



Optimal Repair Strategy Against Advanced Persistent Threats Under Time-Varying Networks

Wang, Zixuan; Li, Jiliang; Wang, Yuntao; Su, Zhou; Yu, Shui; Meng, Weizhi

Published in:
IEEE Transactions on Information Forensics and Security

Link to article, DOI:
[10.1109/TIFS.2023.3318954](https://doi.org/10.1109/TIFS.2023.3318954)

Publication date:
2023

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Wang, Z., Li, J., Wang, Y., Su, Z., Yu, S., & Meng, W. (2023). Optimal Repair Strategy Against Advanced Persistent Threats Under Time-Varying Networks. *IEEE Transactions on Information Forensics and Security*, 18, 5964-5979. <https://doi.org/10.1109/TIFS.2023.3318954>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Optimal Repair Strategy Against Advanced Persistent Threats Under Time-Varying Networks

Zixuan Wang, Jiliang Li, Yuntao Wang, Zhou Su, *Senior Member, IEEE*, Shui Yu, *Fellow, IEEE*, and Weizhi Meng, *Senior Member, IEEE*

Abstract—Advanced persistent threat (APT) is a kind of stealthy, sophisticated, and long-term cyberattack that has brought severe financial losses and critical infrastructure damages. Existing works mainly focus on APT defense under stable network topologies, while the problem under time-varying dynamic networks (e.g., vehicular networks) remains unexplored, which motivates our work. Besides, the spatiotemporal dynamics in defense resources, complex attackers' lateral movement behaviors, and lack of timely defense make APT defense a challenging issue under time-varying networks. In this paper, we propose a novel game-theoretical APT defense approach to promote real-time and optimal defense strategy-making under both periodic time-varying and general time-varying environments. Specifically, we first model the interactions between attackers and defenders in an APT process as a dynamic APT repair game, and then formulate the APT damage minimization problem as the precise prevention and control (PPAC) problem. To derive the optimal defense strategy under both latency and defense resource constraints, we further devise an online optimal control-based mechanism integrated with two backtracking-forward algorithms to fastly derive the near-optimal solution of the PPAC problem in real time. Extensive experiments are carried out, and the results demonstrate that our proposed scheme can efficiently obtain optimal defense strategy in 54481 ms under seven attack-defense interactions with 9.64% resource occupancy in stimulated periodic time-varying and general time-varying networks. Besides, even under static networks, our proposed scheme still outperforms existing representative APT defense approaches in terms of service stability and defense resource utilization.

Index Terms—Cybersecurity, advanced persistent threat (APT), node-level epidemic model, APT repair game, time-varying networks.

I. INTRODUCTION

RECENTLY, many well-protected organizations, such as multinational companies or government departments, have been subjected to various cyber attacks that cause huge financial losses, disclosure of business secrets, and even damages to nation-critical infrastructures [1]–[3]. According to 360 Security's statistics [4], advanced persistent threats (APT) such as Stuxnet, Duqu, Flame, and Gauss, account for nearly 60% of the attacks on government departments and multinational corporations in the past two years.

To launch a typical APT attack, a team of collusive hackers generally use sophisticated, distributed, and stealth attack tech-

This work is supported in part by National Key R&D Program of China (No. 2022YFB2702800), and in part by NSFC (Nos. 62102305, 62302387, U22A2029, U20A20175), China Postdoctoral Science Foundation under Grant 2023M732820.

Z. Wang, J. Li, Y. Wang, and Z. Su are with the School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an 710049, China.

S. Yu is with the School of Computer Science, University of Technology Sydney, Sydney 2007, Australia.

W. Meng is with the Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU), Copenhagen 2800, Denmark.

nologies to (i) gain illicit access to the target system (e.g., via malware), (ii) infiltrate target organization via privilege escalation and lateral movement attacks, and then (iii) achieve illegal purposes. Essentially, APT attacks are characterized by long-term attack cycles, highly sophisticated changes, and unpredictable lateral movements, thereby causing highly complex attack patterns and difficulties in attack path tracing and system repair.

Due to the characteristics and hazards of APT attacks, how to defend against APT attacks has become a hot topic of research in academia and industry. There already have been various works [5]–[11] focusing on APT defense, and many of them are based on the game theory. For example, Yang *et al.* [5] proposed a game-theoretical APT risk management strategy for defenders (e.g., the security department of an organization) to optimally deploy defense resources with mitigated APT attack loss, in which both the assets of the organization and the behavioral returns of the adversary are quantified. Li *et al.* [8] presented a Lyapunov-based security-aware defense mechanism and employed the game theory to seek the equilibrium between the network defenders and APT attackers for optimally allocating the defensive resources with maximized system utility. Yang *et al.* [7] modeled the APT repair problem as an APT defense game and presented a greedy algorithm to seek the Nash equilibrium as the optimal defensive response strategy.

However, existing works mainly assume that the network topology of an organization (e.g., government department and operator of an IoT network) remains stable during APT defense, which is inapplicable to general scenarios with dynamic network topology (e.g., vehicular network [12], [13] and UAV network [14], [15]). Many researchers already have recognized the impact of network typology's dynamicity on real-world attacks and defenses, especially in the field of Internet of Things (IoT) [16]–[18]. Besides, the highly dynamic network topology can result in more complex attackers' lateral movement patterns, as well as a necessity for online low-latency APT defense strategy-making under the constraint of defensive resources. For example, the dynamic nature of satellite networks, exemplified by the successful Hack-A-Sat competition [19] on Aug 18, 2023 at DEF CON [20], highlights how the constantly changing topology can be exploited by attackers, compromising even well-defended satellites (details refer to Appendix A). Particularly, the following key challenges must be addressed to practically deploy the game-theoretical APT countermeasures.

- *Dynamical defense resources.* Typically, to recover from the APT attack and offer normal services, an organization's defensive resources (e.g., audit and bandwidth resources) are usually limited. Moreover, depending on practical applications, the organization's defensive resources might be

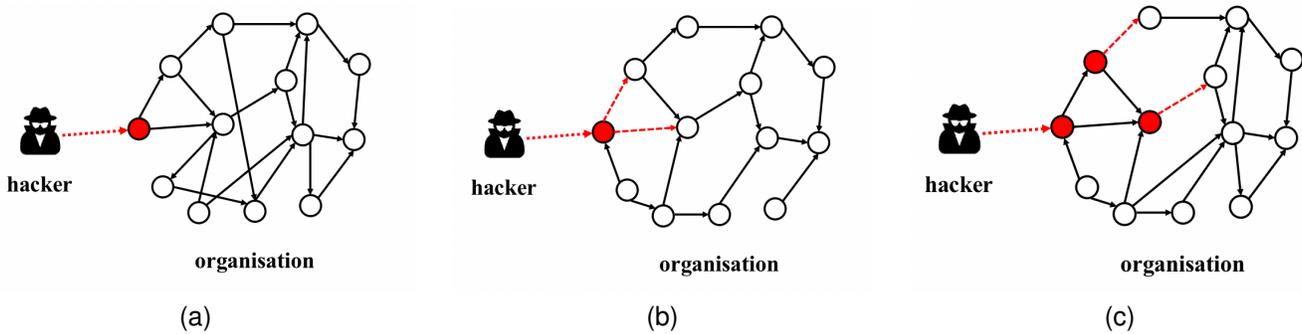


Fig. 1: The lateral movement attack on the entire dynamic network organization after the attacker obtains the ingress machine. The red nodes indicate the machines that have been hijacked. The red dashed lines indicate the potential targets of the lateral movement attack. The change of topology in (a)-(c) indicates the change of network topology.

dynamic in terms of space and time.

- *Inefficiency in time-varying networks.* As shown in Fig. 1, the lateral movement attack (i.e., the attacker penetrates other organization nodes through its compromised nodes) under a dynamic network can severely damage the network's functionality. Furthermore, the time-varying network topology complicates lateral movement attacks, resulting in higher uncertainty and inefficiency in defensive resource allocation.
- *Lack timely defense.* Since the network topology changes over time, the defensive behaviors of defenders should be timely adapted and enforced on the entire network. Besides, considering the impact of defenders' behaviors in the time-varying network topology, there exists a curse of dimensionality due to the huge space of state and action sets.

To bridge the research gap, this paper proposes a fast and efficient APT defense strategy tailored to time-varying networks under both defense resource limitations and network dynamics. Specifically, we model the interactions between APT attackers and defenders (i.e., the network operator) under highly dynamic environments as a sequential APT repair game with the aim of minimizing the damage of the APT attack to the organization. We regard the APT attack process as a finite time horizon divided into multiple time slots, as shown in Fig. 1. The interactions between APT attackers and defenders in each time slot are modeled as a subgame of the dynamic APT repair game. Based on the sequential APT repair game, an optimization problem, i.e., precise prevention and control (PPAC), is formulated under both latency and defense resource constraints.

Considering the time-varying network topology, threat impacts, hosts' differentiated states, and spatio-temporal variability of defensive resources, we devise an online fast searching mechanism based on optimal control to derive the near-optimal solution of the PPAC problem in real time. We summarize the main contributions as follows.

- *Dynamic APT Repair Problem Modeling.* We model the interactions between APT attackers and defenders in each time slot as a subgame of the dynamic APT repair game. The sequential APT repair game is formulated as a precise prevention and control (PPAC) optimization problem, whose solution is the online optimal defense strategy. To the best

of our knowledge, this is the first attempt at the defense strategy against APT attacks under time-varying networks.

- *Efficient and Practical Solving Method.* To solve the PPAC problem, we present a practical online fast strategy searching mechanism to seek the optimal APT defense strategy which minimizes the impact of APT. It consists of two backtracking-forward algorithms and an optimal system. The two backtracking-forward algorithms, i.e., the threat rate grading (TRG) algorithm and the recovery rate grading (RRG) algorithm are developed to compute the nodes' states at each moment. Based on nodes' states, the designed optimal system can overcome the challenges caused by topology changes and complex APT attack patterns to derive the optimal defense strategy.
- *Extensive Experimental Validation.* Experiments on three scenarios (i.e., static, periodic time-varying, and general time-varying networks) demonstrate the feasibility and effectiveness of the proposed scheme. Results from experiments validate that the proposed scheme can reduce the impact of APT attacks on system utility and save defense resources in time-varying dynamic networks. Compared with existing representatives, our proposed approach can effectively enhance service stability to 100% and defense resource utilization to 75.50% in stimulated time-varying networks, thereby greatly enhancing both the service utility and defense performance.

The rest of the paper is organized as follows. Section II reviews the related works, and Section III introduces the system model. Section IV introduces the sequential APT repair game model and formulates the PPAC problem. Section V introduces the online fast strategy searching mechanism to solve the PPAC problem, and Section VI evaluates the performance of the proposed scheme. Section VII summarizes this work using conclusions.

II. RELATED WORK

This section presents a comprehensive review of attack propagation in APT scenarios and provides an in-depth analysis of the existing literature on defense strategies. We review the attack dissemination related literature (in Sect. II.A), game-theoretical APT defense (in Sect. II.B), and optimal control-based APT defense (in Sect. II.C).

A. Attack Dissemination

The attack dissemination, also referred to as the lateral movement phase [21] of a cyber attack, is intricately influenced by the network topology. Extensive research has explored the complex dynamics of propagation, particularly within the domains of synchronization and consensus in static or dynamic multi-agent systems [22].

Kephart *et al.* [23] and Staniford *et al.* [24] are among the first to identify that the propagation of Internet worms exhibits similarities to the spread of epidemics, posing a significant threat to system security. Building upon their pioneering work, subsequent research in the field has continued to explore the analogy between cyber attacks and epidemic spread. For instance, Wang *et al.* [25] apply epidemic models to study the propagation of viruses in mobile phones through multimedia messaging services or Bluetooth connections. Cheng *et al.* [26] extend the application of epidemic models by simulating the propagation of malicious software within generalized social networks comprising both non-local and local connections. Their research further validates the suitability of epidemic models in capturing the dynamics of attack dissemination in such networks.

Another notable contribution comes from Darling *et al.* [27], who demonstrate the efficacy of ordinary differential equations (ODEs) in approximating the dynamics of information dissemination from Markov chains. And, Lapidus *et al.* [28] prove that the backtracing-forward algorithm demonstrates exceptional performance in solving such problems. Therefore, the utilization of Markov chains in conjunction with node states to construct epidemic models [29] has extensive applications in APT defense [6], [7].

B. Game-theoretical APT Defense

Recently, various research efforts have focused on using game theory to find efficient defense strategies for APT attacks. The pioneering work of using game theory (the FlipIt game) to model the payoffs and perform equilibrium analysis for both attackers and defenders was proposed by van Dijk *et al.* [30].

Subsequently, the FlipIt game is used in various cyber situations [10], [11], which contains multiple resources, information leakage, etc. In these applications, [31] uses the PBNE to discuss the APT-related defense strategies.

Zhu *et al.* [32] categorize APT attacks into three distinct phases and introduce a comprehensive framework based on matrix and sequential games. By iteratively determining the optimal defense strategy at each stage, the overall best defense strategy is derived.

In addition to applying game theory to study some general aspects of APT attacks, several significant works have been conducted on resource-limited defense strategies. These strategies focus on optimizing resource utilization and developing effective countermeasures to combat APT threats. This kind of strategies is commonly known as the repair strategy [7]. To identify the most optimal approach, the concept of a repair game has been proposed. In line with this objective, Ye *et al.* [33] study a new differential game based on network spoofing and developed differential privacy methods for Nash equilibrium. But they ignore the lateral movement of each compromised host. Yang *et al.* [7] discuss the control strategy approach for lateral movement in traditional network architectures with a differential

game approach. They give a potential Nash equilibrium under this game from the perspective of the defender and give a reference for the defender's defense strategy. However, they only briefly discuss repair methods while ignoring the diversity of repair methods. In [6], Yang *et al.* propose the idea of isolation zones to solve the APT repair problem in a more reasonable way which constructs a two-order dynamical system to solve the control set for the QAR (quarantine and recovery) problem.

However, all the works mentioned above neglect the dynamic nature of topology in dynamic networks, thereby, cannot be applied to dynamic networks.

