# An Intelligence-Driven Security-Aware Defense Mechanism for Advanced Persistent Threats

Yuqing Li, Wenkuan Dai, Jie Bai, Xiaoying Gan, *Member, IEEE*, Jingchao Wang, and Xinbing Wang, *Senior Member, IEEE*

***Abstract*— Combined with many different attack forms, advanced persistent threats (APTs) are becoming a major threat to cyber security. Existing security protection works typically either focus on one-shot case, or separate detection from response decisions. Such practices lead to tractable analysis, but miss key inherent APTs persistence and risk heterogeneity. To this end, we propose a Lyapunov-based security-aware defense mechanism backed by threat intelligence, where robust defense strategy-making is based on acquired heterogeneity knowledge. By exploring temporal evolution of risk level, we introduce priority-aware virtual queues, which together with attack queues, enable security-aware response among hosts. Specifically, a long-term time average profit maximization problem is formulated. We first develop risk admission control policy to accommodate hosts' risk tolerance and response capacity. Under multiple attacker resources, defense control policy is implemented on two-stage decisions, involving proportional fair resource allocation and host-attack assignment. In particular, distributed auction-based assignment algorithm is designed to capture uncertainty in the number of resolved attacks, where high-risk host-attack pairs are prioritized over others. We theoretically prove our mechanism can guarantee bounded queue backlogs, profit optimality, no underflow condition, and robustness to detection errors. Simulations on real-world data set corroborate theoretical analysis and reveal the importance of security awareness.**

***Index Terms*— APT attacks, threat intelligence, security awareness, priority-based response, distributed auction algorithm.**

## I. Introduction

TODAY'S security threat landscape is experiencing an accelerating evolution, which is far more dangerous than it was ten or even five years ago [1]. Enterprises all of sizes may be overwhelmed by surging and increasingly sophisticated attacks, especially APTs with the damage and costs multiplied at a shocking rate [2]. According to the statistics of Arbor Networks, APTs have become the number one threat on the mind of over 60% of enterprise participants, jumping ahead of DDoS attacks by 2016 [3]. As ATPs' two main intrinsic properties that distinguish from typical attacks, both advancement and persistence touch upon the diversification of attack types and methods [4]. The former manifests stealth and uncertainty in attack path, rendering traditional signature approach targeting known attacks no longer adequate. While the latter indicates they always process through multiple stages over a long period of time, making single point detection technology lose desired effects. All of this is placing enormous pressure on enterprises to keep up the struggle and bringing forward higher request to 'security as a service' offerings.

*Intelligence-driven* security protection integrating detection and response capabilities would be a promising approach [5]–[6]. An intelligent defender is more informed to identify potential risks and take decisive actions to defend against APTs. With joint efforts of industry and academia, dramatic improvements in intelligent-driven protection have been made [7]. Cisco has stayed ahead of the latest threats by virtue of threat-centric security architecture [8]. As leader in intelligence-driven security-as-a-service, FireEye can identify connections between alerts, prioritize alerts and ensure intelligent and rapid response [9].

The key problem in many security protection domains is how to efficiently allocate security resources to protect targeted hosts from potential threats [10]. From perspective of attack-defense confrontation, resource allocation problem can be cast in game-theoretic contexts, providing insights on effective defense decision-making through mutual strategic behavior analysis. Extensive researches have been devoted to this subject [11]–[14]. Another appealing line of research focuses on risk management [15]–[16]. Using security paradigms like attack graphs or attack trees enables defenders assess risks based on cause-consequence relationships between network states, and further determine minimum-cost hardening measures.

We particularly identify two major challenges in defense against APTs, each of which could be addressed in this paper:

### A. Dynamic and Long-Lasting Response

To capture APTs persistence, new requirements for attack response have been raised, undercutting the ability of traditional game models that target episodic and one-off attacks [12]–[13]. Defenders are pressured to explore the right talent to provide dynamic and long-lasting response. Such demand is indispensable for keeping up with any change of attack-defense confrontation and maintaining proactive posture against APTs.

Manuscript received February 14, 2018; revised May 6, 2018; accepted June 3, 2018. Date of publication June 15, 2018; date of current version August 28, 2018. This work was supported in part by NSF China under Grant 61671478, Grant 61532012, Grant 61521062, Grant 61672342, and Grant 61602303, in part by the Science and Technology Innovation Program of Shanghai under Grant 17511105103, and in part by the open research fund of the National Mobile Communications Research Laboratory, Southeast University, under Grant 2018D06. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Sheng Zhong. *(Corresponding author: Xinbing Wang.)*

Y. Li, W. Dai, and X. Wang are with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: liyuqing@sjtu.edu.cn; daiwenkuan119@sjtu.edu.cn; xwang8@sjtu.edu.cn).

J. Bai is with Beijing Cyberxingan Technology Ltd., Beijing 100085, China (e-mail: baijie@cyberxingan.com).

X. Gan is with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China, and also with the National Mobile Communications Research Laboratory, Southeast University, Nanjing 210096, China (e-mail: ganxiaoying@sjtu.edu.cn).

J. Wang is with China Electronic Equipment System Engineering Company, Beijing 100141, China (e-mail: wangjc_61@163.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIFS.2018.2847671

### *B. Security-Aware Response*

The conflict between limited defense budget and dramatic rise in attack number highlights the necessity of security-aware response, i.e., prioritizing high-risk attacks in response. Due attention has been given to risk heterogeneity mainly on attack rate in previous works [17]–[18]. However, which begs the questions: besides heterogeneous rate, is there any new prominent heterogeneity in host security state, especially under threat intelligence.

To this end, we develop a Lyapunov-based intelligence-driven defense mechanism to enable long-lasting and security-aware response among risky hosts. Consider a defense system with $N$ independent hosts, an attacker and a defender. Backed by threat intelligence, we construct an attack graph that explicitly models attack-defense confrontation. Inspired by FlipIt game, each confrontation outcome manifests itself as attack graph changes, where each player takes control over the target host by flipping it subject to a cost. From perspective of the defender, total system profit is the difference between defense utility gained from resolving attacks and defense cost incurred. We are interested in the long-term time average system profit.

Our study highlights the intriguing role of perturbed Lyapunov optimization, where weights used for defense decision-making are carefully perturbed. To accommodate host risk tolerance and avoid resource under-utilization, we develop tolerable risk admission control policy by pushing host risk levels towards certain values. By exploring temporal evolution of risk level, we formulate it as priority-aware virtual queues, which combined with attack queues, provide an integration of host heterogeneity to queueing optimization. The defense control policy involving resource allocation and host-attack assignment is conducted. Such assignment issue, a minimum cost maximization flow problem in essence, is non-trivial to solve, further complicated by uncertainty in the number of resolved attacks. We construct a virtual auction market, where attack events bid for response chances provided by associated hosts. The proposed distributed auction-based assignment algorithm realizes security-aware response among hosts.

Our main contributions are highlighted as follows.

- We propose a Lyapunov-based intelligence-driven defense mechanism against APTs. A salient contribution of our approach is that, risk levels' temporal evolution is captured by priority-aware virtual queues, which combined with attack queues enable security-aware response among hosts. To our knowledge, we are the first to explore security awareness in defense policy from both attack and risk level perspectives.
- Backed by threat intelligence, we model attack-defense confrontation process as the dynamic attack graph. A long-term time average profit maximization problem involving risk admission and response is formulated. To accommodate host risk tolerance, we develop tolerable risk admission control policy by perturbing weights used for defense decision-making. For each host, security-aware defense control policy is implemented on two-stage decisions. Specifically, we first propose distributed auction-based assignment algorithm, where high-risk host-attack pairs are prioritized over others, and then provide proportional fair resource allocation among winning attack events. Furthermore, we theoretically prove our mechanism can guarantee bounded queue backlogs, profit optimality and robustness to detection errors.
- We apply our defense mechanism to a practical machine learning-based anomaly detection system developed by security company, Cyberxingan [19]. Extensive experimental results validate our analysis on the efficiency of our mechanism and reveal the importance of security awareness.

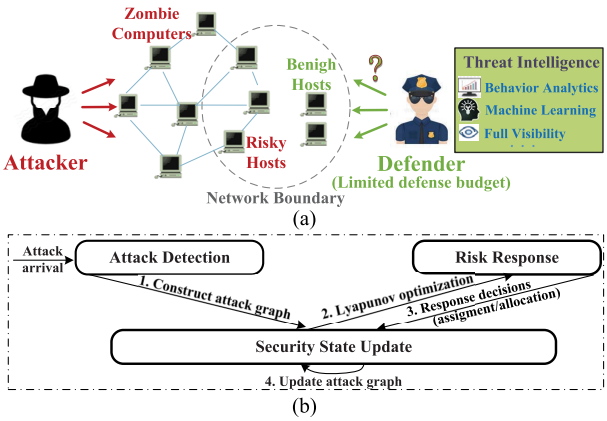



Fig. 1. (a) General intelligence-driven defense system. (b) Illustration of the proposed Lyapunov-based intelligence-driven defense mechanism.

In what follows, we introduce system model and problem formulation in Section II. To proceed, we propose a Lyapunov-based defense mechanism and analyze its performance in Section III. In Section IV, we apply it to a practical detection system. Finally, simulations, related works and conclusions are shown in Sections V, VI and VII. *Due to the space limitation, all technical proofs are provided in the Appendix*.

## II. System Model and Problem Formulation

### *A. The Intelligence-Driven Defense System Model*

Consider the general intelligence-driven defense system consisting of two agents and $N$ independent end hosts containing valuable data that need to be protected, as shown in Fig. 1. The agent who wants to attack the network to achieve some specific goals is called the attacker, while the other agent who tries to defend hosts and minimize attack effects is called the defender. To avoid being trapped, the attacker usually uses multiple zombie computers to launch attacks simultaneously. Suppose one zombie computer carries out only one attack.[1] For a zombie computer launching multiple attacks, we treat it as multiple zombie computers. Backed by threat intelligence, the defender first identifies potential risky zombie computers and infected hosts, and then determines when and which hosts to secure under limited defense budget. Such practice actually constitutes the essence of intelligence-driven defense. Specifically, "intelligence" refers to the threat information acquired by the practical anomaly detection system, as shown in Section IV, while "driven" suggests that our priority-based response policy designed later highly depends on detection results. The time is slotted for $t \in \mathcal{T} = \{0, 1, \cdots\}$, and the following processes will be performed for each time slot:

- *Attack Detection:* Backed by threat intelligence, potential attacks are expected to be identified. The acquired threat knowledge can be explicitly captured by dynamic attack graph.

[1]In the rest of this paper, we will use "zombie computer" and "attack" interchangeably. The same is with "host" and "attack queue".

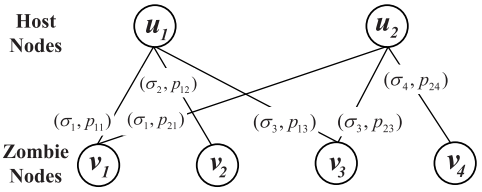



Fig. 2. An example of attack graph with two hosts and four zombie computers.

- *Risk Response:* Under Lyapunov optimization, optimal response decisions are determined, where distributed auction-based assignment algorithm is designed for solution optimality with guaranteed finite termination.
- *Security State Update:* Any move of attack-defense confrontation may result in attack graph changes, which are desired to be closely followed, such as edge adding for attack detection and edge updating/deleting for risk response.

