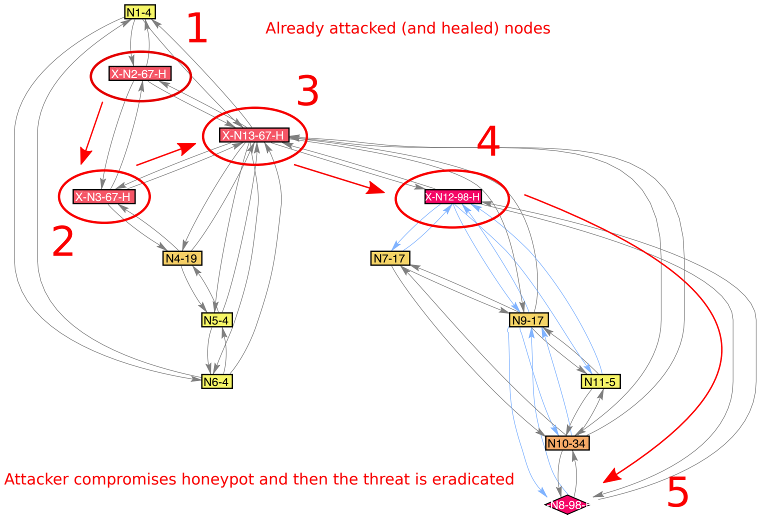

N1-4
1
Already attacked (and healed) nodes
X-N2-67-H
3
X-N13-67-H
4
X-N3-67-H
X-N12-98-H
2
N4-19
N7-17
N5-4
N9-17
N6-4
N11-5
N10-34
Attacker compromises honeypot and then the threat is eradicated
N8-98-
5

Figure 7: Example of defender-win after the attacker compromises a honeypot

B Example of Game Instance with Defender Victory

We have seen that the best results for the defender are achieved when two hops of distance are considered and honeypots are also introduced. In this case, the use of these two tools (besides the redundancy) are enough as to win most of the games. The rationale behind this result is simple: when the attacker attempts to compromise one of this fake nodes, a great anomaly is generated which is detected by the defender, as long as he or she manages to cover a wide area that contains the current position of the attacker (i.e., when 2 or more hops of distance are leveraged by the local Opinion Dynamics). This behavior is shown in Figure 7: in this network, the attacker traverses the nodes and then they are immediately healed (they are labeled with an 'X' when they are attacked and 'H' when they are healed, along with the anomaly measured by Opinion Dynamics). In the last movement, the attacker attempts to compromise a honeypot (depicted with a diamond shape) and the defender manages to locate and eradicate the infection. Since the defender does not possess any other compromised node, the game is over.

C Correctness proof of TI&TO

This section presents the correctness proof of TI&TO for the different cases that may occur during a certain game instance. This problem is solved when these conditions are met:

1. The attacker can find an IT/OT device to compromise within the infrastructure.
2. The defender is able to trace the threat and heal a node, thanks to the Opinion Dynamics detection.
3. The game system is able to properly finish in a finite time (termination condition).

The first requirement is satisfied since we assume that the attacker can perform different attack stages to define his/her strategy over the game board (assuming $V \neq \oslash$), such as lateral movements, links removal or destruction. The modus operandi of the attacker is systematic, beginning with a random node $v_0 \in V_{IT} \cup V_{OT}$ at $t = 0$ which is compromised (see Algorithm 1). Then, A penetrates the infrastructure to ultimately gain control of the operational or corporate network, where a certain node is finally disrupted (V_{OT}) after a set of σ lateral movements. In an intermediate time t of the game, the attacker can execute a new stage as long as there is at least one node v_a such that $N_{v_a}(t) = 1$, which becomes the new $attackedNode$ in Algorithm 1. When the state of all nodes is set to zero, the game terminates.

The second requirement is also met with the inclusion of intrusion detection solutions on every agent $a_i \in A$ that facilitate the correlation of events. With the local execution of the Opinion Dynamics correlation from $t = 1$ on the node that presents the greatest anomaly (using one or two hops of distance), we ensure that the agents associated with the resulting subgraph of nodes will have an opinion $x_i(t) \geq 0$. According to Algorithm 2, this means that D will heal the node with maximum opinion if that value surpasses the threshold (0.5, as explained in Section 3.4), setting its state back to zero and updating the detection area. Otherwise, he/she will remain idle during that turn.

We can demonstrate the third requirement (corresponding to the termination of the approach) through induction. More precisely, we specify the initial conditions and the base case, namely:

Precondition: we assume the attacker models an APT perpetrated against the infrastructure defined by graph $G(V, E)$ where $V \neq \oslash$, following the strategy explained in Algorithm 1. On the other side, the defender leverages Opinion Dynamics to visualize the threat evolution across the infrastructure and eventually repair nodes, following the procedure described in Algorithm 2.

Postcondition: the attacker reaches the network $G(V, E)$ and compromises at least one node in V such that $S^A \neq \oslash$ and continues to compromise more devices in the loop in Algorithm 1, to achieve $numSteps = \sigma$. Player D executes Opinion Dynamics to detect and heal the most affected nodes after executing the correlation. The game evolves until any of the termination states (see Section 3.2) are reached.

Case 1: $numSteps = \sigma$, but $gameState$ is still set to zero. In this case, player A has successfully traversed the network having $Victims \neq \oslash$. Therefore, he/she needs to launch the Destruction movement over the $attackedNode$. This makes $gameState$ comply with TS_1 temporarily until the defender moves. If D manages to heal $attackedNode$ and $Victims = \oslash$, then the game also terminates, with TS_3.

Case 2: $numSteps < \sigma$. In this case, the next stage in S^A implies a lateral movement. If the attacker is still in the first section where the first intrusion took place (whether IT or OT), he/she must locate a firewall to perpetrate the other section before increasing $numSteps$. After this, the defender can make his/her movement and potentially heal a node, which can make the attacker remove a link in the following iteration. If the node healed is $attackedNode$, the attacker must choose another node in $Victims$, resetting $numSteps = 0$. In the event that $Victims = \oslash$, then the game terminates with state TS_2.

Induction: if we assume that we are in step t ($t \geq 1$) in the loop in Algorithm 1, then Case 1 is going to be considered until A completes his/her strategy (TS_1 or TS_3). In any other case, Case 2 applies until achieving $numSteps = \sigma$ (hence applying Case 1 again) or $Victims = \oslash$. In this last case, the game finishes with TS_2.