C. Optimal Control Model for APT Defense

Besides the game-theoretical approach, various works use optimal control to defend against APT attacks. Optimal control theory is used to find the required control law to make a given system perform optimally and is widely used in cybersecurity. The defender simply has to be aware of the average values of a few factors connected to the APT when modeling the APT defense problem using optimal control. As compared to other methods, estimating these factors requires less real APT data, which the APT attack and defense exercises [34] can be used to estimate these factors. Li *et al.* proposed [9] the optimal control methods to solve the APT problem. They used a node-level contagion model to evaluate changes in the state of nodes in the maintenance scheme, which modeled the APT response problem as an optimal control problem. In [35], Zhao *et al.* modeled the isolation-conversion problem as an optimal control problem to find the optimal isolation strategy to reduce the negative impact of rumors. Inspired by [35], Yang *et al.* [6] further considered the isolation operations essential to the APT response process.

However, existing works on APT defense mainly focus on the networks with stable structures (e.g., the internal network inside an organization) and are inapplicable to time-varying networks (e.g., vehicular networks). Besides, these works ignore the impacts of defense behaviors on the utility of the network operator, which eventually deteriorates the defensive effectiveness. Different from existing works, our work jointly considers the dynamic topological characteristics of the nodes and the impacts on the service utility in time-varying networks to develop fast and effective APT defensive strategies.

III. SYSTEM MODEL

This section introduces the system model, including the network model, lateral movement attack model, and potential APT impact model. The key notations in this paper are listed in Table I.

A. Network Model

In the APT attack-defense scenario, it involves an attacker (denoted by \mathbb{A}) and a network operator (i.e., defender, denoted by \mathbb{D}) who operates V number of hosts. Here, \mathbb{A} can be an attacker group conspired to launch the APT attack, and \mathbb{D} can be the cyber security department of an organization (i.e., network operator). At the same time, V also represents the organization's valuable assets (e.g., business secrets). Since the defender is far-sighted¹

¹As the APT attack is persistent, the defender aims to reduce the total loss in the whole APT attack process.

and aims to minimize the loss during the APT process, we study the variation of APT lateral movement attack and the expected impact of APT in a finite time horizon, which is divided into multiple time slots. The time horizon of the defense process is denoted as $[0, T]$, which is determined by the length of the APT process. We denote the sequence of the APT process as

$$\hat{T} = \{A_1 = (0, t_1), A_2 = (t_1, t_2), \dots, A_m = (t_{m-1}, t_m)\}, \quad (1)$$

where $t_j (1 \leq j \leq m)$ means the periods of the APT attack. Due to the characteristics of the time-varying network, we represent the periods of APT by the topology adjacency matrix of the network. Each element $A_j (1 \leq j \leq m)$ is a topology adjacency matrix. For example, if $0 \leq t \leq t_1$, the adjacency matrix $A(t) = A_1$. We denote the links of network's devices as $G(t) = \{V, E(t), A(t)\}$, where $V = \{v_1, v_2, \dots, v_N\}$ stands for the sets of all devices and the edge $\{v_i, v_j\} \in E(t)$ stands for the relation between devices v_i and v_j at time t . Let $A(t) = (a_{ij})_{N \times N}$ denote the adjacency matrix of network G at time t , $a_{ij} = 1$ or 0 represents that there exist a link between v_i and v_j or not, respectively.

Assume the defender discovers the APT warning at time slot $t = 0$ and replies within the allotted time window $[0, t]$ by carrying out several defense activities. Each node has four possible states during the response period at any given time slot: *healthy*, *mild*, *severe*, and *quarantined*. The *quarantined* state stands for the nodes that break all the links. The *healthy* state represents the nodes that do not suffer from attacks. The *mild* state stands for the compromised nodes whose impact on the network is acceptable. In other words, the *mild* nodes bring more revenue than the loss they caused. The *severe* state represents the compromised nodes whose impact on the network is fatal, which means they need to be repaired immediately. For each node v_i , let $X_i(t) = 0, 1, 2$, and 3 denote that v_i is *healthy*, *mild*, *severe*, and *quarantined* at time t , respectively. Then the state of each node at time t can be denoted as:

$$\mathbf{X}(t) = (X_1(t), X_2(t), \dots, X_N(t)). \quad (2)$$

We provide the following set of notations to express the network's evolution process for the $\mathbf{X}(t)$ variable.

- 1) We define the PPAC control factor as a $3N$ -dimensional function \mathbf{w} , i.e.,

$$\mathbf{w}(t) = (\mathbf{\Lambda}(t), \mathbf{\Delta}(t), \mathbf{\Gamma}(t)), \quad 0 \leq t \leq T \quad (3)$$

$\mathbf{W} = (\mathbf{w}(1), \dots, \mathbf{w}(j))$ represents the control vector of whole APT process. Here, the average rate is a fundamental concept in stochastic process theory [36] whose reciprocal represents the mean time before the event occurs.

- 2) The average rate at which each compromised node v_i is categorized as mild or severe is represented by the notation $\lambda_i(t)$ ($\lambda_i(t)$ is severe, $1 - \lambda_i(t)$ is mild). We refer to the N -dimensional function $\mathbf{\Lambda}$ defined by

$$\mathbf{\Lambda}(t) = (\lambda_1(t), \dots, \lambda_N(t)), \quad 0 \leq t \leq T \quad (4)$$

as the threat vector.

- 3) The average rate of quarantining for each severe node v_i at time t is represented by the notation $\delta_i(t)$. We refer to the N -dimensional function $\mathbf{\Delta}$ defined by

$$\mathbf{\Delta}(t) = (\delta_1(t), \dots, \delta_N(t)), \quad 0 \leq t \leq T \quad (5)$$

TABLE I: Key Notations

| Notation | Description |
|----------------------------|---|
| $G(t) = \{V, E(t), A(t)\}$ | The topological graph of the server network, ξ denotes the edge set of the graph at time t , $A(t) = (a_{ij}^t)_{N \times N}$ denotes the adjacency matrix of the G at time t . |
| \hat{T} | The topology schedule of the task. $\hat{T} = \{A_1 = (0, t_1), A_2 = (t_1, t_2), \dots, A_m = (t_{m-1}, t_m)\}$ |
| T | Time to perform the task. |
| $X_i(t) = \{0, 1, 2, 3\}$ | The node v_i is healthy, mild patients, severe patients, and quarantined at time t , respectively. The vector $\mathbf{X}(t) = (X_1(t), \dots, X_N(t))$ represented the state of network at time t . |
| $\beta_i(t)$ | The average rate of direct infiltration that compromises a healthy node v_i . Let $\mathbf{\beta}(t) = (\beta_1(t), \dots, \beta_N(t))$ |
| $\alpha_i(t)$ | The average rate of lateral movement from any nearby compromised APT node that compromises a healthy node v_i . Let $\mathbf{\alpha}(t) = (\alpha_1(t), \dots, \alpha_N(t))$. |
| $\lambda_i(t)$ | The average rate at which compromised node v_i is classified as mild or severe. Let $\mathbf{\Lambda}(t) = (\lambda_1(t), \dots, \lambda_N(t))$. The function $\mathbf{\Lambda}$ represents the threaten scheme. |
| $\delta_i(t)$ | The average rate at which compromised node v_i is quarantined or not. Let vector $\mathbf{\Delta}(t) = (\delta_1(t), \dots, \delta_N(t))$. The function $\mathbf{\Delta}$ represents the quarantined scheme. |
| $\gamma_i(t)$ | The average rate at which severe patients node v_i is recover to mild patients or healthy. Let vector $\mathbf{\Gamma}(t) = (\gamma_1(t), \dots, \gamma_N(t))$. The function $\mathbf{\Gamma}$ represents the recovery scheme. |
| a_i^1, a_i^2 | The average consumption per unit time of M_i and S_i . Vectors $\mathbf{a}^1 = (a_1^1, \dots, a_N^1)$ and $\mathbf{a}^2 = (a_1^2, \dots, a_N^2)$ denote resource impact value vector of the network, respectively. |
| b_i | The average impact of services impact per unit of time. is referred to as a precise prevention and control scheme. $\mathbf{b} = (b_1, \dots, b_N)$ denotes services impact value vector of the network. |
| ϕ_i | The quarantine cost function of the node v_i . Vector $\mathbf{\phi} = (\phi_1, \dots, \phi_N)$ represents the quarantine cost function of the network. |
| ϱ_i^1, ϱ_i^2 | The recovery cost function to M_i and S_i . Vector $\mathbf{\varrho}^1 = (\varrho_1^1, \dots, \varrho_N^1)$ and vector $\mathbf{\varrho}^2 = (\varrho_1^2, \dots, \varrho_N^2)$ represent the recovery cost of the network. |
| R, T | The output of algorithm TRG and RRG, which represent the threaten states and recovery states of the network. |
| U_n, U | U represents the utility of the entire network. U_n represents the requirement of maintaining minimum service effectiveness. |
| E_0 | E_0 represents the average rate of the initial state of nodes. |
| \mathbf{w} | $\mathbf{w}(t) = (\mathbf{\Lambda}(t), \mathbf{\Delta}(t), \mathbf{\Gamma}(t))$ is referred to as a precise prevention and control scheme. |

as the quarantined vector.

- 4) The average rate at which a confined node v_i becomes healthy or mild at time t is expressed as $\gamma_i(t)$ for each quarantined node v_i . We refer to the N -dimensional function $\mathbf{\Gamma}$ defined by

$$\mathbf{\Gamma}(t) = (\gamma_1(t), \dots, \gamma_N(t)), \quad 0 \leq t \leq T \quad (6)$$

as the recovery vector.

- 5) The healthy nodes of the network that suffered from attacks are defined as follows. The healthy node v_i becomes compromised as a result of APT penetration, as represented

by $\beta_i(t)$. The probability that the healthy node v_i becomes compromised due to lateral movement from surrounding compromised nodes is represented by the average infection rate $\alpha_i(t)$. Two statistical properties of the task are $\beta_i(t)$ and $\alpha_i(t)$, which are denoted as $\beta(t) = (\beta_1(t), \dots, \beta_N(t))$ and $\alpha(t) = (\alpha_1(t), \dots, \alpha_N(t))$, respectively. Note that $\beta(t)$ and $\alpha(t)$ can be calculated using APT attack-defense maneuvers [34].

Based on the above notations, we can obtain the state transition of the node v_i at time slot t as shown in Fig. 2 through Markov chain theory. The details of the state transition model are shown in Section III-B.

B. Modelling the Lateral Movement Attack

The time-varying network topology makes lateral movement attacks more complex and difficult to predict. To accurately portray the impact of the lateral attack, we adapt the Markov process and causal analysis to obtain a response model.

Based on the [29], we can get the state transition function:

$$\dot{x}(t) = (1 - x_i)\beta_i(t) \sum_{j=1}^n a_{ij}(t)x_j - \delta_i(t)x_i(t). \quad (7)$$

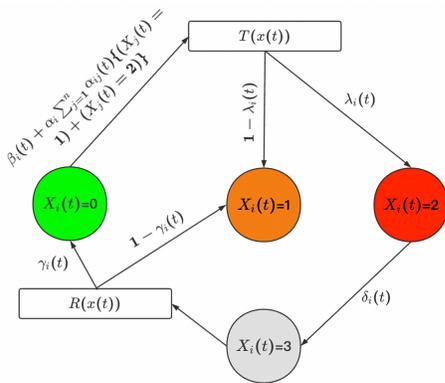


Fig. 2: The state transition of node v_i at time t .

Considering both direct infiltration attacks and lateral movement attacks, we define $\mathbf{X}_i(t)$ as:

$$\mathbf{X}_i(t) = \beta_i(t) + \alpha_i \sum_{j=1}^n a_{ij}(t) \{ (X_j(t) = 1) + (X_j(t) = 2) \}. \quad (8)$$

The state transition process is shown at time t in Fig. 2. We use $H_i(t)$, $M_i(t)$, and $S_i(t)$ to denote the probabilities of node v_i being *healthy* state, *mild* state, and *severe* state at time t , respectively. Each state can be expressed as:

$$\begin{aligned} H_i(t) &= Pr\{X_i(t) = 0\}, & M_i(t) &= Pr\{X_i(t) = 1\}, \\ S_i(t) &= Pr\{X_i(t) = 2\}. \end{aligned} \quad (9)$$

If the node v_i is in *quarantined* state at time t , the probability is $1 - H_i(t) - M_i(t) - S_i(t)$. Vector $\mathbf{E}(t)$ represents the expected state of the network at time t . In practice, we can get the initial state $\mathbf{E}(0)$ by the APT detection program in [37]–[39]. The vector $\mathbf{E}(t)$ can be expressed as:

$$\mathbf{E}(t) = (H_1(t), \dots, H_N(t), M_1(t), \dots, M_N(t), S_1(t), \dots, S_N(t)). \quad (10)$$

Therefore, we have the following theorem to describe the network's expected state evolution.

Theorem 1: The network's expected state evolves according to the following differential dynamical system:

$$\begin{cases} \frac{dH_i(t)}{dt} &= \gamma_i(t)[1 - M_i(t) - S_i(t) - H_i(t)] - H_i(t) * \\ &[\beta_i(t) + \alpha_i \sum_{j=1}^n a_{ij}(t)M_j(t) \\ &+ \alpha_i \sum_{j=1}^n a_{ij}(t)S_j(t)] \\ \frac{dM_i(t)}{dt} &= H_i(t)[\beta_i(t) + \alpha_i \sum_{j=1}^n a_{ij}(t)M_j(t) \\ &+ \alpha_i \sum_{j=1}^n a_{ij}(t)S_j(t)] - \lambda_i(t)M_i(t) \\ \frac{dS_i(t)}{dt} &= \lambda_i(t)M_i(t) - \delta_i(t)S_i(t) \\ \mathbf{E}(0) &= \mathbf{E}_0 \end{cases} \quad (11)$$

Proof: We use \mathbb{E} to denote the mathematical expectation. The expected rate at which *healthy* state node v_i transforms to *severe* state or *mild* state at time t is

$$\begin{aligned} \mathbb{E} \left(\mathbf{X}_i(t) = \beta_i(t) + \alpha_i \sum_{j=1}^n a_{ij}(t) \{ (X_j(t) = 1) + (X_j(t) = 2) \} \right) \\ = \mathbf{X}_i(t) = \beta_i(t) + \alpha_i \sum_{j=1}^n a_{ij}(t) (M_i(t) + S_i(t)), \end{aligned} \quad (12)$$

and the derivation of the conversion process of other states in Fig. 2 is the same. Hence, we can get the Eq. (11). ■

Remark 1: Theorem 1 illustrates the differential equations of each state we defined by finding the expectation for the state function Eq. (8) with different condition restrictions. For example, *healthy* state nodes are composed of attack and recovery two parts at time slot t . On the attack part, each *healthy* node compromise to *mild* or *severe* state by direct attack and lateral movement attack, and the compromised probability is $\beta_i(t) + \alpha_i \sum_{j=1}^n a_{ij}(t) (M_i(t) + S_i(t))$. On the recovery part, each *quarantined* node transforms to the *healthy* state by the defenders' recovery operations whose recovery rate is $\gamma_i(t)$.