*1) Attack Detection:* Imagine the typical attack-defense confrontation. The attacker schemes to launch attack via zombie computer $j$, and unfortunately host $i$ is on the target list. Such attack is detected at a certain time via detection scheme. Once realizing the potential risk posed by attack $j$, the defender does everything to resolve it, and then host $i$ will be recovered after a period of response time. The FlipIt game was recently proposed in security community to model such confrontation process, especially in the context of advanced persistent threats [15].[2] In typical FlipIt, two players (called herein defender and attacker) continuously compete for the control of the target host. Each player takes control over it by flipping subject to a cost. Most importantly, a player only finds out system state when moving, instead of immediately knowing when the other player moves. As the extension of FlipIt game, our work focuses on multi-host setting, which together with limited resource constraints poses challenges in modeling attack-defender interaction and characterizing optimal strategies.

Formally, let $\boldsymbol{U}[t]$ and $\boldsymbol{V}[t]$ denote the sets of hosts and zombie computers at slot $t$. In the following, we use $\boldsymbol{U}$, $\boldsymbol{V}$ instead of $\boldsymbol{U}[t]$,$\boldsymbol{V}[t]$ when there is no confusion. Formally, the confrontation of the entire system can be captured by an attack graph $\mathcal{G}^{\text{Attack}} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$ shown in Fig. 2. Host node $u_i$ is introduced in $\mathcal{U}$ for every host $i \in \boldsymbol{U}$, and zombie node $v_j$ is introduced in $\mathcal{V}$ for every detected zombie computer $j \in \boldsymbol{V}$. $\mathcal{E}$ is a set of system security state, namely which hosts are compromised by which zombie computers at which attack and risk levels. If host $i$ is under attack $j$, we add an edge $(u_i, v_j) \in \mathcal{E}$ between $u_i$ and $v_j$ with the combination weight of attack load and risk level denoted by $(\sigma_j, p_{ij})$. A desirable defense mechanism is able to capture dynamics of zombie node set $\mathcal{V}$ and edge set $\mathcal{E}$.

Consider a set of attacks arrive online.[3] Backed by threat intelligence, each attack $j$ can be specified by a tuple $\theta_j = (t_j, \omega_j, \sigma_j, \boldsymbol{N}_j)$, where it is detected at time $t_j$ with risk score or seriousness, $\omega_j \in [\omega^{\min}, \omega^{\max}]$; attack load $\sigma_j \in [\sigma^{\min}, \sigma^{\max}]$ captures the number of slots requested for response, determined by how long it takes attacker to compromise one host. $\boldsymbol{N}_j$ is the set of infected hosts under the attack of $j$. Such threat information is shared by all attack events launched via the same zombie computer. Regard any attack arrival time as the time when it is detected. The instantaneous attack arrival brought by zombie computer $j$ to host $i$ is

$$A_{ij}[t] = \sigma_j \cdot 1_{\{j \in \boldsymbol{N}_i[t], j \notin \boldsymbol{N}_i[t-1]\}}. \tag{1}$$

Attack arrivals of the system can be given by the vector $\boldsymbol{A}[t] = \{A_{ij}[t] | i \in \boldsymbol{U}, j \in \boldsymbol{V}\}$, where $0 \le A_{ij}[t] \le A^{\max}$. With the development of detection technology, attacks usually can be detected with satisfactory accuracy. In particular, we implement our defense mechanism into practical detection system in Section IV, where robustness to detection errors is also discussed. We're, as a consequence, going to focus on the case $\boldsymbol{A}[t]$ is known to the defender at the end of slot $t$.

*2) Risk Response:* Define the active zombie set of host $i$ as the set of attacks it is suffering, i.e., $\boldsymbol{N}_i[t] = \{j | j \in \boldsymbol{V}, (u_i, v_j) \in \mathcal{E}\}$. Based on acquired heterogeneity in host security state, the defender next takes the corresponding response actions involving target attack selection and security resource allocation.[4] In practice, limited defense budget may be in conflict with dramatic rise in attack number, making it not feasible to resolve all risky attacks at the same time. To capture resource availability to host $i$, we introduce response indicator vector $\boldsymbol{r}_i[t] = \{r_{ij}[t] | j \in \boldsymbol{N}_i[t]\}$, where $r_{ij}[t] = 1$ represents attack event $j$ gets resolved in protecting host $i$ and $r_{ij}[t] = 0$ otherwise. Inspired by non-preemptive scheduling proposed in [21], we consider a more general model: attack $j$ who gets resolved at $t$ continues to occupy resources for $\sigma_j$ slots, directly affecting resource provisioning and response decisions at subsequent slots. Host $i$ will be recovered after $\sigma_j$ slots. For each host, we explicitly model limited defense budget by placing an upper bound on the number of resolved attack events, i.e., $\sum_{j \in \boldsymbol{N}_i[t]} r_{ij}[t] \le B_i, \forall i \in \boldsymbol{U}$, where budget $B_i$ is assumed to be host-specific to capture heterogeneity in host response capacity. To avoid response chances being concentrated on certain attacks, suppose each attack has a parallelism bound $E_j$, i.e., at most $E_j$ attack events launched via zombie computer $j$ can be run in parallel. The response parallelism constraint can be formally represented as $\sum_{i \in \boldsymbol{N}_j} r_{ij}[t] \le E_j, \forall j \in \boldsymbol{V}$. The response rate that host $i$ obtains can thus be

$$r_i[t] = \sum\nolimits_{j \in \boldsymbol{N}_i[t]} \sigma_j \cdot r_{ij}[t], \tag{2}$$

characterizing the amount of attack loads resolved at slot $t$. In addition to such response decisions, how to allocate resources among target attacks is of equal importance to improve defense efficiency, where the allocated resources directly affect how long it will take the attacker to compromise hosts again. Obviously, inefficient allocation will reduce response efficiency and worsen security state of the whole system. In protecting host $i$, the proportion of resources allocated to attack $j$ denoted by $b_{ij}[t]$, should satisfy $\sum_{j \in \boldsymbol{N}_i[t]} b_{ij}[t] \cdot r_{ij}[t] = 1$.

[2]Actually, the FlipIt game model of interest can be implemented in a variety of current real-world applications. Take zero-day exploit with the target resource being computing devices [20], for example. The attacker aims to compromise the device by exploiting a software vulnerability, while the defender focuses on keeping the device clean through software reinstallation, patching, or other defensive steps. The FlipIt provides guidance on how to implement effective decisions, i.e., "When to launch the next attack?" for the attacker and "How regularly should I clean machines?" for the defender.

[3]In practice, the distribution of attack arrival is random and unpredictable. And even if predictable, the prediction accuracy cannot be guaranteed. It is reasonable to assume that attack arrival is random and as the consequence of network optimization, response decision is strategic [12]–[13].

[4]Security resources range from computing resources for vulnerability scanning, to hardware resources for intrusion detection, and so on.

Under limited defense budget, the above response process begs the question: what if there exist too many attack events left untreated? In general, attacks even minor vulnerabilities, if left untreated too long, are more likely to incur irreparable losses and pose huge pressure to the whole defense system. To avoid this dilemma, we suppose response delay for any attack is upper bounded by $D_{\max}$. Such guarantee is violated when explosive attack events go beyond the defender's response capacity. We model the potential dropping of an attack when its response guarantee cannot be met. The dropping decision can be captured by a binary decision variable $d_{ij}[t]$, where $d_{ij}[t] = 1$ represents attack $j$ is dropped from host $i$'s active zombie set and $d_{ij}[t] = 0$ otherwise. Taking attack loads into account, the dropping rate of host $i$ can be characterized as

$$d_i[t] = \sum\nolimits_{j \in \boldsymbol{N}_i[t]} \sigma_j \cdot d_{ij}[t], \tag{3}$$

which is upper-bounded by $d^{\max}$. In practice, however, the defender may never drop any detected attacks, especially APTs with specific goals. The "dropping" here can be interpreted as follows. The defender maintains a set of regular resources (under limited budget) while keeping a set of backup resources just in case, whose provisioning is always expensive due to the advance of adopted defense technology. An attack will be "dropped" if there haven't been available regular resources until the maximum response delay. Then more efforts (i.e., expensive backup resources) will be invested to resolve it subject to a dropping penalty. We introduce the Maximum Response Guarantee (MRG) requirement, which is promising to realize high response efficiency under stable security state.

*Definition 1:* The MRG requirement is satisfied only if all detected attack events will be either resolved or dropped before the maximum response delay $D_{\max}$.

*3) Security State Update:* Security state can be captured by attack graph $\mathcal{G}^{\text{Attack}}$. For each host node $u_i$, neighboring zombie nodes denote attacks that host $i$ is suffering, and host-zombie edge weights suggest attack and risk levels that host $i$ is under the attack. Intuitively, any move of attack-defense confrontation corresponds to attack graph changes. To facilitate online defense mechanism design, security state updates should closely follow such changes, mainly reflected in the following three aspects.

- *Active Zombie Set:* Attack arrival or response may lead to the changes of active zombie set $\boldsymbol{N}_i[t]$. Specifically, add a new zombie node to $\boldsymbol{N}_i[t]$ when its launched attack events are admitted for response; and delete zombie nodes from $\boldsymbol{N}_i[t]$ if all of their target hosts are recovered from being compromised.
- *Attack Level:* Attack level are captured from the perspective of untreated attack loads. Intuitively, stochastic response and dropping process, together with time-varying attack arrivals, may bring about dynamics of host attack loads. Taking ATPs persistence into account, we apply queueing theory to handle such dynamics in the long-term confrontation. In particular, each host $i$ maintains an individual attack queue with queue backlog $Q_i[t]$ characterizing the amount of untreated attack loads. We adopt the convention that attack response and dropping at slot $t$ happen at the beginning of the slot, while arrivals happen at the end, which coordinates the design of threat intelligence-driven defense mechanism. For attack queue of host $i \in \boldsymbol{U}$, queueing dynamics can be described as

$$Q_i[t+1] = \max\{Q_i[t] - r_i[t] - d_i[t], 0\} + a_i[t], \tag{4}$$

where $a_i[t] = \sum_{j \in \boldsymbol{V}} a_{ij}[t]$ denotes the increment of attack loads brought by newly arrived attack events, and $0 \le a_{ij}[t] \le A_i[t]$ indicates not all attack events are allowed into queues due to limited response capacity. To prevent security state worsening, one promising approach is to resort to advanced defense techniques to address these attack events, rather than admitting them into queues for endless response waiting. A queue is stable only if it has a bounded time-average backlog [22], i.e., $\overline{Q}_i = \limsup_{t\to\infty} \frac{1}{t} \sum_{\tau=0}^{t-1} \mathbb{E}\{Q_i[\tau]\} < \infty$.

- *Risk Level:* Risk level, combined with attack level, constitutes the joint weight for host-zombie edges in attack graph, accounting for heterogeneity in host security state. For each host, the value of risk level depends on risk seriousness resulted from target zombie computers as well as how long it has been compromised. Specifically, risk seriousness captured by risk score $\omega_j$, may vary from one attack to another due to the diversity of adopted attack techniques. Too many untreated attacks further pose huge pressure to defense mechanism. The longer one host is compromised, the more serious damage it will suffer, especially for dedicated and purpose-built APTs. We are inspired to model the dynamic risk level of each attack event $j$ towards host $i$ as an exponential function, i.e.,

$$p_{ij}[t] = \omega_j e^{t-t_j} \cdot 1_{\{j \in \boldsymbol{N}_i[t]\}}, \quad t_j \le t \le t_j + D_{\max}, \tag{5}$$

where $t - t_j$ denotes the time when host $i$ has been compromised. Once attack event $j$ associated with host $i$ is identified, i.e., $1_{\{j \in \boldsymbol{N}_i[t]\}} = 1$, add edge $(u_i, v_j)$ into attack graph, and risk level is initialized as $p_{ij}[t_j] = \omega_j$. Under MRG requirement, delete this edge if $j$ fails to be resolved at $t = t_j + D_{\max}$, and the maximum risk level $p_{ij}[t_j + D_{\max}] = \omega_j e^{D_{\max}}$ is obtained. Thus, $p^{\min} = \omega^{\min}$ and $p^{\max} = \omega^{\max} e^{D_{\max}}$. Denote the risk level vector for host $i \in \boldsymbol{U}$ as $\boldsymbol{p}_i[t] = \{p_{ij}[t] | j \in \boldsymbol{N}_i[t]\}$. Our defense mechanism is designed especially for attack-intensive scenarios. To guarantee high response efficiency, it's desired to enable priority-based response especially under limited budget. Let's introduce the concept of Preference Ranking (PR) requirement for further analysis.

*Definition 2:* Given risk level vector $\boldsymbol{p}_i[t]$, host $i$'s preference ranking $R_i[t] = \{R_i^1, \cdots, R_i^{|\boldsymbol{N}_i[t]|}\}$ in response is a permutation of $\{1, \cdots, |\boldsymbol{N}_i[t]|\}$. The PR requirement holds if

$$R_i^j \begin{cases} < R_i^k, & \text{if } p_{ij}[t] < p_{ik}[t], \\ > R_i^k, & \text{if } p_{ij}[t] \ge p_{ik}[t], \end{cases} \tag{6}$$

for any $j \in \{1, \cdots, |\boldsymbol{N}_i[t]| - 1\}$ and $k \in \{j, \cdots, |\boldsymbol{N}_i[t]|\}$. It states the defender prefers attacks with high risk score and early detection time in defense process. Under this concept, the higher preference ranking an attack event possesses, the more likely it will be selected for response.

### *B. Problem Formulation*

*1) Defense System Profit:*

*a) Defense utility:* The defender collects defense utility every time an attack event is resolved. We introduce a security-aware utility to characterize the efficiency of defense decisions in alleviating risk pressure, which is determined by two factors, i.e., risk seriousness captured by risk score

and allocated security resources. Specifically, high-risk attacks pose great pressure to response process and may result in huge defense cost. While resource allocation exerts direct influence on future host's security state, and the more resources are allocated, the stronger risk resistance it possesses, indicating the attacker needs to take longer time to compromise it again.

Intuitively, only maximizing total security-aware utility will lead to severe starvation for low-risk host-attack pairs with no defense resources allocated, making it hard to keep those attacks under control. Therefore, fairness is another issue that cannot be ignored in resource allocation, which is crucial for long-term system stability [23]. In particular, we adopt proportional fairness, a common fairness metric that was proposed by Kelly [24]. In protecting host $i$, under resource allocation decision $b_{ij}[t]$ and response decision $r_{ij}[t]$, the corresponding security-aware utility function can be defined as $U(b_{ij}[t]p_{ij}[t]r_{ij}[t])$. According to [24], proportional fairness can be formally defined as follows.

*Definition 3:* A vector of resource allocation $\boldsymbol{b}_i[t] = \{b_{ij}[t]|j \in \boldsymbol{N}_i[t]\}$, with $\boldsymbol{N}_i[t]$ being the set of attacks related to host $i$, is proportionally fair if it is feasible, and if for any other feasible allocation vector $\boldsymbol{b}'_i[t] = \{b'_{ij}[t]|j \in \boldsymbol{N}_i[t]\}$, the aggregate of proportional changes is either zero or negative, i.e., $\sum_{j\in\boldsymbol{N}_i[t]} \frac{b'_{ij}[t]-b_{ij}[t]}{b_{ij}[t]} \leq 0$.

Consider a small perturbation $b'_{ij}[t] = \delta b_{ij}[t]+b_{ij}[t]$ which increases $U(b_{ij}[t]p_{ij}[t]r_{ij}[t])$ providing that $\sum_{j\in\boldsymbol{N}_i[t]} \delta b_{ij}[t] \cdot U'(b_{ij}[t]p_{ij}[t]r_{ij}[t]) > 0$. From the definition of proportional fairness, we have $\sum_{j\in\boldsymbol{N}_i[t]} \frac{\delta b_{ij}[t]}{b_{ij}[t]} > 0$, which can be rewritten as $\sum_{j\in\boldsymbol{N}_i[t]} (r_{ij}[t]\log(b_{ij}[t]p_{ij}[t]))'\delta b_{ij}[t] > 0$. Thus, it follows that the above proportionally fair allocation can be represented by a local maximum of the logarithmic utility function. Since logarithmic function is differentiable and strictly concave, it has only one maximum and the local maximum is also the global maximum. Accordingly, the total security-aware utility under proportionally fair resource allocation can be expressed by $U_i[t] = \sum_{j\in\boldsymbol{N}_i[t]} r_{ij}[t]\log(p_{ij}[t]b_{ij}[t])$.

*b) Defense cost:* Attack events violating MRG requirement should be "dropped" to prevent security state out of control. Dropping decisions together with response decisions, constitute the major security-aware defense strategies. There is an instantaneous defense cost associated with attack response and dropping due to resource expenditure (including regular and backup resources). The total number of resolved attack events associated with host $i$ is $n_i[t] = \sum_{j\in\boldsymbol{N}_i[t]} r_{ij}[t]$. Denote $c_i[t]$ as the average cost incurred for resolving unit attack load and host $i$'s response cost at slot $t$ can thus be $C_i^r[t] = \sum_{j\in\boldsymbol{N}_i[t]} \sigma_j r_{ij}[t]c_i[t]$. Denote a penalty of $\alpha_j$ enforced for dropping attack event $j$, where $\alpha_j > \max_{i\in\boldsymbol{U}}\sigma_j c_i[t]$.[5] For host $i$, expenditure on penalty occurs at slot $t$ if there are dropped attack events, with the total amount of $C_i^d[t] = \sum_{j\in\boldsymbol{N}_i[t]} \alpha_j d_{ij}[t]$.

The instantaneous system profit can be defined as the difference between overall defense utility and cost, i.e., $f[t] = \sum_{i\in\boldsymbol{U}} U_i[t] - (C_i^r[t] + C_i^d[t])$. The time average expected profit of the defense system is $f_{\text{av}} = \limsup_{t\to\infty} \frac{1}{t}\sum_{\tau=0}^{t-1} \mathbb{E}\{f[t]\}$.

[5]Such condition not only corresponds to the fact that dropping cost is expected to be larger than all potential serving cost due to the usage of more advanced defense techniques, but also ensures high-efficiency response in designing defense control policy later.

TABLE I
MAJOR NOTATIONS

| Notation | Description |
|---|---|
| $\boldsymbol{U},i,\boldsymbol{V},j$ | set and index of hosts and zombie computers |
| $t_j, \omega_j, \sigma_j$ | detection time, risk score, attack load of attack $j$ |
| $\boldsymbol{N}_j,\boldsymbol{N}_i[t]$ | attack $j$'s infected host set, host $i$'s active zombie set |
| $a_{ij}[t],d_{ij}[t]$ | attack admission and dropping decision variables |
| $r_{ij}[t],b_{ij}[t]$ | response and resource allocation decision variables |
| $Q_i[t],p_{ij}[t]$ | attack level or attack queue, risk level |
| $Z_i[t],H_i[t]$ | delay-aware, priority-aware virtual queues |
| $f[t],U_i[t]$ | system profit, security-aware defense utility |
| $C_i^r[t],c_i[t]$ | response cost, average unit cost for response |
| $C_i^d[t],\alpha_j$ | dropping cost, penalty for dropping attack $j$ |
| $B_i[t],E_j$ | defense budget, response parallelism bound |
| $d_i^{\max},D_{\max}$ | allowable dropped attack loads, max response delay |

*2) Defense Problem:* This paper aims to develop an intelligence-driven defense mechanism to enable security-aware response under limited defense budget. The potential benefits of security-awareness can be fully explored from the perspectives of both attack level and risk level, where high-risk host-attack pairs are given priority in response.

We say a defense policy $\Pi$ is feasible if for all $t$, the defender first maintains security state $\phi[t] = \{(\boldsymbol{N}_i[t], Q_i[t], p_{ij}[t])\}$, and then makes defense decisions $\eta[t] = \{(a_{ij}[t], b_{ij}[t], r_{ij}[t], d_{ij}[t])\}$, where both MRG and PR requirements are satisfied. Our objective is to design a feasible policy $\Pi$ to maximize system profit, i.e.,

$$\text{P1:} \max_{\Pi} \ f_{\text{av}}^{\Pi} \tag{7}$$

$$\text{s.t.} \sum_{j\in\boldsymbol{N}_i[t]} \sigma_j d_{ij}[t] \leq d_i^{\max}, \quad \forall i \in \boldsymbol{U} \tag{8}$$

$$\sum_{j\in\boldsymbol{N}_i[t]} b_{ij}[t]r_{ij}[t] \leq 1, \quad \forall i \in \boldsymbol{U} \tag{9}$$

$$\sum_{j\in\boldsymbol{N}_i[t]} r_{ij}[t] \leq B_i, \quad \forall i \in \boldsymbol{U} \tag{10}$$

$$\sum_{i\in\boldsymbol{N}_j} r_{ij}[t] \leq E_j, \quad \forall j \in \boldsymbol{V} \tag{11}$$

$$\overline{a}_i \leq \overline{r}_i + \overline{d}_i, \quad \forall i \in \boldsymbol{U} \tag{12}$$

$$\text{MRG and PR Constraints.} \tag{13}$$

Specifically, constraint (8) ensures a certain response efficiency by placing an upper bound for the amount of dropped attack loads. Constraint (9) guarantees security resources will not be over-utilized. Constraints (10) and (11) specify limited defense budget and parallelism constraints. Constraint (12) guarantees the stability of attack queues. To facilitate reading, the summary of major notations is tabulated as in Table I.

## III. Security-Aware Defense Mechanism

### A. Addressing MRG and PR Requirements

To meet MRG requirement, we associate each attack queue $Q_i, \forall i \in \boldsymbol{U}$ with a delay-aware virtual queue $Z_i$. Such practice is inspired by $\epsilon$-persistent service queue technique [24], which can ensure bounded worst-case response delay. The queue backlog of $Z_i$ is updated by

$$Z_i[t+1] = \max\left\{Z_i[t] - r_i[t] - d_i[t] + \epsilon_i 1_{\{Q_i[t]>0\}}, 0\right\}, \tag{14}$$

where $\epsilon_i > 0$ are pre-specified constants. The intuition is that $Z_i$ has the same service process as attack queue $Q_i$ being $r_i[t] + d_i[t]$, but has an arrival process that adds $\epsilon_i$ whenever $Q_i$ is non-empty, ensuring queue length of $Z_i$ grows if there

are attack events in $Q_i$. By leveraging defense strategies under network stability, attack response delay will be upper bounded.

*Proposition 1:* Suppose an online defense mechanism is implemented so that $Q_i[t] \le Q_i^{\max}$ and $Z_i[t] \le Z_i^{\max}$, where $Q_i^{\max}$ and $Z_i^{\max}$ are finite upper bounds of the queue backlogs. The worst-case response delay for host $i$ is bounded by

$$D_i^{\max} = \left\lceil \left(Q_i^{\max} + Z_i^{\max}\right)/\epsilon_i \right\rceil, \tag{15}$$

where $\lceil x \rceil$ denotes the smallest integer that is no less than $x$. Accordingly, all attacks are either resolved or dropped within $D_{\max} = \max D_i^{\max}$ slots.