C. Modelling the Expected Impact of APT Attack

For the stealthiness of attacks, the attacker hacks the organization slowly via insecure hosts. Therefore, the amount of loss per unit of time of the organization is limited. In this paper, we assume that the attacker will intelligently exploit different insecure hosts to expand the erosion of the organization.

We consider the following three main factors of organizational loss caused by the APT attack (i.e., the APT impact on the organization). 1) The consumption of resources by the hijacked hosts. 2) The service utility loss of the attacked host and its impact on the overall service. 3) The cost of adapting the defense strategies to control APT attacks (e.g., cutting off the connection between hosts or system updates).

1) Resource consumption. For each compromised node v_i , we define a_i^1 and a_i^2 as the average consumption per unit time of M_i and S_i , respectively. Let $\mathbf{a}^1 = (a_1^1, \dots, a_N^1)$ and $\mathbf{a}^2 = (a_1^2, \dots, a_N^2)$ denote resource impact value vector of the network, respectively. In practice, the average consumption can be estimated by the resource per unit of time (e.g., bandwidth usage per unit of time and WAF placement per unit of time). Therefore, the following theorem is introduced to define resource consumption in a time period.

Theorem 2: Due to the consumption of resources, the expected consumption in time horizon $[0, T]$ is

$$L(\mathbf{w}) = \int_0^T \sum_{i=1}^N [a_i^1 M_i(t, \mathbf{w}(t), G(t)) + a_i^2 S_i(t, \mathbf{w}(t), G(t))] dt. \quad (13)$$

Proof: The average consumption in the infinitesimal time horizon $[t, t + dt]$ is $a_i^1 (X_i(t) = 1)$ and $a_i^2 (X_i(t) = 0)$. The expected consumption in $[t, t + dt]$ is $a_i^1 M_i(t, \mathbf{w}(t), G(t)) dt + a_i^2 S_i(t, \mathbf{w}(t), G(t)) dt$. ■

Remark 2: From the consumption history data of the specified hosts after being attacked, defenders can obtain the M_i and S_i state hosts' average consumption a_i^1 and a_i^2 , respectively. In this paper, we assume the average consumption is constant in the infinitesimal time. Through the parameters a_i^1 and a_i^2 , we can define the consumption in the infinitesimal time of one node as $a_i^1 M_i(t, \mathbf{w}(t), G(t)) + a_i^2 S_i(t, \mathbf{w}(t), G(t))$. In this way, the integral formulation can calculate resource consumption for a given time period.

2) Service utility loss. For each quarantined node v_i , we define b_i as the average impact of service per unit of time. $\mathbf{b} = (b_1, \dots, b_N)$ denotes services impact value vector of the network. In actuality, the vector can be calculated by examining the server's traffic flow. Based on the above, Theorem 3 is introduced to define service utility loss in a time period.

Theorem 3: Due to the impact of services, the expected impact in the time horizon $[0, T]$ is

$$E(\mathbf{w}) = \int_0^T \sum_{i=1}^N [b_i(1 - M_i(t, \mathbf{w}(t), G(t)) - S_i(t, \mathbf{w}(t), G(t)) - H_i(t, \mathbf{w}(t), G(t)))] dt. \quad (14)$$

Proof: The proof of service's impact is comparable to the proof in Theorem 2, and we omit it here. ■

Remark 3: In our state evolution model, we perform quarantine operations based on the threat rate for the hijacked hosts. So the service utility loss for a given period is only considered for the nodes in the quarantine state, and it is simple to get the service parameter b_i of each host for the defender. Through the parameters b_i , we define the service utility loss in the infinitesimal time of one node as $(1 - M_i(t, \mathbf{w}(t), G(t)) - S_i(t, \mathbf{w}(t), G(t)) - H_i(t, \mathbf{w}(t), G(t)))$. The integral formulation can calculate the service utility loss for a given period.

3) Expected repair cost. For each *severe* state node, $\phi_i(\delta)$ denotes the cost of quarantining at the average rate δ . We define ϕ_i as the quarantine cost function of the node v_i . Therefore, the quarantine cost function of the network is represented as a vector $\phi = (\phi_1, \dots, \phi_N)$. The function ϕ_i is strictly increasing and $\phi_i(0) = 0$. In practice, we can approximate ϕ through statistical regression of APT attack-defense maneuvers [34]. The dynamic quarantine schedule, denoted by

$$(\phi_1(\delta_1(t)), \dots, \phi_i(\delta_N(t))) \quad 0 \leq t \leq N, \quad (15)$$

is decided directly by the defender.

For each quarantined node v_i , we define ϱ_i^1, ϱ_i^2 as cost functions to M_i and S_i , respectively. Therefore, the recovery cost of the network is represented as vector $\varrho^1 = (\varrho_1^1, \dots, \varrho_N^1)$ and vector $\varrho^2 = (\varrho_1^2, \dots, \varrho_N^2)$. The dynamic recovery schedule is defined as

$$\begin{aligned} &(\varrho^1(\gamma_1(t)), \dots, \varrho^1(\gamma_N(t))) \quad 0 \leq t \leq N \\ &(\varrho^2(1 - \gamma_1(t)), \dots, 1 - \varrho^2(\gamma_N(t))) \quad 0 \leq t \leq N. \end{aligned} \quad (16)$$

Then, Theorem 4 is introduced to define the expected repair cost of the repair operations of the network in a period.

Theorem 4: The expected amount of cost in the time horizon $[0, T]$ is

$$\begin{aligned} C(\mathbf{w}) = & \int_0^T \sum_{i=1}^N \phi_i(\delta_i(t)) S_i(t, \mathbf{w}(t), G(t)) dt \\ & + \int_0^T \sum_{i=1}^N \varrho_i^1(\gamma_i(t)) [1 - M_i(t, \mathbf{w}(t), G(t)) \\ & - S_i(t, \mathbf{w}(t), G(t)) - H_i(t, \mathbf{w}(t), G(t))] dt \\ & + \int_0^T \sum_{i=1}^N \varrho_i^2(1 - \gamma_i(t)) [1 - M_i(t, \mathbf{w}(t), G(t)) \\ & - S_i(t, \mathbf{w}(t), G(t)) - H_i(t, \mathbf{w}(t), G(t))] dt. \end{aligned} \quad (17)$$

Proof: The proof of the cost is comparable to the proof in Theorem 2, and we omit it here. ■

Remark 4: Through the quarantine cost function ϕ with recovery cost function ϱ^1 and ϱ^2 , we define the expected cost of the repair operations in the infinitesimal time of one node. Based on the infinitesimal cost of each node, it is easy to represent the expected cost of the repair operations in a time period by the integral formulation.

According to Theorems 2, 3, and 4, the total expected impact is

$$\begin{aligned} I(\mathbf{w}) = & L(\mathbf{w}) + E(\mathbf{w}) + C(\mathbf{w}) \\ = & \int_0^T I(\mathbf{S}(t, \mathbf{w}(t)), \mathbf{w}(t), G(t)) dt, \end{aligned} \quad (18)$$

where

$$\begin{aligned} I(\mathbf{E}(t, \mathbf{w}(t)), \mathbf{w}(t)) = & \int_0^T \sum_{i=1}^N [(\phi_i(\delta_i(t)) + a_i^2) S_i(t) + a_i^1 M_i(t)] dt \\ & + \int_0^T \sum_{i=1}^N [b_i + \varrho_i^1(\gamma_i(t)) + \varrho_i^2(1 - \gamma_i(t))] * \\ & [1 - M_i(t) - S_i(t) - H_i(t)] dt. \end{aligned} \quad (19)$$

IV. DYNAMIC APT REPAIR GAME MODEL AND PPAC PROBLEM FORMULATION

In this section, we first model the interactions between defenders and attackers as a dynamic APT repair game. Then, we formulate the dynamic APT repair game as a PPAC problem targeted to minimize the defender's potential loss.

A. Modelling the Dynamic APT Repair Game

In this subsection, we first analyze the subgame, i.e., the dynamic APT repair game in one time slot of the ATP attack process as below.

1) Subgame Modelling: In an APT attack process slot, defenders deploy defensive measures far-sighted to minimize the organization's loss. We model the behaviors of attackers and defenders in each time slot as a repair game as follows:

Game 1: (Repair Game in an APT Attack Slot).

- **Players:** each host $v_i \in A_j$ of the organization.
- **Strategies:** depending on each node's average rate $\delta_i \in \{\underline{\delta}_i, \bar{\delta}_i\}$, $\lambda_i \in \{\underline{\lambda}_i, \bar{\lambda}_i\}$, and $\gamma_i \in \{\underline{\gamma}_i, \bar{\gamma}_i\}$ defined in Eq. (11),

defenders deploy the different defense measures at each node v_i defined in Eq. (3) for organization loss reducing.

- *Objectives*: defenders aim to minimize the total impact of APT defined in Eq (18).

Remark 5: In this paper, we assume the attacker adopts the greedy strategy, i.e., maximizing the loss to the target organization in the current game. The reason behind the attacker's inclination towards a greedy strategy stems from the inherent information asymmetry, where the attacker lacks comprehensive access to the organization's global information, i.e., network topology, access control settings, asset distribution, etc. Limited to local information, such as the organization's network segment or specific service versions, the attacker strives to maximize immediate gains in each operation. According to [40], under the circumstances of information asymmetry, it is considered a judicious course of action for attackers to adopt a greedy strategy.

Each element of APT process sequence Eq. (1) denotes the APT attack time slot of the subgame, such as $A_i = (t_j, t_{j+1})$ represents i -th subgame and $[t_j, t_{j+1}]$ represents the time horizon of this subgame. Next, we model the defenders resource allocation behaviors as a sequential game in the APT attack process.

2) *Sequential Game Modelling*: Before modeling the sequential game, we introduce the interactions between the subgame. Defenders are often passive [41] in an APT defense process because defenders have no grounds to deny users access to the network services. The defenders' sensible choice is to respond quickly after observing the attack. Therefore in a subgame, the attacker first determines its attack strategy in Stage I, and then the defender determines its defense strategy in Stage II as a response to the observed behaviors of the attacker. These two stages repeat in the subgame, and the strategies of this subgame are known for the next subgame. The sequential game to model the interactions among attacks and defenders in the finite time horizon is as follows.

Game 2: (Sequential Game in APT Attack Process).

- *Players*: each host $v_i \in A_j$ of the organization.
- *Strategies*: average rate vector $w(t)$ defined in Eq. (11) in each time slot t .
- *Histories*: the action of previous subgame $w(t-1)$.
- *Objectives*: defenders aim to minimize the total impact of APT in the whole APT attack process.

Remark 6: In the above sequential game, defenders decide the defense strategy timely to minimize the loss of the entire APT attack process.

3) *Game Analysis*: As demonstrated in Eq. (8), the attacker can manipulate the selection of target hosts by controlling the parameters α and β , and can adjust the intensity of attacks on the target hosts by tuning the parameter λ . We define $(PC[0, T])^k$ as the set of all piecewise continuous k -dimensional functions defined on the interval $[0, T]$.

Define a (N) -dimensional function \mathbf{A}_t as follows:

$$\mathbf{A}_t = (\lambda_1(t), \dots, \lambda_N(t)) \quad 0 \leq t \leq N \quad (20)$$

Then the strategy of the attacker can be represented as $\mathbb{S}_A = \mathbf{A}_t \in (PC[0, T])^N$.

In contrast, defenders have the ability to optimize host maintenance strategies by controlling the parameters δ and γ . Define a $(N+N)$ -dimensional function \mathbf{D}_f as follows:

$$\mathbf{D}_f = (\delta_1(t), \dots, \delta_N(t), \dots, \gamma_1(t), \dots, \gamma_N(t)) \quad 0 \leq t \leq N. \quad (21)$$

The strategy of defender can be represented as $\mathbb{S}_D = \mathbf{D}_f \in (PC[0, T])^{N+N}$. After defining the strategy sets for the attacker and defender, we assume that the strategy spaces for both can be partitioned into discrete sets [42]. Let the strategy set for attackers have m options, denoted as $\mathbb{S}_A = \{\mathbf{A}_1^t, \dots, \mathbf{A}_m^t\}$. Similarly, the strategy set for defenders has n options, denoted as $\mathbb{S}_D = \{\mathbf{D}_1^f, \dots, \mathbf{D}_n^f\}$. The decisions of attackers and defenders form the strategy pairs $(\mathbf{A}_i^t, \mathbf{D}_j^f)$. Naturally, there are a total of $m \times n$ strategy pairs in this situation. Assuming that the winning strategy for the defender is denoted as \mathbf{A}_{ij}^t , we can obtain the matrix \mathbf{A} in Eq. (22).

$$\begin{bmatrix} \mathbf{A}_{11}^t & \mathbf{A}_{12}^t & \dots & \mathbf{A}_{1n}^t \\ \mathbf{A}_{21}^t & \mathbf{A}_{22}^t & \dots & \mathbf{A}_{2n}^t \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{m1}^t & \mathbf{A}_{m2}^t & \dots & \mathbf{A}_{mn}^t \end{bmatrix} \quad (22)$$

Based on the analysis of attack impact in Sect. III.C, the attacker's benefit is determined as follows:

$$\mathbf{A}_b = \mathbf{W}^A \cdot I(\mathbf{E}(t, w(t)), w(t)). \quad (23)$$

The defender's benefit is determined as

$$\mathbf{D}_b = \mathbf{W}^D \cdot I(\mathbf{E}(t, w(t)), w(t)). \quad (24)$$

The \mathbf{W}^A and \mathbf{W}^D represent the coefficients of impact of attacks and defenders, where $\mathbf{W}^A > 0$ and $\mathbf{W}^D < 0$. The impact of attacks is a positive gain for attackers, but a loss for defenders. It should be noted that the matrix \mathbf{A} also represents the loss matrix for the attacker, thus the payoff matrix for the attacker is $-\mathbf{A}$. In the process of the game, it is prudent to consider the opponent's motivation to maximize one's own loss (given the intense competition). Therefore, one should strive for the best possible outcome in the worst-case scenario. In mathematical terms, this corresponds to first taking the minimum of each row element (assuming the opponent minimizes one's loss), and then taking the maximum of all the minimum values across rows (resulting in the minimum loss), such as

$$I(\mathbf{E}(t, (\mathbb{S}_A^*, \mathbb{S}_D^*), (\mathbb{S}_A^*, \mathbb{S}_D^*))) = \max_i \min_j \mathbf{A}_{mn}^t. \quad (25)$$

In such a zero-sum game, a strategy pair $(\mathbb{S}_A^*, \mathbb{S}_D^*) \in \mathbf{A}_t \times \mathbf{D}_f$ is referred to as a Nash equilibrium if

$$I(\mathbf{E}(t, (\mathbb{S}_A^*, \mathbb{S}_D^*), (\mathbb{S}_A^*, \mathbb{S}_D^*))) \geq I(\mathbf{E}(t, (\mathbb{S}_A, \mathbb{S}_D), (\mathbb{S}_A, \mathbb{S}_D))). \quad (26)$$

Once the subgame equilibrium has been determined, we can then proceed to analyze the equilibrium of the entire sequential game. Considering the strategic interactions and decision-making of the players across all stages of the game, we aim to identify the strategies and outcomes that constitute an equilibrium for the overall sequential game. This entails examining the consistency and optimality of player strategies, taking into account the sequential nature of the game and the information available to each player at different stages.

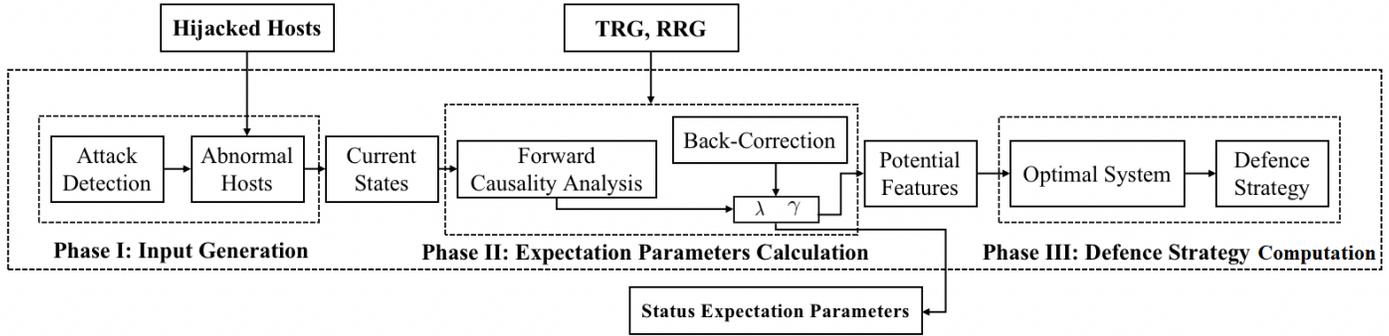


Fig. 3: Architecture of the proposed scheme

Theorem 5: Dynamic games in which players have a finite set of actions and engage in a finite number of interactions exhibit equivalence, while every finite extensive-form game characterized by perfect information possesses a pure-strategy Nash equilibrium [43].

Proof: The proof for Theorem 6 is provided in [44]. ■

Theorem 5 shows the existence of a Nash equilibrium in the formulated sequential game. Moreover, it guarantees the convergence of the game to this equilibrium within a finite number of iterations. Hence, the equilibrium of the game is $\mathbb{S} = ((\mathbb{S}_A^*, \mathbb{S}_D^*)^1, \dots, (\mathbb{S}_A^*, \mathbb{S}_D^*)^N)$.

B. PPAC Problem Formulation

In this subsection, we formulate the PPAC Problem based on the game model. Firstly, we assume the solution function of PPAC is piecewise continuous in this paper. Hence, we define $PC[0, T]$ as the function space of the attack process, and the PPAC's solution set is $\mathbf{w} \in (PC[0, T])^{3N}$. Secondly, we assume all quarantine rate functions, all threat rate functions, and all recovery rate functions in each feasible strategy are bounded by the dynamic defense resource. Thereby, for $1 \leq i \leq N$, $\underline{\delta}_i, \bar{\delta}_i$ represent upper and lower bounds of δ_i , $\underline{\lambda}_i, \bar{\lambda}_i$ represent upper and lower bounds of λ_i , and $\underline{\gamma}_i, \bar{\gamma}_i$ represent upper and lower bounds of γ_i . For brevity, we denote $\underline{\delta} = (\underline{\delta}_1, \dots, \underline{\delta}_N)$, $\bar{\delta} = (\bar{\delta}_1, \dots, \bar{\delta}_N)$, $\underline{\lambda} = (\underline{\lambda}_1, \dots, \underline{\lambda}_N)$, $\bar{\lambda} = (\bar{\lambda}_1, \dots, \bar{\lambda}_N)$, $\underline{\gamma} = (\underline{\gamma}_1, \dots, \underline{\gamma}_N)$, $\bar{\gamma} = (\bar{\gamma}_1, \dots, \bar{\gamma}_N)$. Note that the value of $\underline{\delta}, \bar{\delta}, \underline{\lambda}, \bar{\lambda}, \underline{\gamma}$, and $\bar{\gamma}$ can be estimated by the previous data and the experience of defenders for different scenarios. Finally, the feasible solution set of the PPAC problem can be summarized as

$$\mathcal{W} = \{\mathbf{w} \in (PC[0, T])^{3N} | \mathbf{w}(t) \in \prod_{i=1}^N [\underline{\delta}_i, \bar{\delta}_i] \times \prod_{i=1}^N [\underline{\lambda}_i, \bar{\lambda}_i] \times \prod_{i=1}^N [\underline{\gamma}_i, \bar{\gamma}_i]\}. \quad (27)$$

After defining the above basic variables, we proceed with formulating the PPAC problem. To better align the problem with real-world attack-defense scenarios, our work is based on three practical assumptions: (a) We assume that the entire attack-defense process can be divided into multiple sub-stages based on time [45], and the defender has access to the current stage's global network information and all the information from the previous round, while the attacker can only access the information from

the previous round. (b) In each round of the process, the attacker takes action first, and due to the unequal information between the attacker and defender, the attacker adopts a greedy strategy (further details can be found in Remark 5), pursuing maximum immediate gains without considering the impact on the next stage. On the other hand, the defender leverages their advantage of having more information and considers the impact of their belief about the attacker's decision in the next round [46]. (c) At time period $t + 1$, the defender's relative belief is solely determined by the historical information and the defender's action at time period t [47].

Define vector $\mu = (\mu^1, \dots, \mu^m)^T$ as the belief vector of defenders, and vector $\rho = (\rho^1, \dots, \rho^m)^T$ as the belief vector of attackers. Then the expectation of defenders win is $\mathbb{E}(\mathbb{S}_A, \mathbb{S}_D) = \mu^T \mathbf{A} \rho$. Then, the objective from defender's perspective can be expressed as

$$\max_{\mu} \min_{\rho} \mu^T \mathbf{A} \left(\sum_{j=1}^n \rho_j e_j \right), \quad (28)$$

where e_j represents a vector with only the j -th component being 1 and all other components being 0. Therefore, from the defender's perspective and the aforementioned assumptions, we can formulate the optimization problem for the defenders as

$$\min I(\mathbf{w}) = \max_{\mu} \left(\sum_{j=1}^n \rho_j e_j \right). \quad (29)$$

The optimization problem for the attackers is

$$\max -I(\mathbf{w}) = \min_{\rho} \left(\sum_{i=1}^n \mu_i e_i \right). \quad (30)$$

Therefore, the decision problems of the attacker and defender are dual linear problems and have the same optimal objective function value. Hence, finding the optimal solution for either of them would suffice. To capture the influence of attacker's decisions on defender's decisions, we enhance the optimization problem for the defender by incorporating the factors of attacker's decisions into defender's objective function. Therefore, for the

PPAC problem, a decision can be represented as $\mathbf{w}(t)$. Hence, the PPAC problem can be formulated as

$$\min_{\mathbf{w} \in \mathcal{W}} I(\mathbf{w}) = \int_0^T I(\mathbf{E}(t, \mathbf{w}(t)), \mathbf{w}(t), G(t)) dt, \quad (31)$$

$$s.t. \begin{cases} \frac{d\mathbf{E}(t)}{dt} = \mathbf{F}(\mathbf{E}(t), \mathbf{w}(t)) \\ \lambda_i(t) = T(v_i, t) \\ \gamma_i(t) = R(v_i, t) \\ \mathbf{E}(0) = \mathbf{E}_0 \\ \mathbf{U}_n \in (0, U) \end{cases}$$

where $\frac{d\mathbf{E}(t)}{dt} = \mathbf{F}(\mathbf{E}(t), \mathbf{w}(t))$ is a brief expression of Eq. (11). $T(v_i, t)$ and $R(v_i, t)$ represent the node v_i in the threat algorithm TRG and recovery algorithm RRG, respectively. \mathbf{U} is the utility of the network work without APT attack, and \mathbf{U}_n is the minimum utility of the task need.

Remark 7: The admissible control set of the optimal problem in Eq. (31) represents the feasible defense strategies of the dynamic APT repair game. The objective function means the APT impact of the organization under a workable sequence, and each optimal solution denotes potential PPAC defense strategies that minimize the expected APT impact.

The PPAC problem can be summarized as

$$\mathbb{M} = (G, a_i^1, a_i^2, b_i, \phi, \varrho^1, \varrho^2, \alpha, \beta, \delta, \lambda, \gamma, \hat{T}, R, T, \mathbf{U}_n, \mathbf{U}, \mathbf{E}_0), \quad (32)$$

where λ and γ are parameters determined by algorithm TRG and algorithm RRG, respectively.

V. SOLUTION TO THE PPAC PROBLEM

A. Solution Overview

Fig. 3 shows the workflow of our proposed game solution to design optimal defense strategies under an APT alert (i.e., the current anomalous hosts). As previously analyzed, the dynamic variation of the network and the impact of attack and defense interactions make the solution strategy easy to fall into a locally optimal solution. To escape the local optimal solution trap, we develop two backtracking-forward rating algorithms for risk assessment in phase II. For timely defense, we build an optimal system in phase III. The detail of the three phases in our solution is shown below.

- *Phase I:* In this phase, the attacker obtains the anomalous hosts in the current network based on the attack detection methods [39], [48]–[51] and uses these anomalous hosts as input for the next phase.
- *Phase II:* Depending on phase I, the TRG and RRG compute the status expectation parameters λ and γ , respectively. The details are shown in Section V-B.
- *Phase III:* We construct a dynamic system based on the solution objective of the PPAC problem. To simplify the solving, we extend the solving set to segmented functions through Pontryagin maximum principle. Based on the above and the parameters of phase II, phase III can give the optimal strategies to solve the PPAC problem. The detail of phase III is shown in Section V-C.

B. Backtracking-Forward Rating Algorithm

The time-varying network topology leads to the optimal behavior in the current network state becoming a locally optimal solution in the global aspect. Although this problem can be solved by increasing the dynamic inputs to the optimal system, the system with various dynamic inputs will make the interactions between inputs complex. To solve this problem, we develop two backtracking-forward rating algorithms, TRG and RRG. The three key factors and the corresponding algorithm operations for the classification of a node are described as follows.

- **Structural features:** The importance of the topological relationship between nodes in the network. Some nodes in the network are critical to its integrity due to their relative position, degree, and the betweenness [52]. To represent the topological importance of a node on the network, we use the k-shell [53] algorithm to characterize the range of node's influence on the network and use the betweenness to characterize the influence of a node on the connectivity of the network.
- **Lateral features, which is also called infection features:** The node's importance to the state matrix's spectral radius after a lateral attack on the entire network. For example, if a service host in an organization with an open public port is attacked, the attacker can use the machine as a broiler to attack devices that require a specific IP to open the port. Our algorithms use epidemic thresholds [54] for node lateral attack descriptions. We use the spectral radius of the entire state matrix as described in the [29] to discuss epidemic thresholds.
- **Service features:** The importance of nodes to the service performance of the entire network. The impact of the node itself on the overall service function is determined by two main aspects: its original service capability and its role in the network for information transfer. Service utility information is readily available to defenders. In this paper, we take in the experiment according to the topology of randomly generated service weights to represent his service capabilities.

Note that, the spectral radius in this paper is $\rho(\mathbf{G}) = \max_{1 \leq i \leq n} |\lambda_i|$, where $\rho(\Lambda A - M)$ indicates the epidemic threshold. In $\rho(\Lambda A - M)$, Λ represents the infection rate diagonal matrix, M represents the recovery rate diagonal matrix, and A is the adjacency matrix of the network. The utility of the overall network is $Z = N \times \bar{I}$, where the \bar{I} represents the average utility of each node, and N is the number of the nodes. In what follows, we illustrate the TRG algorithm and RRG algorithm in detail, respectively.

The computational complexity of Algorithm 1 is $O(n \cdot N^2)$, where n and N represent the number of nodes and the number of periods, respectively. In line 6 of Algorithm 1, n nodes are performing utility calculations. This calculation involves a nested loop, where the number of iterations is $n \cdot N$. In lines 12–18 of algorithm 1, we can observe that in the worst case, the entire nested loop will backtrack to the initial position after each iteration. Thus, the number of worst iterations in this loop is $\frac{N(1+N)}{2}$. The overall number of iterations is $n \cdot \frac{N(1+N)}{2}$.