Under PR requirement, high-risk attacks are prioritized over others in response, facilitating security-aware defense mechanism design. Recall that risk level $p_{ij}[t] = \omega_j e^{t-t_j} 1_{\{j \in \boldsymbol{N}_i[t]\}}$, whose time evolution can be captured as

$$p_{ij}[t+1] = p_{ij}[t] \cdot e = p_{ij}[t] + p_{ij}[t](e-1). \tag{16}$$

To enable priority-based response, we introduce a novel priority-aware virtual queue [25] with queue backlog $H_i[t] = \sum_{j \in \boldsymbol{N}_i[t]} p_{ij}[t]$, which is actually the sum of risk level of untreated attack events and provides the basis for identifying which hosts are at high risk levels for which zombie computers' attack. For attack $j$ associated with host $i$, add risk level $p_{ij}[t]$ into $H_i$ when $j$ is admitted into attack queue $Q_i$, and remove $p_{ij}[t]$ from $H_i$ when $j$ is dropped or resolved. In view of dynamics of risk level and active zombie set, queue backlog of $H_i$ evolves as follows

$$H_i[t+1] = \max\left\{H_i[t] + \beta_i[t] - r_i'[t] - d_i'[t], 0\right\} + a_i'[t], \tag{17}$$

where $r_i'[t] = \sum_{j \in \boldsymbol{r}_i[t]} p_{ij}[t]e = \sum_{j \in \boldsymbol{N}_i[t]} e\sigma_j p_{ij}[t] r_{ij}[t]$, $\beta_i[t] = \sum_{j \in \boldsymbol{N}_i[t]} p_{ij}[t]\,(e-1)$, $d_i'[t] = \sum_{j \in \boldsymbol{d}_i[t]} p_{ij}[t]e = \sum_{j \in \boldsymbol{N}_i[t]} e\sigma_j \cdot p_{ij}[t] d_{ij}[t]$, $a_i'[t] = \sum_{j \in \boldsymbol{a}_i[t]} p_{ij}[t] = \sum_{j \in \boldsymbol{V}} p_{ij}[t] a_{ij}[t]$, respectively.[6] Under queue stability, virtual queue $H_i$ is stable only if attack queue $Q_i$ is stable and risk level $p_{ij}[t]$ is limited.

### B. Dynamic Algorithm Design

Let $\boldsymbol{\Theta}[t] = [\boldsymbol{Q}[t], \boldsymbol{Z}[t], \boldsymbol{H}[t]]$ as the aggregate queue vector. To start, we define the *perturbed Lyapunov function* [22] as

$$L(\boldsymbol{\Theta}[t]) = \frac{1}{2}\left\|\boldsymbol{Q}[t]\right\| + \frac{1}{2}\left\|\boldsymbol{Z}[t]\right\| + \frac{1}{2}\left\|\boldsymbol{H}[t] - \boldsymbol{\theta}\right\|, \tag{18}$$

where $\boldsymbol{\theta} = \theta_i \cdot \boldsymbol{1}^N$ with $\theta_i$ being perturbation parameters. We can understand the perturbation from the following two aspects. First, analogous to immune system, hosts usually possess certain risk tolerance. Network stability can still be guaranteed when pushing $H_i[t]$ towards a small tolerable risk level related to $\theta_i$. Here $\theta_i$ is assumed to be host specific, since different hosts may have various available security resources and defense ability. Second, to guarantee no-underflow in attack queue, $\theta_i$ can be carefully leveraged to enhance response efficiency. Under no-underflow condition, each host will have certain amount of risks to resolve, preventing resources unallocated for a long time. To achieve queue stability, we define Lyapunov drift as $\Delta(\boldsymbol{\Theta}[t]) = \mathbb{E}\{L(\boldsymbol{\Theta}[t+1]) - L(\boldsymbol{\Theta}[t]) | \boldsymbol{\Theta}[t]\}$ to capture expected changes in the quadratic function of queue backlogs over each slot. We incorporate system profit into Lyapunov drift, providing network stability and profit maximization jointly. At every slot, we try to minimize the drift-plus-penalty function greedily, i.e.,

$$\min\ \Delta(\boldsymbol{\Theta}[t]) - V\mathbb{E}\left\{f[t] | \boldsymbol{\Theta}[t]\right\}, \tag{19}$$

where $V > 0$ is a tunable parameter weighting how much importance we stress on maximizing system profit.

*Proposition 2:* Under any feasible defense policy $\Pi$, we have

$$\begin{aligned}
&\Delta(\boldsymbol{\Theta}[t]) - V\mathbb{E}\left\{f[t] | \boldsymbol{\Theta}[t]\right\} \\
&\le B_1 + B_2[t] + \sum_{i \in \boldsymbol{U}} \mathbb{E}\left\{Q_i[t] a_i[t] + (H_i[t] - \theta_i)\, a_i'[t] | \boldsymbol{\Theta}[t]\right\} \\
&\quad - \sum_{i \in \boldsymbol{U}} \mathbb{E}\{(Q_i[t] + Z_i[t]) d_i[t] + (eH_i[t] - \theta_i) d_i'[t] \\
&\quad - V C_i^d[t] | \boldsymbol{\Theta}[t]\} \\
&\quad - \sum_{i \in \boldsymbol{U}} \mathbb{E}\{(Q_i[t] + Z_i[t]) r_i[t] + (eH_i[t] - \theta_i)\, r_i'[t] \\
&\quad + V(U_i[t] - C_i^r[t]) | \boldsymbol{\Theta}[t]\}
\end{aligned} \tag{20}$$

where $B_1 = \frac{1}{2}\sum_{i \in \boldsymbol{U}} (\sigma^{\max} B_i^{\max} + d_i^{\max})^2 + (A^{\max})^2 + \max\{(\sigma^{\max} \cdot B_i^{\max} + d_i^{\max})^2, \epsilon_i^2\} + ((\sigma^{\max} B_i^{\max} + d_i^{\max}) \sigma^{\max} \omega^{\max} e^{1+D_{\max}})^2 + (A^{\max} \cdot \omega^{\max} e^{D_{\max}})^2 + \frac{(Q_i^{\max} \omega^{\max} e^{D_{\max}})^2 (e^2 - e)}{(\sigma^{\min})^2} > 0$ is a finite constant, and $B_2[t] = \sum_{i \in \boldsymbol{U}} \epsilon_i Z_i[t] + (1-e)\theta_i H_i[t]$ is a known constant at slot $t$ since queue backlogs are known at $t$.

*Remark:* Minimizing drift-plus-penalty in (19) is thus equivalent to minimizing the Right-Hand-Side (RHS) of (20), which amounts to minimizing its last three terms. Such problem involves risk admission and response subproblems, which are decoupled in decision variables. Under equivalence to drift-plus-penalty minimization, we have the following two parallel parts constituting the main idea of our mechanism.

*1) Tolerable Risk Admission Control Policy:* Risk admission decisions can be made by minimizing the third term of the RHS of (20). Since admission decisions of different hosts are independent from each other, we can concurrently obtain $\boldsymbol{a}_i[t] = \{a_{i1}[t], \cdots, a_{i|\boldsymbol{V}|}[t]\}$ by solving

$$\begin{aligned}
&\min_{a_{ij}[t]} \sum_{j \in \boldsymbol{V}} a_{ij}[t] \cdot (Q_i[t] + (H_i[t] - \theta_i)\, p_{ij}[t]) \\
&\text{s.t. } 0 \le a_{ij}[t] \le A_{ij}[t]
\end{aligned} \tag{21}$$

The optimal solution thus reduces to a simple threshold rule:

$$a_{ij}[t] = \begin{cases} 0, & \text{if } Q_i[t]/p_{ij}[t] + H_i[t] > \theta_i, \\ A_{ij}[t], & \text{if } Q_i[t]/p_{ij}[t] + H_i[t] \le \theta_i. \end{cases} \tag{22}$$

*Remark:* As for such threshold-based admission strategy, $Q_i[t]/p_{ij}[t] + H_i[t]$ can be viewed as equivalent threat situation involving host security state and attack risk level. When threat situation is no larger than $\theta_i$, newly detected attack event $j$ will be admitted into system with the increase of attack level $A_{ij}[t]$ and risk level $p_{ij}[t]$. But $j$ will be rejected when threat situation exceeds $\theta_i$ (i.e., severe host security state or high-risk attack). The intuitive behind no admission is that current risk arrivals go beyond host response capability, and the best way to avoid security state worsening is to seek

[6] We use $\boldsymbol{a}_i[t]$ to denote the set of newly arrived attack events associated with $a_i[t]$. The same is with $\boldsymbol{r}_i[t]$ and $r_i[t]$, $\boldsymbol{d}_i[t]$ and $d_i[t]$.

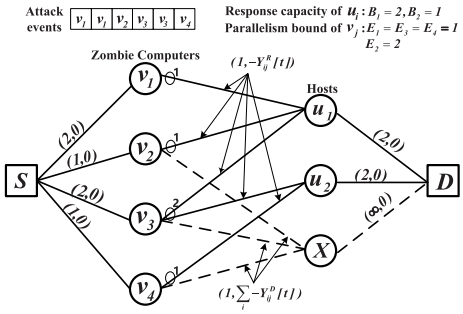



Fig. 3. Response problem interpreted under MCMF.

for more advanced defense techniques to resolve it. Compared to low-risk attack events, high-risk attacks are less likely to be admitted.

*2) Security-Aware Defense Control Policy:* Once admitted into attack queue $Q_i$, attack event $j \in \boldsymbol{N}_i[t]$ next will be responded or dropped. The defense control policy determines which host-attack pairs to drop or resolve with what resource allocation proportion to maximize the last two terms of the RHS of (20), where security-awareness is fully explored by highlighting high-risk attack events.

The defense problem is, in essence, a Minimum Cost Maximization Flow (MCMF) problem [26] shown in Fig. 3. Taking defense system illustrated in Fig. 2 for example, we create a source node $S$ and connect it to all zombie nodes $\{v_1, v_2, v_3, v_4\}$, and create a destination node $D$ and connect all host nodes $\{u_1, u_2\}$ and the specially created drop node $X$ to $D$. We add an edge from $v_j$ to $X$ if there exists at least one associated host $i$ with positive dropping weight $Y_{ij}^D[t]$ (to be specified later), and add an edge from $v_j$ to $u_i$ if response weight $Y_{ij}^R[t] > 0$ (to be specified later). We further assign one (capacity, cost) pair to each edge. The capacity from $S$ to each zombie node is the number of launched attack events with cost 0. Since each zombie computer can launch at most one attack towards a host, the capacity for each host-zombie edge is 1 with cost $-Y_{ij}^R[t]$. Consider all dropped attack events via the same zombie node as a whole, and the capacity from each zombie node to $X$ is 1 with cost $\sum_{i\in\boldsymbol{N}_j} -Y_{ij}^D[t]$. The capacity from a host to $D$ is host response capacity $B_i[t]$ with cost 0. The capacity and cost for drop-destination edge are $\infty$ and 0. In addition, each zombie node is labeled with weight $E_j$ to accommodate parallelism bound. The desired defense control policy involves the following two phases.

*a) Attack dropping:* Since dropping decisions $\boldsymbol{d}_i[t] = \{d_{i1}[t], \cdots, d_{i|\boldsymbol{N}_i[t]|}[t]\}$ of different hosts are independent from each other, we can determine them in a fully distributed manner by solving

$$\begin{aligned}\max_{d_{ij}[t]} \sum_{j\in\boldsymbol{N}_i[t]} & d_{ij}[t](\sigma_j(Q_i[t]+Z_i[t]+(eH_i[t]-\theta_i) \\ & \times ep_{ij}[t])-V\alpha_j) \\ \text{s.t.} \sum_{j\in\boldsymbol{N}_i[t]} & \sigma_j d_{ij}[t] \le d_i^{\max}, \quad d_{ij}[t]\in\{0,1\} \end{aligned} \tag{23}$$

Actually, the above maximization problem is equivalent to the typical Maximum-Weight Matching (MWM) problem. Denote the weight of decision variable $d_{ij}[t]$ as $Y_{ij}^D[t] = \sigma_j\ (Q_i[t] + Z_i[t] + (eH_i[t] - \theta_i)ep_{ij}[t]) - V\alpha_j$, which is determined for any $j \in N_i[t]$. The optimal solution for (23) would prefer to make $d_{ij}[t]$ with larger positive weight as big as possible, especially under constraint $\sigma_j d_{ij}[t] \le d_i^{\max}$. Locate the optimal attack event $j^* = \arg\max_{j\in\boldsymbol{N}_i[t]} Y_{ij}^D[t]$ for each host, where ties are broken randomly. There are two cases: (1) If $Y_{ij^*}^D[t] \le 0$, objective function in (23) is negative for all attacks, suggesting large dropping penalty and good security state. In this case, host $i$ has enough security resources and there is no need to drop any attack. Set $d_{ij}[t] = 0$ for all $j \in \boldsymbol{N}_i[t]$. (2) If $Y_{ij^*}^D[t] > 0$, current security state goes beyond host response capacity and penalty for dropping $j^*$ is relatively low. This attack should be dropped, otherwise it cannot receive prompt response, further aggravating system security state. The optimal solution for (23) is to set $d_{ij^*}[t]$ with larger positive weight as 1 under allowable dropping constraint, i.e., $\sigma_j d_{ij^*}[t] \le d_i^{\max}$. Then perform the dropping process on $Q_i[t]\backslash j^*$ similarly if the leftover allowable dropped loads $d_i^{\max} - \sigma_{j^*} d_{ij^*}[t] > 0$, otherwise end dropping process.

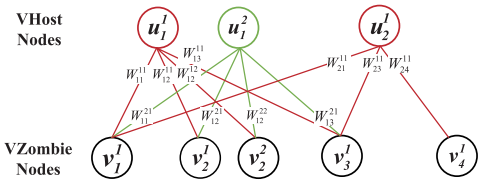



Fig. 4. An example of attack graph with two hosts and four zombie computers.

*b) Risk response:* For each host, decisions on $\boldsymbol{r}_i[t] = \{r_{i1}[t], \cdots, r_{i|\boldsymbol{N}_i[t]|}[t]\}$ and $\boldsymbol{b}_i[t] = \{b_{i1}[t], \cdots, b_{i|\boldsymbol{N}_i[t]|}[t]\}$ can be made by solving

$$\begin{aligned}\max \sum_{j\in\boldsymbol{N}_i[t]} & r_{ij}[t](\sigma_j(Q_i[t]+Z_i[t]+(eH_i[t]-\theta_i)ep_{ij}[t]) \\ & + V\log(p_{ij}[t]b_{ij}[t]) - V\sigma_j c_i[t]) \\ \text{s.t.} \sum_{j\in\boldsymbol{N}_i[t]} & r_{ij}[t] \le B_i, \quad \sum_{i\in\boldsymbol{N}_j} r_{ij}[t] \le E_j, \quad r_{ij}[t]\in\{0,1\} \\ \sum_{j\in\boldsymbol{N}_i[t]} & b_{ij}[t]r_{ij}[t] \le 1, \quad b_{ij}[t] \ge 0. \end{aligned} \tag{24}$$

Such problem, compared to dropping issue, is rather more complicated. It's actually a two-stage decision problem, where in the first stage, the defender determines which attack events towards which hosts to resolve, and with this information, then determines how much resources are allocated to target attack events. Therefore, one feasible solution is backward induction, i.e., first to convert it to a pure resource allocation problem, and on that basis, to address host-attack assignment problem.

*i) The second stage problem:* We first consider resource allocation in the second stage, where host-attack assignment $\boldsymbol{r}_i[t]$ is taken as a given parameter. The total number of resolved attack events towards host $i$ is $n_i[t] = \sum_{j\in\boldsymbol{N}_i[t]} r_{ij}[t]$. The goal of this stage is to find the optimal allocation or proportional fair share $\boldsymbol{b}_i[t] = \{b_{i1}[t], \cdots, b_{in_i[t]}[t]\}$ such that

$$\begin{aligned}\max_{b_{ij}[t]} & \sum_{j\in\{k|r_{ik}[t]=1,\forall k\in\boldsymbol{N}_i[t]\}} r_{ij}[t]\log(p_{ij}[t]b_{ij}[t]) \\ \text{s.t.} & \sum_{j\in\{k|r_{ik}[t]=1,\forall k\in\boldsymbol{N}_i[t]\}} b_{ij}[t]r_{ij}[t] \le 1, \quad b_{ij}[t]\in[0,1]. \end{aligned} \tag{25}$$

The following proposition suggests the optimal resource allocation among target attack events.

*Proposition 3:* For any host $i$, the optimal security resource allocation is equal allocation, i.e., $b_{ij}[t] = 1/n_i[t], \forall j \in \{k|r_{ik}[t] = 1, k \in \boldsymbol{N}_i[t]\}$.

*ii) The first stage problem:* With resource allocation information $\boldsymbol{b}_i[t]$, the goal of this stage is to determine an optimal host-attack assignment $\boldsymbol{r}[t] = \{r_{ij}[t] | i \in \boldsymbol{U}, j \in \boldsymbol{N}_i[t]\}$ that

$$\max_{r_{ij}[t]} \sum_{i\in\boldsymbol{U}} \sum_{j\in\boldsymbol{N}_i[t]} r_{ij}[t](\sigma_j(Q_i[t]+Z_i[t] + (eH_i[t]-\theta_i)ep_{ij}[t] - Vc_i[t]) + V\log\frac{p_{ij}[t]}{n_i[t]})$$
$$\text{s.t.} \sum_{j\in\boldsymbol{N}_i[t]} r_{ij}[t] \le B_i, \quad \sum_{i\in\boldsymbol{N}_j} r_{ij}[t] \le E_j, \quad r_{ij}[t] \in \{0,1\}. \tag{26}$$

Such assignment problem is actually a variation of MWM problem, where $Y_{ij}^R[t] = \sigma_j(Q_i[t] + Z_i[t] + (eH_i[t] - \theta_i) ep_{ij}[t] - Vc_i[t]) + V\log\frac{p_{ij}[t]}{n_i[t]}$ can be viewed as the weight of decision $r_{ij}[t]$, showing the effectiveness of resolving attack $j$ in protecting host $i$. Compared with dropping problem in (23), the only difference is that the weight $Y_{ij}^R[t]$ is not constant and related with assignment decision $r_{ij}[t]$ since $n_i[t] = \sum_{j\in\boldsymbol{N}_i[t]} r_{ij}[t]$. The dynamics in $n_i[t]$ further increase the difficulty in guaranteeing solution optimality, making traditional approaches to MWM no longer applicable.

Based on MCMF in Fig. 3, we first construct the bipartite response graph $\mathcal{G}^{\text{Response}} = (\mathcal{U}', \mathcal{V}', \mathcal{E}')$ shown in Fig. 4. To accommodate defense budget and attack parallelism constraints, we introduce $B_i$ virtual host (VHost) nodes $u_i^1, u_i^2, \cdots, u_i^{B_i}$ in the first vertex set $\mathcal{U}'$ for each host $i \in \boldsymbol{U}$, and $E_j$ virtual zombie (VZombie) nodes $v_j^1, v_j^2, \cdots, v_j^{E_j}$ in the second vertex set $\mathcal{V}'$ for each zombie computer $j \in \boldsymbol{V}$. We add an edge $(u_i^k, v_j^l) \in \mathcal{E}'$ between $u_i^k$ and $v_j^l$, $k \in \{1, \cdots, B_i\}, l \in \{1, \cdots, E_j\}$, with the weight of $W_{ij}^{kl}[t] = W_{ij}[t] + V\log\frac{(k-1)^{k-1}}{k^k}$, where $W_{ij}[t] = \sigma_j(Q_i[t] + Z_i[t] + (eH_i[t] - \theta_i)ep_{ij}[t] - Vc_i[t]) + V\log p_{ij}[t]$.

*Proposition 4:* The host-attack assignment optimization problem in (26) is equivalent to the MWM problem based on response graph $\mathcal{G}^{\text{Response}}$ under proportional fairness guarantee.

*Remark:* One well-known approach to MWM is distributed auction algorithm, but complex host-attack interactions make applying the general solution directly less desirable [27]–[28]. Note that $W_{ij}^{lk}[t]$ consists of $W_{ij}[t]$ and $V\log\frac{(k-1)^{k-1}}{k^k}$, where the former captures utility achieved from resolving $j$, and the latter is penalty on the number of resolved attacks since defense efficiency is lowered with the reduction of allocated resources. These two parts can be maintained separately by zombie computers and hosts, where risk heterogeneity and dynamics in $n_i[t]$ can be addressed. By exploiting this structure, we imagine a virtual auction market, where zombie computers bid for response chances provided by associated hosts. For each slot $t$, we develop a distributed auction-based assignment algorithm, concluded in Algorithm 1, involving the following steps, where references to $t$ are dropped to facilitate expression.

- *Initialization:* Each host generates VHost nodes $u_i^k$ with price $\pi_i^k = -V\log\frac{(k-1)^{k-1}}{k^k}$. Each zombie computer generates VZombie nodes $v_j^l$ and measures the utility $W_{ij}$ achieved from hosts. VHost and VZombie nodes are initially unassigned.

**Algorithm 1** Host-Attack Assignment Algorithm

**Initialize** Assignment $\mathbf{R} \leftarrow \mathbf{0}$, VZombie set $S_j \leftarrow \{v_j^l\}$, VHost set $S_i \leftarrow \{u_i^k\}$, VHost price $\pi_i^k \leftarrow -V\log\frac{(k-1)^{k-1}}{k^k}$;
Calculate response utility $W_{ij}$ for each host-zombie pair;
**while** *Assignment not finalized* **do**
  $\pi_i \leftarrow \min_k \pi_i^k$, $k_i^* \leftarrow \arg\min_k \pi_i^k$;
  ∘ Phase 1: Bidding. For each zombie computer $j$,
  **if** *VZombie node* $v_j^l$ *not assigned* **then**
    $\hat{l}_j \leftarrow \min_l l$;
    **for** *all hosts* $i$ *under* $j$*'s attack* **do**
      $\rho_{ij} \leftarrow W_{ij} - \pi_i$, $i^* \leftarrow \arg\min_i \rho_{ij}$,
      $\rho_j^* \leftarrow \max_i \rho_{ij}$, $\rho_j' \leftarrow \max_{i\neq i_j} \rho_{ij}$;
    Submit $v_j^{\hat{l}_j}$'s bid $bid_j \leftarrow \rho_j^* - \rho_j' + \varepsilon$ to host $i^*$;
  ∘ Phase 2: Assignment. For each host $i$,
  **if** *collects bids from potential zombie computers* **then**
    $win_j \leftarrow \arg\max_j bid_j$,
    Assign VHost $k_i^*$ to attack $win_j$'s VZombie $\hat{l}_{win_j}$ and remove $k_i^*$ and $\hat{l}_{win_j}$ from $S_i$ and $S_j$;
    $\pi_i^{k_i^*} \leftarrow \pi_i^{k_i^*} + bid_{win_j}$;
**return R**;

- *Auction:* This process can be divided into multiple iterations. At the beginning of each iteration, host $i$ announces price $\pi_i = \min_k \pi_i^k$ to zombie computers. In bidding phase, each zombie computer chooses target VZombie $\hat{l}_j = \min l$ and calculates margin $\rho_{ij} = W_{ij} - \pi_i$ achieved from associated hosts. Locating host $i^*$ with the highest margin $\rho_j^*$ and host $i'$ with the second highest margin $\rho_j'$, the zombie submits bid $bid_j = \rho_j^* - \rho_j' + \varepsilon$ to host $i^*$ or one of them randomly in case of tie. In assignment phase, each host picks VZombie node with the highest bid as winner and removes the corresponding VHost and VZombie nodes from alternative sets. The price of VHost nodes is raised by the highest bid. A new iteration starts after VHost nodes announce the new assignment.
- *Termination:* The auction process terminates when there is no change in assignment.