1) *The TRG Algorithm:* The TRG algorithm is used to classify the hazard level. The dynamic contagion problem is summarized in [29], which is proved to be NP-hard under the edge-level

Algorithm 1 TRG

Input: $t, \mathbf{E}(t), \hat{\mathbf{T}}, \mathbf{U}, \mathbf{U}_n, \bar{I}, k, t_w$
Output: the threaten state list M

- 1: $N = |\hat{\mathbf{T}}|$, which is the elements number of $\hat{\mathbf{T}}$;
- 2: $G = \hat{\mathbf{T}}(t) = (V, E)$, $V = (v_1, \dots, v_n)$;
- 3: **for** $1 \leq i \leq n$ **do**
- 4: P_{v_i} : Compromise rate of neighbor nodes within one hop of aggregation node v_i by Eq. (8) and t_w ;
- 5: **end for**
- 6: **for** $P_{v_i} \in \{P_{v_1}, \dots, P_{v_n}\}$ **do**
- 7: **while** $k \leq N$ **do**
- 8: $G = \hat{\mathbf{T}}(k)$
- 9: $P_{v_i} = P_{v_i} + \rho_\phi \rho_\epsilon + Z_\phi Z_\epsilon$;
- 10: Compute $Z^{(k)}$, which is the overall utility of the entire network after removing node v_i from the network;
- 11: Compute $\rho^{(k)}$, which is the spectral radius at moment t after removing node v_i from the network;
- 12: **if** $[\rho_\phi(\rho^{(k)} - \rho^{(k-1)}) - Z_\phi(Z^{(k)} - Z^{(k-1)})] > 0$ **then**
- 13: $k + +$;
- 14: **else**
- 15: $\rho_\epsilon = \rho_\epsilon + \rho^{(k)} - \rho^{(k-1)}$;
- 16: $Z_\epsilon = Z_\epsilon + Z^{(k)} - Z^{(k-1)}$;
- 17: $k - -$;
- 18: **end if**
- 19: **end while**
- 20: **end for**
- 21: sorting V in descending order;
- 22: $j = \frac{(\bar{U} - U_n)}{\bar{I}}$;
- 23: $P = P_{v_1} + \dots + P_{v_j}$;
- 24: **for** $1 \leq i \leq n$ **do**
- 25: $M_i = \frac{P_{v_i}}{P}$;
- 26: **end for**
- 27: return(T).

model. TRG adopts the node-level method to solve the dynamic contagion problem.

The incoming time t and sequence $\hat{\mathbf{T}}$ are used to determine the current position k in the time. Then, we adopt the k-shell algorithm to get each node's threat weight k_s , which can help us better quantify each node's importance in the network. The k-shell algorithm obtains the k_s coefficient of each node, and the k_s coefficient is defined as the threat weight of each node. We denote \bar{I} representing each host's average service performance.

The algorithm starts by ranking the utility of the systems with initializing the input sequence lines 1-3. Next, we depend on the Eq. (8) and the threat weight to aggregate the compromise rate of neighbors within one hop of each node which we define as p_{v_i} , lines 4. Then, we take each node with its p_{v_i} to determine the threat factor. We measure nodes' contagion by the epidemic threshold of the network after removing the node and its utility by the change in the service utility of the network after removing the node, lines 4-7. Then we rely on the topological relation transformations throughout the task as well as the variation of the spectral radius to give two coefficients ρ_ϕ and Z_ϕ to measure the impact. Furthermore, the algorithm describes the impact of the current judgment on the next epoch by ρ_ϵ and Z_ϵ , lines 8-13. The primary objective is to ascertain the adherence of the preceding period's decision to the prescribed criteria. Suppose we find an increase in contagiousness or a decrease in utility. In that case, we backtrack to the previous epoch with this information until a node has completed all epochs to get its final value p_{v_i} , lines 14-20.

These values are propagated and assimilated into the network model, facilitating the evaluation and validation of the updated network topology during this period. The primary objective is to ascertain the adherence of the preceding period's decision to the prescribed criteria. If the decision is deemed compliant, the progression continues to the next period. However, if discrepancies are detected, and the decision fails to meet the stipulated requirements, the disparity value ρ_ϵ and Z_ϵ are utilized as an adjacent solution to rectify the decision made in the previous stage.

To ensure the credibility of the rating, we make a worst-case assumption about the evolution of the node and find the potentially infected node with the largest structural and efficacy characteristics as the default infected node. Finally, we sort the nodes by these final values and get the threat rate by the ratio of the theoretical values. The threat rate is generated by comparing the theoretical worst total impact value at that moment with the theoretical value for each specific node, lines 20-24. The algorithm finally outputs a list of threat registration assessments M for all nodes at that moment, line 24. The details of RRG algorithm are shown in Algorithm 1.

Algorithm 2 RRG

Input: $t, \mathbf{E}(t), \hat{\mathbf{T}}, \mathbf{U}, \mathbf{U}_n, \bar{I}, k, t_w$
Output: the recovery state list M

- 1: $N = |\hat{\mathbf{T}}|$, which is the elements number of $\hat{\mathbf{T}}$;
- 2: $G = \hat{\mathbf{T}}(t) = (V, E)$
- 3: **for** $1 \leq i \leq n$ **do**
- 4: P_{v_i} : Compromise rate of neighbor nodes within one hop of aggregation node v_i by Eq. (8) and t_w ;
- 5: **end for**
- 6: **for** $P_{v_i} \in \{P_{v_1}, \dots, P_{v_n}\}$ **do**
- 7: **while** $k \leq N$ **do**
- 8: $G = \hat{\mathbf{T}}(k)$
- 9: $P_{v_i} = P_{v_i} + \rho_\phi \rho_\epsilon + Z_\phi Z_\epsilon$;
- 10: Compute $Z^{(k)}$, which is the overall utility of the entire network after adding node v_i to the network;
- 11: Compute $\rho^{(k)}$, which is the spectral radius at moment t after adding node v_i to the network;
- 12: **if** $(Z_\phi(Z^{(k)} - Z^{(k-1)}) - \rho_\phi(\rho^{(k)} - \rho^{(k-1)})) > 0$ **then**
- 13: $k + +$;
- 14: **else**
- 15: $\rho_\epsilon = \rho_\epsilon + \rho^{(k)} - \rho^{(k-1)}$;
- 16: $Z_\epsilon = Z_\epsilon + Z^{(k)} - Z^{(k-1)}$;
- 17: $k - -$;
- 18: **end if**
- 19: **end while**
- 20: **end for**
- 21: sorting V in descending order;
- 22: $P = P_{v_1} + \dots + P_{v_n}$;
- 23: **for** $1 \leq i \leq n$ **do**
- 24: $M_i = \frac{P_{v_i}}{P}$;
- 25: **end for**
- 26: return(R).

2) *The RRG Algorithm:* Due to the RRG algorithm having the same principle as TRG, we mainly introduce the difference between them. TRG and RRG are different in inputs; the inputs of RRG are the *quarantined* nodes. RRG considers the contagion and efficiency impact of rejoining the network for the nodes separated from the network in the quarantine zone. Therefore, the RRG algorithm's back-tracing is determined by whether the effectiveness expectation is greater or smaller than the contagion

$$\begin{aligned} \mathbf{H}(\mathbf{E}, \mathbf{w}, \boldsymbol{\kappa}, \boldsymbol{\rho}, \boldsymbol{\xi}) &= \sum_{i=1}^N \kappa_i \left[\gamma_i(t) [1 - M_i(t) - S_i(t) - H_i(t)] - H_i(t) \left(\beta_i(t) + \alpha_i \sum_{j=1}^n a_{ij}(t) M_j(t) + \alpha_i \sum_{j=1}^n a_{ij}(t) S_j(t) \right) \right] \\ &+ I(\mathbf{E}(t), \mathbf{w}(t)) + \sum_{i=1}^N \xi_i (\lambda_i(t) M_i(t) - \delta_i(t) S_i(t)) + \sum_{i=1}^N \rho_i \left[H_i(t) \left(\beta_i(t) + \alpha_i \sum_{j=1}^n a_{ij}(t) M_j(t) + \alpha_i \sum_{j=1}^n a_{ij}(t) S_j(t) \right) - \lambda_i(t) M_i(t) \right] \end{aligned} \quad (33)$$

expectation after the node is joined. The detail of RRG is shown in Algorithm 2.

The computational complexity of Algorithm 2 is $O(n \cdot N^2)$, where n and N represent the number of nodes and the number of periods, respectively. The analysis procedure follows a similar methodology as described in Algorithm 1, and for the sake of conciseness, we omit its detailed exposition in this section.

C. Optimal Control-based Defensive Strategy-making

For the accuracy and feasibility of the defense strategy, we denote the Eq. (32) as a 3_{rd} order dynamic system [55]. In this system, the state transition parameters of the lateral movement attack model are used as the inputs.

Following the optimal theory [55], the Hamiltonian of PPAC problem Eq. (31) is summarized as Eq. (33), where $\boldsymbol{\kappa} = (\kappa_1, \dots, \kappa_N)$, $\boldsymbol{\rho} = (\rho_1, \dots, \rho_N)$ and $\boldsymbol{\xi} = (\xi_1, \dots, \xi_N)$ are the associated adjoints. Then, the conditions of optimal control are proven in Theorem 6.

Theorem 6: Assume that \mathbf{w} is an optimal control of the PPAC problem, and \mathbf{E} is the solution to the associated system (Eq. (11)). Then the adjoint functions are $\boldsymbol{\kappa}$, $\boldsymbol{\rho}$ and $\boldsymbol{\xi}$ when $\boldsymbol{\kappa}(T) = \boldsymbol{\rho}(T) = \boldsymbol{\xi}(T) = 0$. Furthermore,

$$\begin{cases} \delta_i(t) \in \arg \min_{\delta \in [\underline{\delta}_i, \bar{\delta}_i]} S_i(t) [\phi_i(\delta) - \xi_i(t) \delta] \\ \lambda_i(t) \in \arg \min_{\lambda \in [\underline{\lambda}_i, \bar{\lambda}_i]} M_i(t) [\xi_i(t) \lambda - \rho_i(t) \lambda] \\ \gamma_i(t) \in \arg \min_{\gamma \in [\underline{\gamma}_i, \bar{\gamma}_i]} [1 - H_i(t) - M_i(t) - S_i(t)] \\ \quad [\varrho_i^1(\gamma) + \varrho_i^2(1 - \gamma) + \kappa_i(t) \gamma] \\ 0 \leq t \leq T, i = 1, \dots, N. \end{cases} \quad (35)$$

Proof: According to the Pontryagin's minimum principle, the differential equations of $\boldsymbol{\kappa}$, $\boldsymbol{\rho}$ and $\boldsymbol{\xi}$ are illustrated as

$$\begin{cases} \frac{d\kappa_i(t)}{dt} = -\frac{\partial H(\mathbf{E}(t), \mathbf{w}(t), \boldsymbol{\kappa}(t), \boldsymbol{\rho}(t), \boldsymbol{\xi}(t))}{\partial \kappa_i} \\ \frac{d\rho_i(t)}{dt} = -\frac{\partial H(\mathbf{E}(t), \mathbf{w}(t), \boldsymbol{\kappa}(t), \boldsymbol{\rho}(t), \boldsymbol{\xi}(t))}{\partial \rho_i} \\ \frac{d\xi_i(t)}{dt} = -\frac{\partial H(\mathbf{E}(t), \mathbf{w}(t), \boldsymbol{\kappa}(t), \boldsymbol{\rho}(t), \boldsymbol{\xi}(t))}{\partial \xi_i} \\ 0 \leq t \leq T, i = 1, \dots, N. \end{cases} \quad (36)$$

Theorem 6 is proved. ■

$$\begin{aligned} \frac{d\kappa(t)}{dt} &= b_i + (\varrho_i^1 - \varrho_i^2 + \kappa_i(t)) \gamma_i + (\kappa_i(t) - \rho_i(t)) \left(\beta_i + \alpha_i \sum_{j=1}^n a_{ij}(t) M_j(t) + \alpha_i \sum_{j=1}^n a_{ij}(t) S_j(t) \right) \\ \frac{d\rho(t)}{dt} &= -a_i^1 + b_i + \varrho_i^2 + (\varrho_i^1 - \varrho_i^2 + \kappa_i(t)) \gamma_i(t) + (\rho_i(t) - \xi_i(t)) \lambda_i + [\rho_i(t) - \kappa_i(t)] \sum_{j=1}^N \alpha_i a_{ij} H_j(t) \\ \frac{d\xi(t)}{dt} &= -a_i^2 - \phi_i(\delta_i(t)) + b_i + \varrho_i^2 + (\varrho_i^1 - \varrho_i^2 + \kappa_i(t)) \gamma_i + \xi_i(t) \delta_i(t) + [\rho_i(t) - \kappa_i(t)] \sum_{j=1}^N \alpha_i a_{ij} H_j(t). \end{aligned} \quad (34)$$

Due to the terminal cost and final state are unlimited, the conditions $\boldsymbol{\kappa}(T) = \boldsymbol{\rho}(T) = \boldsymbol{\xi}(T) = 0$ hold. Again, by Pontryagin's minimum principle, we have

$$\mathbf{w}(t) \in \arg \min_{\tilde{\mathbf{w}} \in \mathcal{W}} H(\mathbf{E}(t), \tilde{\mathbf{w}}(t), \boldsymbol{\kappa}(t), \boldsymbol{\rho}(t), \boldsymbol{\xi}(t)), 0 \leq t \leq T. \quad (37)$$

Combining Eq. (33), Eq. (35), and Eq. (37), we can deduce the Eq. (38).

According to optimal control theory, we can get the control of the PPAC problem Eq. (31) by solving Eq. (11), (34), (35), (38), and condition $\boldsymbol{\kappa}(T) = \boldsymbol{\rho}(T) = \boldsymbol{\xi}(T) = 0$. We point out that the solutions of this system are potential optimal strategies of the PPAC problem. The analysis of the solutions' structure is detailed in Appendix B.

VI. PERFORMANCE EVALUATION

We adopt the following three experimental settings. *Setting 1:* a static network (i.e., in Setting VI-A); *Setting 2:* a periodic time-varying network (i.e., in Setting VI-B); *Setting 3:* a time-varying network (i.e., in Setting VI-C). To be fair in performance comparison, we adapt well-performing techniques ER [33] (differential private game) and QAR [6] (quarantine and recovery) from static networks. We compute the results separately for each time interval and averaged them to obtain the solutions for these methods. This allows us to adapt the methods to the dynamic nature of the network and provide meaningful comparisons.

ER performs overall repairs without node isolation, while QAR isolates all nodes with anomalies. To ensure fairness, we compared the number of repair operations over a specific time period using the PPAC metric. In the experiments, for ER, we set the number of iterations equal to the number of nodes in the network. For QAR, we also set the number of time steps in the forward-backward method [6], [28] equal to the number of nodes in the network. As such, the computational complexity of ER is $O(N \cdot n^2)$, where n and N represent the number of nodes and the number of periods. The computational complexity of QAR is also $O(N \cdot n^2)$, where n and N represent the number of nodes and the number of periods, respectively. For detailed settings of N and n in each experiment, please refer to the following subsections.

In general, the value of N is significantly smaller than n .

$$\begin{aligned}
 H(\mathbf{E}, \tilde{\mathbf{w}}(t), \boldsymbol{\kappa}, \boldsymbol{\rho}, \boldsymbol{\xi}) &= \int_0^T \sum_{i=1}^N [(\phi_i(\tilde{\delta}_i(t)) + a_i^2)S_i(t) + a_i^1 M_i(t)] dt \int_0^T \sum_{i=1}^N [b_i + \varrho_i^1(\tilde{\gamma}_i(t)) + \varrho_i^2(1 - \tilde{\gamma}_i(t))[1 - M_i(t) - S_i(t) - H_i(t)] dt \\
 &+ \sum_{i=1}^N \xi_i (\tilde{\lambda}_i(t)M_i(t) - \tilde{\delta}_i(t)S_i(t)) + \sum_{i=1}^N \kappa_i \left[\tilde{\gamma}_i(t)[1 - M_i(t) - S_i(t) - H_i(t)] - H_i(t) \left(\beta_i(t) + \alpha_i \sum_{j=1}^n a_{ij}(t)M_j(t) + \alpha_i \sum_{j=1}^n a_{ij}(t)S_j(t) \right) \right] \\
 &+ \sum_{i=1}^N \rho_i \left[H_i(t) \left(\beta_i(t) + \alpha_i \sum_{j=1}^n a_{ij}(t)M_j(t) + \alpha_i \sum_{j=1}^n a_{ij}(t)S_j(t) \right) - \tilde{\lambda}_i(t)M_i(t) \right]
 \end{aligned} \tag{38}$$

We evaluate the performance of our proposed optimal strategy in terms of the following three important metrics.

1) Resource occupancy: The resource occupancy is the ratio between the resource consumption and the overall resource (i.e., the CPU, memory, human resource, etc., which we regard as equivalent to U) at each moment. In this paper, we assume that lower resource occupancy corresponds to better method performance. The resource consumption of each method comprises the repair resource consumption C_r and the quarantine resource consumption C_q . Furthermore, the repair resource consumption C_r correlates with the number of repair nodes N_r , i.e., $C_r \propto N_r$, and the quarantine resource consumption C_q correlates with the number of repair nodes N_q , i.e., $C_q \propto N_q$. Hence, resource occupancy is $\frac{W_c^r \times \sum_{i=1}^T C_r + W_c^q \times \sum_{i=1}^T C_q}{W_c \times \frac{1}{10} \sum_{i=1}^T U}$, where the W_c^r , W_c^q , and W_c is the consumption correction factor.

Different organizations often allocate varying degrees of importance to the four categories of resources: human resources, node-specific utility, network resources, and storage resources. In our experimental setup, we aim to enhance the generalizability of our results by assuming equal weights for these resource categories. For convenience, we assume the consumption correction factor of repair, quarantine, and overall resource is $W_c^r : W_c^q : W_c = 1 : 2 : 4$.²

2) Defense resource utilization: For convenience, we assume the resource beyond the minimum are used for defense resources, i.e., $U - U_n$, and we assume that the defender expends all defensive resources to defend against the attack without strategy guidance. Hence, defense resource utilization is the ratio between the defensive resource and resource consumption, i.e., $1 - \frac{W_c \times \frac{1}{10} \sum_{i=1}^T (U - U_n)}{W_c^r \times \sum_{i=1}^T C_r + W_c^q \times \sum_{i=1}^T C_q}$, where $W_c \times \frac{1}{10} \sum_{i=1}^T (U - U_n)$ represents the defense resource of the network.

3) Service stability: The service stability is measured by N_s , the number of time periods when the organization's minimum service requirements are maintained, i.e., $\frac{W_c \times \frac{1}{10} \sum_{i=1}^T N_s}{T}$.

The expression for solving the problem with the mechanism is formulated as

$$\mathbb{M} = (G, \mathbf{a}_i^1, \mathbf{a}_i^2, \mathbf{b}_i, \phi, \boldsymbol{\varrho}^1, \boldsymbol{\varrho}^2, \boldsymbol{\alpha}, \boldsymbol{\beta}, \underline{\boldsymbol{\delta}}, \bar{\boldsymbol{\delta}}, \underline{\boldsymbol{\lambda}}, \bar{\boldsymbol{\lambda}}, \underline{\boldsymbol{\gamma}}, \bar{\boldsymbol{\gamma}}, \hat{T}, R, T, \mathbf{U}_n, \mathbf{U}, \mathbf{E}_0). \tag{39}$$

²Regarding resource-intensive operations, we consider that repair operations only consume human resources. For quarantine operations, both human resources and utility of the nodes are taken into account. However, no additional expenditures are required for communication and storage resources. It is worth noting that the overall resource allocation encompasses these four kinds of resources. To underscore the equitable representation of resource proportions, we introduce a coefficient that serves as a normalization factor.

Based on \mathbb{M} , we set the experimental parameters using the following steps.

Step 1: Set \mathbf{a}_i^{1*} , \mathbf{a}_i^{2*} , and \mathbf{b}_i . Define \mathbf{a}_i^{1*} , \mathbf{a}_i^{2*} , and \mathbf{b}_i^* as the vectors chosen randomly and uniformly from $(0, 1]^{100}$. In the following experiments, except in special cases, $\mathbf{a}_i^1 = \mathbf{a}_i^{1*}$, $\mathbf{a}_i^2 = \mathbf{a}_i^{2*}$, and $\mathbf{b}_i = \mathbf{b}_i^*$.

Step 2: Set $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$. Define $\boldsymbol{\alpha}^* = (0.1, \dots, 0.1)$, $\boldsymbol{\beta}^* = (0.1, \dots, 0.1)$. In our experiments, unless otherwise specified, $\boldsymbol{\alpha} = \boldsymbol{\alpha}^*$, $\boldsymbol{\beta} = \boldsymbol{\beta}^*$.

Step 3: Set $\underline{\boldsymbol{\delta}}$, $\bar{\boldsymbol{\delta}}$, $\underline{\boldsymbol{\lambda}}$, $\bar{\boldsymbol{\lambda}}$, $\underline{\boldsymbol{\gamma}}$, and $\bar{\boldsymbol{\gamma}}$. Define

$$\begin{aligned}
 \underline{\boldsymbol{\delta}}^c &= (0.c, \dots, 0.c), & \bar{\boldsymbol{\delta}}^c &= (0.c, \dots, 0.c) \\
 \underline{\boldsymbol{\lambda}}^c &= (0.c, \dots, 0.c), & \bar{\boldsymbol{\lambda}}^c &= (0.c, \dots, 0.c) \\
 \underline{\boldsymbol{\gamma}}^c &= (0.c, \dots, 0.c), & \bar{\boldsymbol{\gamma}}^c &= (0.c, \dots, 0.c)
 \end{aligned}$$

where $c \in \{1, 2, 3, 4, 5, 6\}$.

Step 4: Define the initial expected state \mathbf{E}_0 . Let

$$\mathbf{E}^* = (0.8, \dots, 0.8, 0.1, \dots, 0.1, 0.1, \dots, 0.1),$$

represent the average rate of the initial state of the node. In the following experiments, we set $\mathbf{E}_0 = \mathbf{E}^*$.

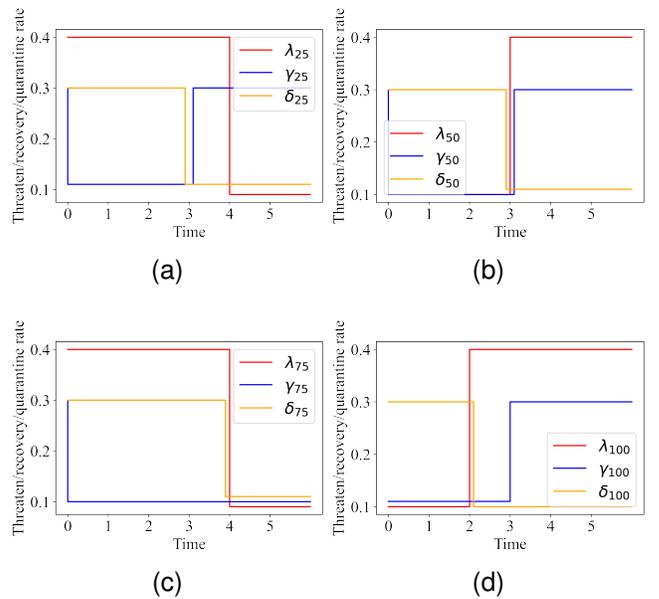


Fig. 4: A sketch of the PPAC strategy in setting 1.

A. Numerical Results of Static Networks (Setting 1)

Using the Pajek software [56], we simulate a static network G_{ST} with $N = 100$ nodes, the edge-rewiring probability is 0.1

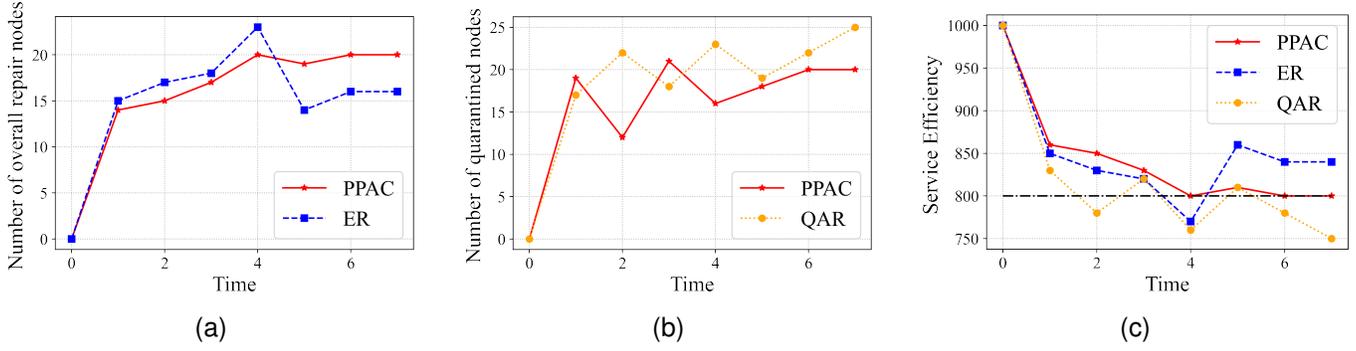


Fig. 5: (a) the comparison between our method and the ER by the number of nodes for overall maintenance at each moment in setting 1. (b) the comparison between our method and the QAR by the number of quarantined nodes at each moment in setting 1. (c) the comparison between our method and ER, QAR by the service quality at each time.

and maintain network connectivity. The $A_{ST}(t)$ represents the adjacency matrix of G_{ST} at time t and \hat{T} represents the graph sequence of G_{ST} .

Consider the APT repair game in static Networks:

$$\mathbb{M} = (G_{ST}, \mathbf{a}_i^{1*}, \mathbf{a}_i^{2*}, \mathbf{b}_i^*, \phi, \varrho^1, \varrho^2, \alpha, \beta, \delta^1, \delta^4, \lambda^1, \lambda^6, \gamma^1, \gamma^5, \hat{T}, R, T, U_n, U, E_0),$$

where $G = G_{ST}$, $T = 12$, and $\hat{T} = \{A_1 = (0, 2), A_1 = (2, 3), A_1 = (3, 4), A_1 = (4, 6)\}$. We set the following parameters under **Setting 1**:

- 1) The utility of the entire network $U = 1000$.
- 2) Maintain minimum service effectiveness requirements for the mission $U_n = 80\%U$.
- 3) Threaten rate list and recovery rate list are obtained by executing the TRG and RRG algorithms.
- 4) The specific representation of each function is

$$\phi(\delta) = (\sqrt{\delta}, \dots, \sqrt{\delta}), \varrho^1(\gamma) = (\sqrt{\gamma}, \dots, \sqrt{\gamma}), \varrho^2(1 - \gamma) = (\sqrt{1 - \gamma}, \dots, \sqrt{1 - \gamma}).$$

Executing the solving mechanism, we get the potential defense strategy. Fig. 4 (a)-(d) plot four threaten, quarantine, and recovery rate functions in the strategy. Fig. 5 (a) represents the number of repair nodes at each moment in setting 1. We can see that the difference between the PPAC and ER methods for repair is not obvious under a fixed network, which is also in line with our initial assumptions. Because without considering subsequent topology changes, our method still requires decisions on quarantine, repair, and local repair based on effectiveness and cost. Fig. 5 (b) represents the number of quarantined nodes at each moment in setting 1. The difference between PPAC and QAR in quarantined nodes is also insignificant. Still, the difference exists because our restrictions require that the number of nodes in the isolation zone can not exceed the threshold. Fig. 5 (c) represents the service quality at each moment in setting 1.

In summary, the resource occupancy of PPAC, QAR, and ER are 13.43%, 15.21%, and 13.5%, respectively. The defense resource utilization of PPAC rises 48.94%, QAR rises 31.46%, and ER rises 48.15%. The service stability of each method is as follows: PPAC 100%, QAR 42.85%, and ER 85.71%.