*Theorem 1 (Finite Termination [26]):* Let $E_a = \max_j E_j$. The distributed auction-based assignment algorithm terminates with a feasible host-attack assignment which is within $\varepsilon E_a |\boldsymbol{V}|$ of being optimal for any positive $\varepsilon$.

*Remark:* Since the price for any assigned VHost node increases by at least $\varepsilon$ in each iteration, the maximum number of iterations for bidding is upper bounded by $\frac{E_a}{\varepsilon} \cdot \max\{\sigma_j(Q_i[t] + Z_i[t] + (eH_i[t] - \theta_i)ep_{ij}[t] - Vc_i[t]) + V\log p_{ij}[t]\}$. Accordingly, the tradeoff between convergence time and response efficiency can be achieved by tuning the value of $\varepsilon$.

Furthermore, we summarize our complete defense algorithm to solve one-shot minimization problem (20) in Algorithm 2,

### C. Queueing Performance Analysis

*Theorem 2 (Bounded Queues):* Suppose $0 \le \epsilon_i \le d_i^{\max}$, $\alpha_j > \max_{i\in\boldsymbol{U}} \sigma_j c_i[t]$. If defense decisions and security state

**Algorithm 2** Dynamic Defense Algorithm At Slot $t$

```
∘ Risk Detection and Admission Control
Implement detection scheme to obtain newly arrived attack set N^d with type θ_j = (t_j, ω_j, σ_j, N_j), ∀j ∈ N^d;
for Each detected attack j ∈ N^d do
    for Each host i ∈ N_j do
        Determine admission decisions a_ij[t] by (22)
∘ Risk defense control
Apply Algorithm 1 to obtain host-attack assignment R
for Each host i ∈ U do
    R_i[t] ← {j | r_ij[t] = 1, ∀j ∈ N_i[t]};
    for Each assigned attack j ∈ R_i[t] do
        Determine resource allocation b_ij[t] using (25)
    Determine dropping decisions d_ij[t] for each attack j ∈ N_i[t] by solving (23)
∘ Security State Update
for Each host i ∈ U do
    for Each attack j ∈ N_i[t] do
        p_ij[t] ← e · p_ij[t]
    Update Q_i[t], Z_i[t], H_i[t] according to (4), (14), (17).
```

updates are done by Algorithm 2 with $V > 0$, we have

$$Q_i[t] \le Q_i^{\max} \triangleq \theta_i p^{\max} + A^{\max} \tag{27}$$

$$Z_i[t] \le Z_i^{\max} \triangleq V\alpha^{\max} + e\theta_i p^{\max} + \epsilon_i \tag{28}$$

$$H_i[t] \le H_i^{\max} \triangleq \frac{e\theta_i \sigma^{\max} p^{\max} + V\alpha^{\max}}{e^2 \sigma^{\min} p^{\min}} + \beta_i^{\max} + (a_i')^{\max} \tag{29}$$

where $\beta_i^{\max} = (e-1)Q_i^{\max} p^{\max}$ and $(a_i')^{\max} = A^{\max} p^{\max}$.

*Theorem 3 (No Underflow Condition):* For any perturbation parameter $\theta_i$, if it satisfies $\theta_i \ge 2e^2 p^{\max}(B_i + d_i^{\max}) - \frac{V\sigma^{\min} c_i[t]}{ep^{\max}}$, attack queue $Q_i$ will not suffer from underflow at slot $t$.

*Remark:* Theorem 3 illustrates how to leverage $\theta_i$ to guarantee no underflow condition. Intuitively, the larger $d_i^{\max}$ is (or the less response cost $c_i[t]$ is), the larger the lower bound of $\theta_i$ will become. Taking Fig. 2 as an example, there are 2 hosts under the attack of 4 zombie computers. Suppose host $u_1$ contains more valuable data than host $u_2$ that needs to be protected. Obviously, the defender is willing to invest more budget to respond to high-risk attacks. Under the same attack, host $u_1$ usually suffers from much larger loss than host $u_2$. Hence the tolerable risk level of host $u_1$ denoted by $\theta_1$ should be set smaller than that of $u_2$ denoted by $\theta_2$.

*Theorem 4 (No Dropping Condition):* There is no attack dropping for any host if the total response rate $\sum_{i\in U} r_i[t]$ satisfies $|\boldsymbol{U}|\sigma^{\max}(A^{\max} + \epsilon^{\max} + \frac{V\log e}{\sigma^{\max}} + (ep^{\max})^2(\frac{(e-1)Q^{\max}}{\sigma^{\min}} + A^{\max})) \le \frac{\Gamma-\sigma^{\max}}{\Gamma}\sigma^{\min}((2+e^3(p^{\min})^2)\sum_{i\in U} r_i[t] + e(e-1)\theta^{\min} p^{\min})$, where $\epsilon^{\max} = \max \epsilon_i^{\max}$ and $Q^{\max} = \max Q_i^{\max}$.

*Theorem 5 (Profit Optimality):* Under non-preemptive scheduling, we group $\Gamma$ time slots into a refresh frame, where $\Gamma > \sigma^{\max}$. Suppose unit response cost $c_i[t] \in [c_i^{\min}, c_i^{\max}]$ is ergodic processes. Under no attack dropping condition, there exists some $\delta > 0$ such that the time-average system profit achieved by Algorithm 2 is within a constant gap from the offline optimal system profit $f^{\frac{(1+\delta)\Gamma}{\Gamma-\sigma^{\max}}}$, i.e.,

$$\begin{aligned}
&\lim_{\eta\to\infty} \frac{1}{\eta\Gamma}\sum_{n=0}^{\eta-1}\sum_{t=n\Gamma}^{(n+1)\Gamma-1} \mathbb{E}\{f[t]\} \\
&\ge f^{\frac{(1+\delta)\Gamma}{\Gamma-\sigma^{\max}}} - \frac{(\Gamma-\sigma^{\max})(\Gamma-\sigma^{\max}-1)B_3}{2\Gamma V} \\
&\quad - \frac{B_1}{V} - \frac{\Gamma-1}{2V}\sum_{i\in U}(\epsilon_i^2 + (A^{\max})^2 \\
&\quad + p^{\max}(\frac{(e-1)Q^{\max}}{\sigma^{\min}} + A^{\max})(\theta_i(1-e) + p^{\max}A^{\max})) \\
&\quad - \frac{(\Gamma-\sigma^{\max})(\Gamma-\sigma^{\max}-1)}{2\Gamma}\sum_{i\in U}\sigma^{\max}B_i(c_i^{\max} - c_i^{\min}) \\
&\quad - \frac{(\sigma^{\max})^2}{\Gamma}\sum_{i\in U} B_i(c_i^{\max} + \log\omega^{\max} + \frac{ep^{\max}\theta_i}{V})
\end{aligned} \tag{30}$$

where $B_3 = \sum_{i\in U} B_i\sigma^{\max}(A^{\max} + \epsilon^{\max} + 2\sigma^{\max}B_i + (ep^{\max})^2 \cdot (\frac{(e-1)Q^{\max}}{\sigma^{\min}} + A^{\max} + e\sigma^{\max}B_i) + V\log\omega^{\max})$.

Threat knowledge is often assumed to be detected accurately in traditional risk management. Consider a realistic scenario. What if defense strategies are made based on detected risk levels (i.e., $\hat{\omega}_j$) and attack levels (i.e., $\hat{Q}_i[t]$, $\hat{Z}_i[t]$, $\hat{H}_i[t]$) that differ from actual threat knowledge (i.e., $\omega_j$, $Q_i[t]$, $Z_i[t]$, $H_i[t]$)?

*Theorem 6 (Profit Optimality With Detection Errors):* Suppose there exists constants $\xi^{\omega}$, $\xi^{Q}$ such that at all $t$, $|\hat{\omega}_j - \omega_j| \le \xi^{\omega}$, $|\hat{X}_i[t] - X_i[t]| \le \xi^{Q}$ for $Q_i[t]$, $Z_i[t]$, $H_i[t]$ hold. We have

$$\begin{aligned}
&\lim_{\eta\to\infty} \frac{1}{\eta\Gamma}\sum_{n=0}^{\eta-1}\sum_{t=n\Gamma}^{(n+1)\Gamma-1} \mathbb{E}\{f[t]\} \\
&\ge f^{\frac{(1+\delta)\Gamma}{\Gamma-\sigma^{\max}}} - \frac{(\Gamma-\sigma^{\max})(\Gamma-\sigma^{\max}-1)B_3}{2\Gamma V} \\
&\quad - \frac{B_1'}{V} - \frac{\Gamma-1}{2V}\sum_{i\in U}(\epsilon_i^2 + (A^{\max})^2 \\
&\quad + p^{\max}(\frac{(e-1)Q^{\max}}{\sigma^{\min}} + A^{\max})(\theta_i(1-e) + p^{\max}A^{\max})) \\
&\quad - \frac{(\Gamma-\sigma^{\max})(\Gamma-\sigma^{\max}-1)}{2\Gamma}\sum_{i\in U}\sigma^{\max}B_i(c_i^{\max} - c_i^{\min}) \\
&\quad - \frac{(\sigma^{\max})^2}{\Gamma}\sum_{i\in U} B_i(c_i^{\max} + \log\omega^{\max} + \frac{ep^{\max}\theta_i}{V})
\end{aligned} \tag{31}$$

where $B_1' = B_1 + \sum_{i\in U}(\xi^{Q}(\epsilon_i + \theta_i(1-e) + A^{\max}(1+p^{\max})) + B_i(2\sigma^{\max} + ep^{\max}) + d_i^{\max}(2 + ep^{\max})) + \xi^{\omega}e^{D_{\max}}(A^{\max}(H_i^{\max} + \xi^{Q} + \theta_i) + (B_i\sigma^{\max} + d_i^{\max})(e^2H_i^{\max} + e^2\xi^{Q} + e\theta_i))$.

*Remark:* Comparing (30) with (31), we observe there exists a gap $\frac{B_1'-B_1}{V}$ between the optimal system profits in cases of accurate knowledge and detection errors. As expected, control parameter $V$ can be carefully leveraged in online scheduling to further reduce the effect of detection errors. Especially when $V$ goes to infinity, the optimal profit gap tends to 0. It is indicated that with inaccurate detected threat information, larger $V$ is desired to achieve the same system profit as with accurate knowledge, but may result in higher queue backlogs as shown

in Theorem 2. Thus, our mechanism is robust to detection errors, but at the expense of decreased stability.

## IV. Applying Security-Aware Defense Mechanism to the Anomaly Detection System

The implementation of our defense mechanism is driven by threat intelligence. Due to diversity of attack means, there is no standard method for acquiring threat knowledge, and many threat intelligence platforms such as FireEye have managed it in their own ways [9]. Here we apply our mechanism to a practical anomaly detection system deployed by Cyberxingan Technology Ltd. just as an initial attempt for defending APTs.

- *Feature Extraction and Fusion:* Stealthy APTs tend to exhibit no evident character but always differentiate themselves from normal subjects in feature distribution, making it hard to identify anomalies only by volume features. To capture dispersal degree, entropy features are introduced as the basis of the detection scheme design [29]. Based on collected two weeks' worth of log data from widely deployed security devices, a 7-tuple <t,pkts, bytes, HsIP, HdIP, HsPort, HdPort> is particularly extracted, where the first two volume features denote the size and number of packets at time $t$, and the other entropy features denote entropy information about source address, destination address, source port and destination port. To address the feature redundancy issue caused by high dependency among such features and detection errors, we incorporate principle component analysis-based feature fusion scheme into outlier detection, mapping data onto a new set of principal components. The principal components are ordered by the amount of energy in the data they capture.
- *Density-Based Clustering and Attack Identification:* Several recent works have been devoted to outlier mining and among them, unsupervised $K$-means clustering is promising to deal with difficulty in acquiring label data [30]-[32]. In view of its inherent drawbacks, e.g., predetermining cluster centroid number or falsely clustering caused by unbalanced density, we further adopt density-based clustering for outlier mining, which can identify centroids with high density and large distance from others. Consider a set of time-series data points $S = \{x_m\}_{m=1}^{N}$ with $I_S = \{1, \cdots, N\}$. Under cut-off kernel model, we define point density as $\rho_m = \sum_{n \in I_S \setminus \{m\}} \chi(d_{mn} - d_c)$, where $d_c$ is cutoff distance, $d_{mn}$ is distance between $x_m$ and $x_n$, $\chi(x)$ equals 1 if $x < 0$ and 0 otherwise. Denote $I_m^S = \{k | k \in I_S, \rho_k > \rho_m\}$. The minimum distance of $x_m$ from points with higher density, $d_m^{\min}$, equals $\min_{n \in I_S}\{d_{mn}\}$ if $I_m^S \neq \emptyset$, and $\max_{n \in I_S}\{d_n^{\min}\}$ if $I_m^S = \emptyset$. From $\rho_m$ vs $d_m^{\min}$ figure, we find the majority of points are close to coordinate, and only few points are of high $\rho_m$ and $d_m^{\min}$, which are potential to be centroids. $x_m$ is in cluster $A$ only if it has a minimum distance from $A$'s centroid. Let $d_m$ denote the distance between $x_m$ and its centroid. Intuitively, points with large distance and small density are more likely to be outliers. We assign each point one outlier score $O_m = O(d_m, \rho_m)$ to characterize risk seriousness of the system at that point. Joint with security experts' experience, retrospective analysis is performed on detected outliers and then high-risk attacks are identified.
- *Threat Information Acquisition:* Backed by threat intelligence, threat information $\theta_j = (t_j, \omega_j, \sigma_j, \boldsymbol{N}_j)$ for each detected attack can be acquired. That is, which hosts (i.e., $i \in \boldsymbol{N}_j$) are compromised by attack $j$ at which time (i.e., $t_j$) at which attack and risk levels (i.e., $\sigma_j$ and $\omega_j$). Based on how long it takes attacker to compromise hosts, security experts first determine attack load $\sigma_j$, i.e., the number of slots requested for response. Risk score $\omega_j$ here is established on outlier feature, capturing attack security state from the perspective of risk seriousness. Since risk seriousness escalates over time, we suppose there is a one-to-one correspondence between $\omega_j$ and detection time $t_j$. The earlier attacks are detected, the lower risk score will be. Hence we capture risk score by monotone increasing risk function $h(\cdot)$, i.e., $\omega_j = h(O_m, t_j)$.

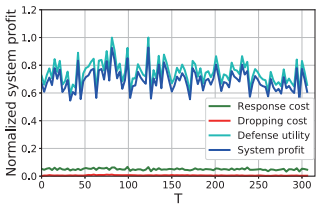
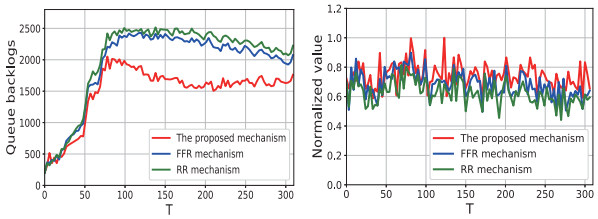



Fig. 5. System profit, defense utility, response cost and dropping cost vs. $T$.




(a) (b)

Fig. 6. (a) Queue backlogs vs. $T$. (b) System profit vs. $T$.

## V. Simulation Results

To show the applicability and effectiveness of our mechanism, we experiment with a real-world dataset with acquired threat knowledge introduced in Sec. IV. Attack events within this dataset include scanning, key-compromise impersonation, zero-day exploit, and so on. To accommodate diversity in attack security state, attack load and risk score are adjusted to $\sigma_j \in [1, 5]$ and $\omega_j \in [1, 6]$. The attacker schemes to attack $|\boldsymbol{U}| = 214$ hosts via zombie computers with different IP addresses. Host heterogeneity in response capacity is leveraged from two aspects, i.e., dynamic resources provisioned for left-over attack events and resources allocated to attacks newly selected to respond under defense budget uniformly distributed in [1, 10]. We implement our mechanism for $T = 310$ time slots with parameters $D_{\max} = 10$ and compare it with two other baseline mechanisms: (1) *FIFO-Based Response (FFR) Mechanism [33]*: For any host, attack events are processed in First-In First-Out (FIFO) order; (2) *Random Response (RR) Mechanism [34]*: Attack events queued at any host are processed randomly.

Fig. 5 demonstrates the variance of system profit, defense utility, response cost and dropping cost over time. We observe a stable system profit is achieved by our defense mechanism. The dropping cost approaching zero indicates our mechanism can ensure high response rate, which is consistent with our analysis on no attack dropping shown in Sec. III.

We illustrate the comparison of security risk and system profit in terms of $T$ shown in Fig. 6(a) and Fig. 6(b). As

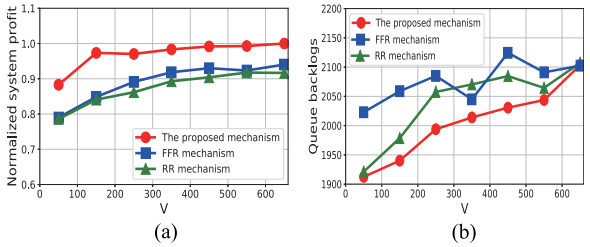



Fig. 7. (a) Average system profit vs. $V$. (b) Average queue backlogs vs. $V$.

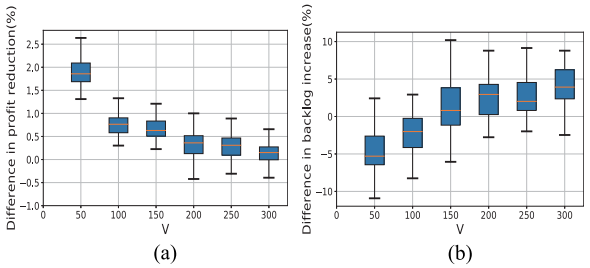



Fig. 8. (a) Differences in system profit reduction vs. $V$. (b) Differences in queue backlogs increase vs. $V$.

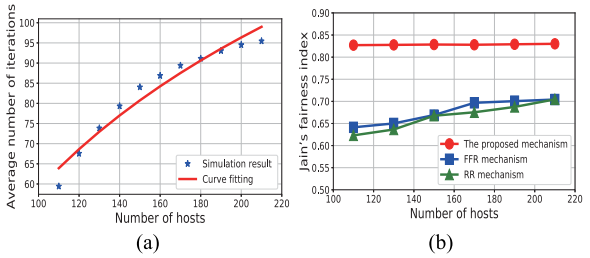



Fig. 9. (a) Average number of iterations vs. Host number. (b) Fairness among attack events.

expected, our mechanism is superior to others in alleviating risk pressure while guaranteeing high system profit. The intuitive is that attacks with high-risk score or early detection time, under dropping control policy, are more likely to be dropped when current attack state goes beyond host response capability. In our mechanism, high-risk attacks are given priority, which is promising to avoid security state spinning out of control and reduce incidence of attack dropping. High response completion is further ensured by adjusting host response capacity. While in FFR mechanism, attacks are resolved in FIFO order and response chances are more likely to be seized by low-risk attacks with early detection time and large attack loads, increasing the risk of high-risk attacks' being dropped. High attack and risk levels are reflected in increased queue backlogs. Compared to RR mechanism, FFR can alleviate risk pressure posed by attack events with early detection time, guaranteeing a shortened queue length. Thus, queue backlog in our mechanism is approaching to a steady state with the highest speed of convergence. Under priority-based response, large amount of high-risk attacks are potential to be resolved before maximal response delay, greatly reducing incidence of attack dropping. High system profit can thus be achieved in our mechanism.

We illustrate the impact of control parameter $V$ on system stability. Fig 7(a) shows the average system profit converges quickly as $V$ grows, and the highest profit is achieved by our mechanism. From Fig 7(b), we observe our mechanism is superior to other mechanisms in alleviating risk seriousness, where all average queue backlogs grow linearly with $V$. Combining these two figures coincides with our analysis about Theorems 2 and 5, where $V$ can be leveraged to resolve the tradeoff between system profit and security risks.

We try to explore the influence of detection errors on defense efficiency. For each arrived attack, we add a random detection error ($\pm 50\%$, uniformly distributed) to attack load and risk score it brings. We conduct our mechanism on such error dataset. For any attack, exact threat information is known by defender only when it gets resolved. We illustrate the differences in average system profit reduction and average queue backlog increase due to injected detection errors with varying $V$, where results on original datasets without errors serve as baseline. Fig 8(a) shows for all $V$ we experiment with, the difference in profit reduction is between $-0.45\%$ and $2.7\%$. As shown in Fig 8(b), detection errors result in increased queue backlogs within the range of $-12\%$ to $10\%$. These results indicate our mechanism is robust to detection errors by leveraging $V$, but at the expense of decreased stability.

Fig 9(a) shows average number of auction iterations required under our assignment algorithm for different system sizes. For system with $|\boldsymbol{U}| = 214$, the algorithm takes an average of 95 iterations each slot to converge to an optimal host-attack assignment. Through curve fitting, we obtain the curve of average number of iterations, i.e., $124.9\log|\boldsymbol{U}| - 191.1$, suggesting average number of iterations increases as $O(\log|\boldsymbol{U}|)$. We study Jain's fairness index [27] for average response rate among attack events, which is defined as $J = \frac{1}{|\boldsymbol{U}|}\sum_{i=1}^{|\boldsymbol{U}|}\frac{\left(\sum_{j=1}^{|\boldsymbol{V}|}\sigma_j\overline{r}_{ij}\right)^2}{|\boldsymbol{V}|\sum_{j=1}^{|\boldsymbol{V}|}(\sigma_j\overline{r}_{ij})^2}$ with $\overline{r}_{ij} = \limsup_{t\to\infty}\frac{1}{t}\sum_{\tau=0}^{t-1}\mathbb{E}\{r_{ij}[t]\}$. As shown in Fig 9(b), we observe our mechanism has the highest fairness index. The reason that other mechanisms are poor in fairness is that attack events are resolved in FIFO or random orders without considering diversity of risk seriousness. It is more likely that response chances are seized by certain attacks with large attack loads, increasing the risk of high-risk attacks' being dropped when current attack state goes beyond host response capacity. The gap among attack response rates is thus expanded. While our mechanism adjusts response capacity dynamically based on host security state, and gives priority to high-risk attacks in response, guaranteeing much high response rates with a small increase in average delay.