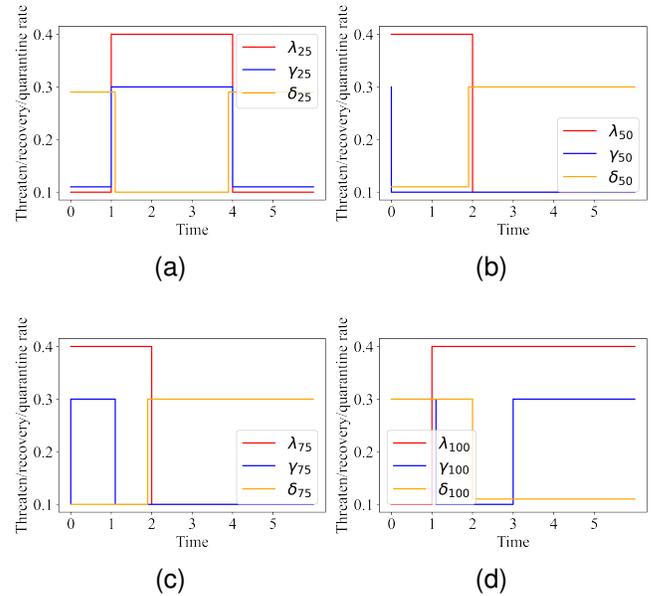


Fig. 6: A sketch of the PPAC strategy in setting 2.

B. Numerical Results of Periodic Time-varying Network (Setting 2)

Using the Pajek software [56], we simulate a periodic time-varying network G_{PT} with $N = 100$ nodes, the edge-rewiring probability is 0.1 and maintain network connectivity. The $A_{PT}(t)$ represents the adjacency matrix of G_{PT} at time t and \hat{T} represents the graph sequence of G_{PT} .

Consider the APT repair game in periodic time-varying Network:

$$\mathbb{M} = (G_{PT}, \mathbf{a}_i^{1*}, \mathbf{a}_i^{2*}, \mathbf{b}_i^*, \phi, \varrho^1, \varrho^2, \alpha, \beta, \delta^1, \delta^4, \lambda^1, \lambda^6, \gamma^1, \gamma^5, \hat{T}, R, T, U_n, U, E_0),$$

where $G = G_{PT}$, $T = 6$, and $\hat{T} = \{A_1 = (0, 2), A_2 = (2, 3), A_3 = (3, 4), A_4 = (4, 6)\}$. We set the following parameters under **Setting 2**:

- 1) The utility of the entire network $U = 1000$.
- 2) Maintain minimum service effectiveness requirements for the mission $U_n = 80\%U$.

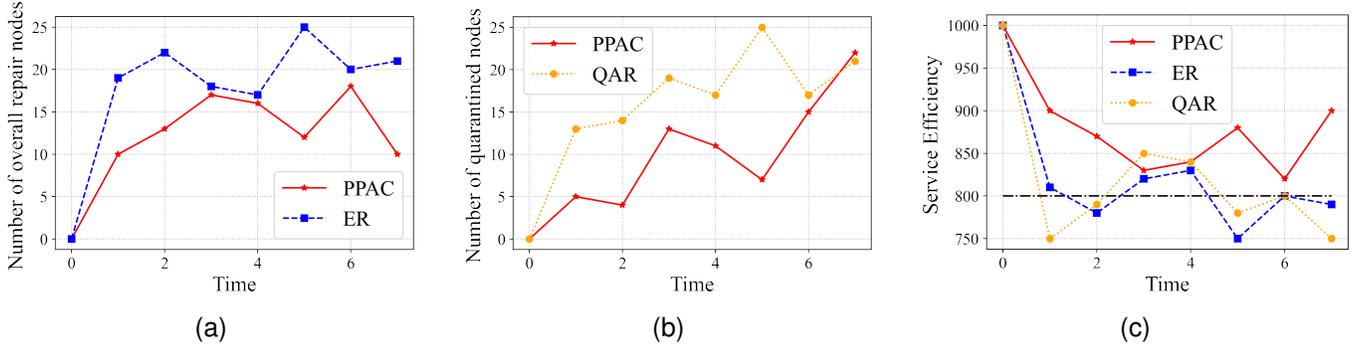


Fig. 7: (a) the comparison between our method and the ER by the number of nodes for overall maintenance at each moment in setting 2. (b) the comparison between our method and the QAR by the number of quarantined nodes at each moment in setting 2. (c) the comparison between our method and ER, QAR by the service quality at each time.

- 3) Threaten rate list and recovery rate list T and R are obtained by executing the TRG and RRG algorithms.
- 4) The specific representation of each function is

$$\phi(\delta) = (\sqrt{\delta}, \dots, \sqrt{\delta}), \mathbf{e}^1(\gamma) = (\sqrt{\gamma}, \dots, \sqrt{\gamma}),$$

$$\mathbf{e}^2(1 - \gamma) = (\sqrt{1 - \gamma}, \dots, \sqrt{1 - \gamma}).$$

Executing the solving mechanism, we get the potential defense strategy. Fig. 6 (a)-(d) plots four threat, quarantine, and recovery rate functions in the strategy. From the figure, we can see that the importance of our approach to the node's service effectiveness and the overall maintenance of resources results in less maintenance of the attacked nodes in the early stages of the whole task. Therefore, it leads to a later surge in the number of nodes to maintain effectiveness and mitigate the previous impact. Fig. 7 (a) shows the number of overall maintenance for the node at each moment in setting 2. We can see that the overall curve of the number of repairs for PPAC at each moment is significantly lower than ER, which shows that PPAC achieves accurate control under periodic time-varying networks. Fig. 7 (b) represents the number of quarantined nodes at each moment in setting 2. The overall curve of the number of quarantined nodes for PPAC at each moment is significantly lower than QAR.

In summary, the resource occupancy of PPAC, QAR, and ER are 9.61%, 14.5%, and 15.14%, respectively. The defense resource utilization of PPAC rises 61.85%, QAR rises 37.93%, and ER rises 32.08%. The service stability of each method is as follows: PPAC 100%, QAR 41.67%, and ER 66.67%.

C. Numerical Results of Time-varying Network (Setting 3)

Using the Pajek software [56], we simulate a time-varying network G_{TV} with $N = 100$ nodes, the edge-rewiring probability is 0.1 and maintain network connectivity. The $A_{TV}(t)$ represents the adjacency matrix of G_{TV} at time t and \hat{T} represents the graph sequence of G_{TV} .

Consider the APT repair game in time-varying Network:

$$\mathbb{M} = (G_{TV}, \mathbf{a}_i^{1*}, \mathbf{a}_i^{2*}, \mathbf{b}_i^*, \phi, \mathbf{e}^1, \mathbf{e}^2, \alpha, \beta, \delta^1, \delta^4,$$

$$\lambda^1, \lambda^6, \gamma^1, \gamma^5, \hat{T}, R, T, U_n, U, \mathbf{E}_0),$$

where $G = G_{TV}$, $T = 12$, and $\hat{T} = \{A_1 = (0, 2), A_2 = (2, 5), A_3 = (5, 6), A_4 = (6, 9), A_5 = (9, 11), A_6 = (11, 12)\}$. We conduct a statistical analysis based on the structural changes

of the Xi'an Jiaotong University AWD platform (Attack-Defense War platform) and the data from attack-defense exercises on the platform, to provide real-world settings for time-varying networks. We set the following parameters under **Setting 3**:

- 1) The utility of the entire network $U = 1000$.
- 2) Maintain minimum service effectiveness requirements for the mission $U_n = 80\%U$.
- 3) Threaten rate list and recovery rate list T and R are obtained by executing the TRG,RRG algorithms.
- 4) The specific representation of each function is

$$\phi(\delta) = (\sqrt{\delta}, \dots, \sqrt{\delta}), \mathbf{e}^1(\gamma) = (\sqrt{\gamma}, \dots, \sqrt{\gamma}),$$

$$\mathbf{e}^2(1 - \gamma) = (\sqrt{1 - \gamma}, \dots, \sqrt{1 - \gamma}).$$

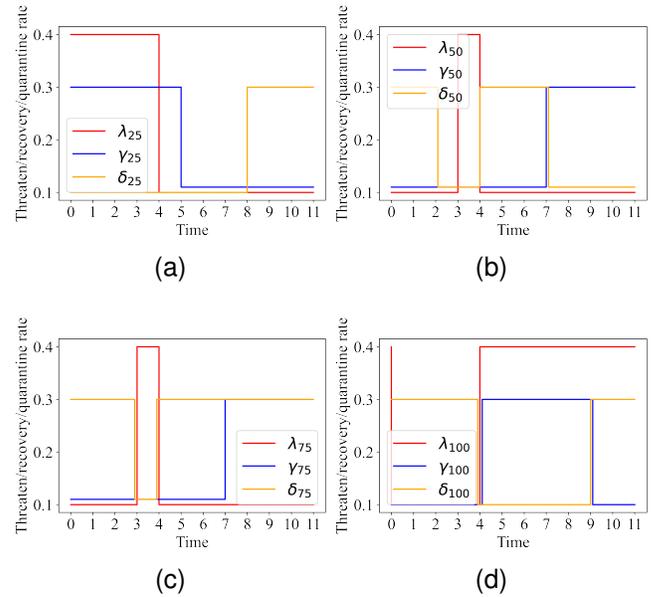


Fig. 8: A sketch of the PPAC strategy in setting 3.

Executing the solving mechanism, we get the potential defense strategy. Fig. 8 (a)-(d) plots threat, quarantine, and recovery rate functions in the strategy. Fig. 9 (a) represents the number of overall maintenance for the node at each moment in setting 3. We can see that the overall curve of the number of repairs for PPAC

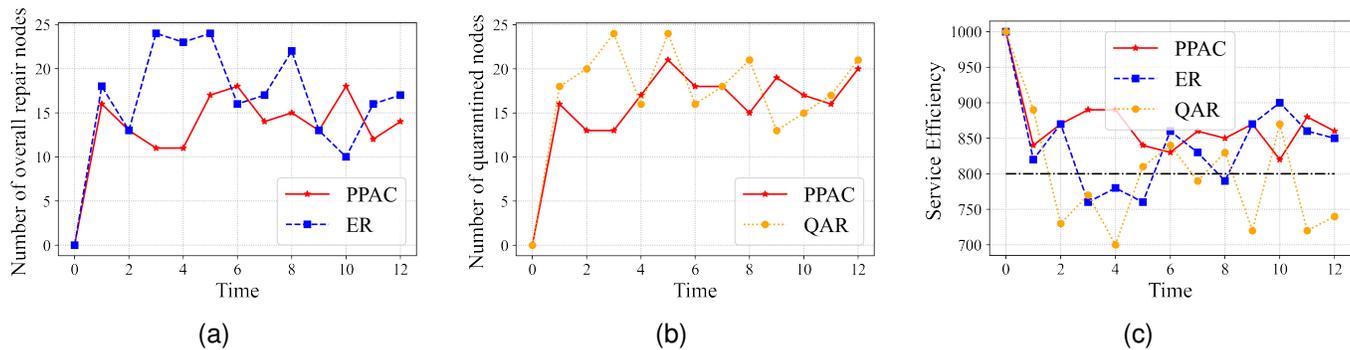


Fig. 9: (a) the comparison between our method and the ER by the number of nodes for overall maintenance at each moment in setting 3. (b) the comparison between our method and the QAR by the number of quarantined nodes at each moment in setting 3. (c) the comparison between our method and ER, QAR by the service quality at each time.

at each moment is significantly lower than ER, which shows that our solution achieves accurate control under time-varying networks. Fig. 9 (b) represents the number of quarantined nodes at each moment in setting 3. The overall curve of the number of quarantined nodes for PPAC at each moment is significantly lower than QAR. PPAC maintains a stable and high-quality service performance of the network and achieves the victim status of the node. Fig. 9 (c) represents the service quality at each moment in setting 3. We can see that all the methods except PPAC have a performance lower than the required performance at some time.

In summary, the resource occupancy of PPAC, QAR, and ER are 11.40%, 10.48%, and 11.79%, respectively. The defense resource utilization of PPAC rises 75.50%, QAR rises 36.56%, and ER rises 44.14%. The service stability of each method is as follows: PPAC 100%, QAR 58.3%, and ER 66.67%.

D. Key Insights

Our simulation is based on the rules of AWD (Attack with Defence) competitions [57], [58], and the strategies we derive can be directly applied during the exercise. For example, **Setting 3** involves a dynamic attack and defense battle lasting 12 hours, and the strategies obtained from our method can be directly used by the defenders. In other words, the $\Lambda(t)$, $\Delta(t)$, $\Gamma(t)$ in strategy $w(t)$ represent the recommended probabilities for performing specific actions on each node.

In the real world, defenders can simulate such exercises by placing their own organizations within a set of rules tailored to their needs. By simulating these exercises and obtaining strategies during the exercise, defenders can identify weaknesses in their organization, make adjustments, and receive tactical training to better respond to attacks.

VII. SUMMARY AND FUTURE WORK

This paper investigates a new problem of defending against APT attacks under time-varying networks. We model the interactions among attacks and defenders in an APT process as a dynamic APT repair game and derive its near-optimal solution by formulating the optimization problem as a PPAC problem. We derive a reliable solving mechanism that can search for a timely defense strategy with cost savings and efficiency. Specifically, our defense strategy assigns discriminatory weights for each node to reveal critical nodes and the attacking intent of the adversary.

Experimental results validate the effectiveness of defense performance under a moderately sized and time-varying network.

For future work, it is interesting to analyze the influence of variation of nodes in the heterogeneous networks scenario where attackers do not know the participation strategies of different types of nodes or in the asymmetric information scenario where the node type and function are the secret information of defenders.

REFERENCES

- [1] "Worldwide infrastructure security report Arbor Network Burlington, MA, USA White Paper 12," Jan. 2017.
- [2] Y. Wang, Z. Su, N. Zhang, R. Xing, D. Liu, T. H. Luan, and X. Shen, "A survey on metaverse: Fundamentals, security, and privacy," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 319–352, 2023.
- [3] Y. Wang, Y. Pan, M. Yan, Z. Su, and T. H. Luan, "A survey on ChatGPT: AI-generated contents, challenges, and solutions," *IEEE Open Journal of the Computer Society*, 2023, doi:10.1109/OJCS.2023.3300321.
- [4] "360's attack cases database," <https://sc.360.net/>.
- [5] L.-X. Yang, P. Li, X. Yang, and Y. Y. Tang, "A risk management approach to defending against the advanced persistent threat," *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 6, pp. 1163–1172, 2018.
- [6] L.-X. Yang, P. Li, X. Yang, Y. Xiang, F. Jiang, and W. Zhou, "Effective Quarantine and Recovery Scheme Against Advanced Persistent Threat," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 10, pp. 5977–5991, Oct. 2021.
- [7] L.-X. Yang, P. Li, Y. Zhang, X. Yang, Y. Xiang, and W. Zhou, "Effective Repair Strategy Against Advanced Persistent Threat: A Differential Game Approach," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 7, pp. 1713–1728, Jul. 2019.
- [8] Y. Li, W. Dai, J. Bai, X. Gan, J. Wang, and X. Wang, "An intelligence-driven security-aware defense mechanism for advanced persistent threats," *IEEE Transactions on Information Forensics and Security*, vol. 14, pp. 646–661, 2018.
- [9] P. Li, X. Yang, Q. Xiong, J. Wen, and Y. Y. Tang, "Defending against the advanced persistent threat: An optimal control approach," *Security and Communication Networks*, 2018.
- [10] R. Zhang and Q. Zhu, "Flipin: A game-theoretic cyber insurance framework for incentive-compatible cyber risk management of internet of things," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2026–2041, 2019.
- [11] S. Feng, Z. Xiong, D. Niyato, and P. Wang, "Dynamic resource management to defend against advanced persistent threats in fog computing: A game theoretic approach," *IEEE Transactions on Cloud Computing*, vol. 9, no. 3, pp. 995–1007, 2019.
- [12] Y. Wang, W. Chen, T. H. Luan, Z. Su, Q. Xu, R. Li, and N. Chen, "Task offloading for post-disaster rescue in unmanned aerial vehicles networks," *IEEE/ACM Transactions on Networking*, vol. 30, no. 4, pp. 1525–1539, 2022.
- [13] Y. Wang, Z. Su, J. Ni, N. Zhang, and X. Shen, "Blockchain-empowered space-air-ground integrated networks: Opportunities, challenges, and solutions," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 1, pp. 160–209, 2021.

- [14] Y. Wang, Z. Su, T. H. Luan, J. Li, Q. Xu, and R. Li, "Seal: A strategy-proof and privacy-preserving uav computation offloading framework," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 5213–5228, 2023.
- [15] Y. Wang, Z. Su, Q. Xu, R. Li, T. H. Luan, and P. Wang, "A secure and intelligent data sharing scheme for uav-assisted disaster rescue," *IEEE/ACM Transactions on Networking*, 2023, doi:10.1109/TNET.2022.3226458.
- [16] L. Xiao, X. Wan, C. Dai, X. Du, X. Chen, and M. Guizani, "Security in mobile edge caching with reinforcement learning," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 116–122, 2018.
- [17] T. T. Nguyen and V. J. Reddi, "Deep reinforcement learning for cyber security," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [18] M. Wang, C. Xu, X. Chen, L. Zhong, Z. Wu, and D. O. Wu, "Bc-mobile device cloud: A blockchain-based decentralized truthful framework for mobile device cloud," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 1208–1219, 2020.
- [19] "Supported by nasa and spacex, the first-ever satellite-focused hacking competition, hack-a-sat," <https://cyberscoop.com/mhackeroni-hackasat-space-def-con/>.
- [20] "The computer security convention def con," <https://defcon.org/>.
- [21] "Below are the tactics and techniques representing the mitre atck&matrix for enterprise," <https://attack.mitre.org/matrices/enterprise/>.
- [22] J. Lu and G. Chen, "A time-varying complex dynamical network model and its controlled synchronization criteria," *IEEE Transactions on Automatic Control*, vol. 50, no. 6, pp. 841–846, 2005.
- [23] J. O. Kephart and S. R. White, "Directed-graph epidemiological models of computer viruses," in *Computation: the micro and the macro view*, 1992, pp. 71–102.
- [24] S. Staniford, V. Paxson, N. Weaver *et al.*, "How to own the internet in your spare time," in *USENIX security symposium*, vol. 2, 2002, pp. 14–15.
- [25] P. Wang, M. C. González, C. A. Hidalgo, and A.-L. Barabási, "Understanding the spreading patterns of mobile phone viruses," *Science*, vol. 324, no. 5930, pp. 1071–1076, 2009.
- [26] S.-M. Cheng, W. C. Ao, P.-Y. Chen, and K.-C. Chen, "On modeling malware propagation in generalized social networks," *IEEE Communications Letters*, vol. 15, no. 1, pp. 25–27, 2010.
- [27] R. W. Darling and J. R. Norris, "Differential equation approximations for markov chains," 2008.
- [28] L. Lapidus and J. H. Seinfeld, *Numerical solution of ordinary differential equations*, 1971.
- [29] L. Zino and M. Cao, "Analysis, prediction, and control of epidemics: A survey from scalar to dynamic network models," *IEEE Circuits and Systems Magazine*, vol. 21, no. 4, pp. 4–23, 2021.
- [30] M. Van Dijk, A. Juels, A. Oprea, and R. L. Rivest, "Flipit: The game of stealthy takeover," *Journal of Cryptology*, vol. 26, no. 4, pp. 655–713, 2013.
- [31] L. Huang and Q. Zhu, "Adaptive strategic cyber defense for advanced persistent threats in critical infrastructure networks," *ACM SIGMETRICS Performance Evaluation Review*, vol. 46, no. 2, pp. 52–56, 2019.
- [32] Q. Zhu and S. Rass, "On multi-phase and multi-stage game-theoretic modeling of advanced persistent threats," *IEEE Access*, vol. 6, pp. 13958–13971, 2018.
- [33] D. Ye, T. Zhu, S. Shen, and W. Zhou, "A Differentially Private Game Theoretic Approach for Deceiving Cyber Adversaries," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 569–584, 2021.
- [34] Y. Diogenes and E. Ozkaya, *Cybersecurity-Attack and Defense Strategies: Infrastructure security with Red Team and Blue Team tactics*, 2018.
- [35] J. Zhao, L.-X. Yang, X. Zhong, X. Yang, Y. Wu, and Y. Y. Tang, "Minimizing the impact of a rumor via isolation and conversion," *Physica A: Statistical Mechanics and its Applications*, vol. 526, p. 120867, 2019.
- [36] W. J. Stewart, "Introduction to the numerical solution of markov chains," in *Introduction to the Numerical Solution of Markov Chains*, 2021.
- [37] C. Xiong, T. Zhu, W. Dong, L. Ruan, R. Yang, Y. Cheng, Y. Chen, S. Cheng, and X. Chen, "Conan: A practical real-time apt detection system with high accuracy and efficiency," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 01, pp. 551–565, 2022.
- [38] I. Ghafir, M. Hammoudeh, V. Prenosil, L. Han, R. Hegarty, K. Rabie, and F. J. Aparicio-Navarro, "Detection of advanced persistent threat using machine-learning correlation analysis," *Future Generation Computer Systems*, vol. 89, pp. 349–359, 2018.
- [39] S. M. Milajerdí, R. Gjomemo, B. Eshete, R. Sekar, and V. Venkatakrishnan, "Holmes: real-time apt detection through correlation of suspicious information flows," in *2019 IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 1137–1152.
- [40] D. Ye, T. Zhu, S. Shen, and W. Zhou, "A differentially private game theoretic approach for deceiving cyber adversaries," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 569–584, 2020.
- [41] M. Husák, V. Bartoš, P. Sokol, and A. Gajdoš, "Predictive methods in cyber defense: Current experience and research challenges," *Future Generation Computer Systems*, vol. 115, pp. 517–530, 2021.
- [42] R. B. Myerson, *Game theory: analysis of conflict*, 1991.
- [43] M. Deng, H. Tian, and X. Lyu, "Adaptive sequential offloading game for multi-cell mobile edge computing," in *2016 23rd international conference on telecommunications (ICT)*, 2016, pp. 1–5.
- [44] H. Kuhn¹, "Extensive games and the problem of information," *Contributions to the Theory of Games*, no. 28, p. 193, 1953.
- [45] T. Wrightson, *Advanced persistent threat hacking: the art and science of hacking any organization*, 2014.
- [46] J. Dawkins and J. Hale, "A systematic approach to multi-stage network attack analysis," in *Second IEEE International Information Assurance Workshop, 2004. Proceedings.*, 2004, pp. 48–56.
- [47] D. Fudenberg and J. Tirole, *Game theory*, 1991.
- [48] P. Fang, P. Gao, C. Liu, E. Ayday, K. Jee, T. Wang, Y. F. Ye, Z. Liu, and X. Xiao, "Back-propagating system dependency impact for attack investigation," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 2461–2478.
- [49] A. Alsaheel, Y. Nan, S. Ma, L. Yu, G. Walkup, Z. B. Celik, X. Zhang, and D. Xu, "Atlas: A sequence-based learning approach for attack investigation," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021, pp. 3005–3022.
- [50] A. Tuor, S. Kaplan, B. Hutchinson, N. Nichols, and S. Robinson, "Deep learning for unsupervised insider threat detection in structured cybersecurity data streams," in *Workshops at the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
- [51] J. Zengy, X. Wang, J. Liu, Y. Chen, Z. Liang, T.-S. Chua, and Z. L. Chua, "Shadewatcher: Recommendation-guided cyber threat analysis using system audit records," in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 489–506.
- [52] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, pp. 35–41, 1977.
- [53] G. Maji, A. Namtirtha, A. Dutta, and M. C. Malta, "Influential spreaders identification in complex networks with improved k-shell hybrid method," *Expert Systems with Applications*, vol. 144, p. 113092, 2020.
- [54] H.-J. Li, W. Xu, S. Song, W.-X. Wang, and M. Perc, "The dynamics of epidemic spreading on signed networks," *Chaos, Solitons & Fractals*, vol. 151, p. 111294, 2021.
- [55] D. Bertsekas, *Reinforcement learning and optimal control*, 2019.
- [56] W. De Nooy, A. Mrvar, and V. Batagelj, *Exploratory social network analysis with Pajek: Revised and expanded edition for updated software*. Cambridge university press, 2018, vol. 46.
- [57] "Large-scale awd competition events held worldwide," <https://ctftime.org/>.
- [58] "Rules and regulations of awd," <https://ctf-wiki.mahaloze.re/introduction/experience/>.



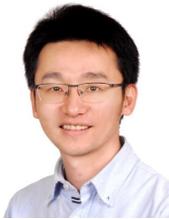
Zixuan Wang received the B.E. degree from School of Materials Science and Engineering, Xi'an University of Architecture and Technology, Xi'an, China, in 2020. He is currently pursuing his M.S. degree in the School of Cyberspace Security, Xi'an Jiaotong University. His research interests include game theory, software and network security, and blockchain.



Jiliang Li received the Dr. rer. nat. degree in computer science from the University of Göttingen, Germany, in 2019. He is currently a Researcher Professor and PhD Supervisor with the School of Cyber Science and Engineering, Xi'an Jiaotong University, Xi'an, China. His research interests include information security, cryptography, blockchain and IoT security.



Yuntao Wang received the Ph.D. degree in Cyberspace Security from Xi'an Jiaotong University, Xi'an, China, in 2022, where he is currently an Associate Professor with the School of Cyber Science and Engineering. His research interests include security and privacy in intelligent IoT, network games, and blockchain.



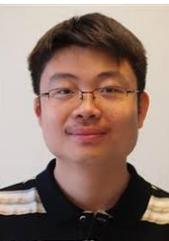
Zhou Su has published technical papers, including top journals and top conferences, such as IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE/ACM TRANSACTIONS ON NETWORKING, and INFOCOM. Dr. Su received the Best Paper Award of International Conference IEEE ICC2020, IEEE BigdataSE2019, and IEEE CyberSciTech2017. He is an Associate Editor of IEEE

INTERNET OF THINGS JOURNAL, IEEE OPEN JOURNAL OF THE COMPUTER SOCIETY, and IET COMMUNICATIONS.



Shui Yu (IEEE F23) obtained his PhD from Deakin University, Australia, in 2004. He is a Professor of School of Computer Science, Deputy Chair of University Research Committee, University of Technology Sydney, Australia. His research interest includes Cybersecurity, Network Science, Big Data, and Mathematical Modelling. He has published five monographs and edited two books, more than 500 technical papers at different venues, such as IEEE TDSC, TPDS, TC, TIFS, TMC, TKDE, TETC, ToN, and INFOCOM. His current h-index is 71. Professor Yu promoted the research field

of networking for big data since 2013, and his research outputs have been widely adopted by industrial systems, such as Amazon cloud security. He is currently serving the editorial boards of IEEE Communications Surveys and Tutorials (Area Editor) and IEEE Internet of Things Journal (Editor). He served as a Distinguished Lecturer of IEEE Communications Society (2018-2021). He is a Distinguished Visitor of IEEE Computer Society, and an elected member of Board of Governors of IEEE VTS and ComSoc, respectively. He is a member of ACM and AAAS, and a Fellow of IEEE.



Weizhi Meng is currently an Associate Professor in the Department of Applied Mathematics and Computer Science, Technical University of Denmark (DTU), Denmark. He obtained his Ph.D. degree in Computer Science from the City University of Hong Kong, Hong Kong. Prior to joining DTU, he worked as Research Scientist in Institute for Infocomm Research, A*Star, Singapore. He won the Outstanding Academic Performance Award during his doctoral study, and is a recipient of the Hong Kong Institution of Engineers (HKIE) Outstanding Paper Award for Young Engineers/Researchers in both

2014 and 2017. He also received the IEEE ComSoc Best Young Researcher Award for Europe, Middle East, & Africa Region (EMEA) in 2020. His primary research interests are cyber security and intelligent technology in security, including intrusion detection, IoT security, biometric authentication, and blockchain. He is senior member of IEEE.