## VI. Related Work

### A. Risk Management

The prominent intelligent tendency in risk management supported by emerging threat intelligence has promoted the resolution of defending against APTs [5]–[9], where the key issue is how to acquire threat knowledge accurately [29]–[32] and assess risks efficiently [35]–[36].

Access to threat knowledge begins with the identification of attack features and potential threat sources, which paves the way for making efficient defense strategies. Veeramachaneni *et al.* [7] combined analysts' experience with machine learning techniques to provide an artificially intelligent solution. Zheng *et al.* [37] proposed a time associative bandit model for optimal timing of security updates, where defender can obtain partial feedback under stealthy attacks via online learning. Although power of machine learning is leveraged, these studies do not consider detection error that might appear in practice, which is our perspective. Zhan *et al.* [17] presented a methodology for predicting attack rates more accurately by accommodating extreme-value phenomenon.

However, this work does not answer the fundamental question in intelligence acquisition: besides attack rate, is there any other salient heterogeneity in host security state, especially with the advances of detection technology? We tackle this open problem by designing an intelligence-driven defense mechanism from perspectives of risk seriousness and attack rate, enhancing defender's ability to respond quickly, decisively and effectively to potential risks.

In practice, threats change over time, as do risks. As expected, there also exists a large body of work for risk assessment by using paradigms like attack graphs and attack trees to capture dynamics of security state. Under flow dependency graph, Rezvani *et al.* [15] considered the provenance and interdependency between hosts and flows to assess risk on network activities. While this work can show good effect for given accurate threat knowledge, there is, however, an aspect of detection error to security planning process. Providing better robustness to detection errors constitutes one of key theoretical contributions of our mechanism. Poolsappasit *et al.* [16] designed a Bayesian attack graph-based risk management framework to assess risks and select hardening measures to maximize resource utilization. But they only focus on centralized case to globally furnish defender with better control of risk assessment, which is insufficient in the face of large networks with intensive attacks and rapidly changing threats. In this paper, we fill the void with a distributed auction-based assignment algorithm to determine minimum-cost response decisions with guaranteed finite termination. Backed by threat intelligence, we harness Lyapunov optimization to explore dynamics in security awareness and enable long-term response.

### *B. Defense Strategy Making*

There has been extensive research attempting to make effective defense strategies.

One line of research focuses on determining which attacks to respond. In this paper, we build a connection between target attack selection and response scheduling design. Wei *et al.* [38] analyzed defensive abilities of three classic queue management algorithms under DDoS attack. Considering SYN flooding attack as an unfair scheduling that gives more chances to attack requests, Jamali and Shaker [33] presented several popular scheduling algorithms, where attack requests under FIFO are served in the order of arrival time. Xu *et al.* [34] modeled the attack-defense interaction as Markov process, where node security state (secure or compromised) is captured by a random variable. However, existing response scheduling works typically separate attack detection from response process, making it hard to keep system security state in control, especially under limited defense budget. To break this barrier, we propose a novel security-aware defense mechanism based on acquired threat intelligence. In particular, the potential benefit of security-awareness is fully explored from perspectives of both attack level and risk level, where high-risk host-attack pairs are given priority in response.

Another line of research focuses on using incentive mechanism to improve the efficiency of defense strategies [39]. Hu *et al.* [40] investigated the joint threats from APT attacker and insiders for one target system resource, and proposed a two-layer differential game model including a defense/attack game between defender and attacker and an information-trading game among multiple insiders. In practical scenarios, however, attacker can simultaneously attack multiple targets. Inspired by prospect theory, Xiao *et al.* [12] developed a dynamic storage defense game, where the protected interval is modeled as part of players' utility functions while ignoring strict resource constraints. For a large attack-intensive system, ignoring such constraints can lead to either resource over- and under-provisioning or revenue loss. Intuitively, multi-target setting together with resource constraints imposes significant challenges in achieving a tradeoff between response efficiency and number of targets. To this end, we construct a virtual auction market to cope with uncertainty in the number of resolved attacks. As a future work, it would be worthwhile to further extend our mechanism to large-scale networks [41]. Under resource constraints, Zhang *et al.* [13] proposed a two-player game model for defending against APTs with asymmetric feedback structure, where attacker can fully observe target states while largely hiding its actions from defender. The major advantage of our work over it is to integrate detection into defense strategy making, and capture intelligent defender's ability to acquire threat knowledge, which are vital to enabling high response efficiency. All of this is producing new challenges for defense mechanism design.

## VII. Conclusion

We provide a Lyapunov-based intelligence-driven security-aware defense mechanism against APTs. Backed by threat intelligence, we develop tolerable risk admission control policy to accommodate host risk tolerance, and further implement security-aware defense control policy, where high-risk host-attack pairs are prioritized over others. Simulations based on real-world dataset validate the effectiveness of our mechanism.

## Appendix A
## Proof of Proposition 1

*Proof:* Let $a_i[t] > 0$ represent attack loads that enter $Q_i$ at any slot $t$, which are actually part of $Q_i[t+1]$. By contradiction, we suppose there exist parts of $a_i[t]$ that remain in $Q_i$ after time $t + D_i^{\max}$. The total departure loads during $D_i^{\max}$ slots following $t$ is at most $Q_i^{\max}$, i.e., $\sum_{\tau=t}^{t+D_i^{\max}} r_i[\tau] + d_i[\tau] \le Q_i^{\max}$. In the subsequent $D_i^{\max}$ slots after $t$, if $Q_i[\tau]$ equals 0, all target attack loads are processed within $D_i^{\max}$ slots; otherwise $Z_i$ has a constant arrival rate $\epsilon_i$, and the same departure rate $r_i[t] + d_i[t]$ as that in $Q_i$. By dynamics of delay-aware virtual queues (14), we obtain for all $\tau \in \{t+1, \cdots, t+D_i^{\max}\}$, $Z_i[\tau + 1] \ge Z_i[\tau] - r_i[\tau] - d_i[\tau] + \epsilon_i$. Summing both sides over $\tau \in \{t+1, \cdots, t+D_i^{\max}\}$ yields $Z_i[\tau + D_i^{\max} + 1] - Z_i[\tau + 1] \ge \epsilon_i D_i^{\max} - \sum_{\tau=t}^{t+D_i^{\max}} r_i[\tau] + d_i[\tau]$. Since $Z_i[t+1] \ge 0$ and $Z_i[t + D_i^{\max} + 1] \le Z_i^{\max}$, we obtain $\sum_{\tau=t}^{t+D_i^{\max}} r_i[\tau] + d_i[\tau] \ge \epsilon_i D_i^{\max} - Z_i^{\max}$. Since $\sum_{\tau=t}^{t+D_i^{\max}} r_i[\tau] + d_i[\tau] \le Q_i^{\max}$, we have $Q_i^{\max} > \epsilon_i D_i^{\max} - Z_i^{\max}$. This implies $D_i^{\max} < \lceil (Q_i^{\max} + Z_i^{\max}) / \epsilon_i \rceil$, contradicting the definition of $D_i^{\max}$ given in (15). Thus the proposition follows. □

## Appendix B
## Proof of Proposition 2

*Proof:* Since $Q_i^2[t+1] - Q_i^2[t] \le (r_i[t] + d_i[t])^2 + a_i^2[t] + 2Q_i[t](a_i[t] - r_i[t] - d_i[t])$, $Z_i^2[t+1] - Z_i^2[t] \le (\epsilon_i - r_i[t] - d_i[t])^2 + 2Z_i[t](\epsilon_i - r_i[t] - d_i[t])$ and $(H_i[t+1] - \theta_i)^2 - (H_i[t] - \theta_i)^2 \le (r_i'[t] + d_i'[t])^2 +$

$(a_i'[t])^2+\beta_i^2[t]-2\beta_i[t](r_i'[t]+d_i'[t])+2(H_i[t]-\theta_i)(a_i'[t]+\beta_i[t]-r_i'[t]-d_i'[t])$, summing the above yields $\Delta(\boldsymbol{\Theta}[t])$, as shown at the bottom of this page.

Adding $-Vf[t]$ to both sides and taking conditional expectation on both sides given $\boldsymbol{\Theta}[t]$, the proposition is proven. □

## APPENDIX C
## PROOF OF PROPOSITION 3

*Proof:* We drop references to $t$ in addressing such allocation issue. Given assignment $\boldsymbol{R}_i$, optimal allocation problem is $\boldsymbol{b}_i^*=\arg\max_{b_{ij}}\sum_{j\in\{k|r_{ik}=1,\forall k\in\boldsymbol{N}_i\}}\log(\omega_j b_{ij})=\arg\max_{b_{ij}}\sum_{j\in\{k|r_{ik}=1,\forall k\in\boldsymbol{N}_i\}}\log b_{ij}+\log\omega_j$, where the second term is a constant about host target attack events. Then this optimal allocation is equivalent to maximizing the geometric mean, i.e., $\max_{b_{ij}}\sum_{j\in\{k|r_{ik}=1,\forall k\in\boldsymbol{N}_i\}}\log b_{ij}\Leftrightarrow\max_{b_{ij}}\frac{1}{n_i}\log(\prod_{j=1}^{n_i}b_{ij})\Leftrightarrow\max_{b_{ij}}\sqrt[n_i]{b_{i1}b_{i2}\cdots b_{in_i}}$. Since geometric mean is no larger than arithmetic mean, we have $\sqrt[n_i]{b_{i1}b_{i2}\cdots b_{in_i}}\le\frac{b_{i1}+b_{i2}+\cdots+b_{in_i}}{n_i}$, where the equality holds if and only if $b_{i1}=\cdots=b_{in_i}$. Hence to maximize overall defense efficiency, $b_{ij}$ should be equal for all attacks, i.e., $b_{ij}=1/n_i$. Thus, the proposition follows. □

## APPENDIX D
## PROOF OF PROPOSITION 4

*Proof:* Let $w_{ij}=\sigma_j(Q_i+Z_i+(eH_i-\theta_i)ep_{ij}-Vc_i)$. The objective of assignment problem for $n_i$ zombie computers is $O_A=\sum_{i\in\boldsymbol{U}}\sum_{j\in\boldsymbol{N}_i}r_{ij}(w_{ij}+V\log\frac{\omega_j}{n_i})=\sum_{i\in\boldsymbol{U}}\sum_{k=1}^{n_i}w_{ii_k}+V\log\frac{\prod_{k=1}^{n_i}\omega_{i_k}}{n_i^{n_i}}$. Denote $O_M$ as the objective of MWM. Consider this problem from the following two aspects:

- $O_M\le O_A$: Construct an assignment in $\mathcal{G}^{\text{Response}}$ where $u_i^k$ is assigned to $v_j^l$. The sum of edge weights $W_{ij}^k$ is $O_M\le\sum_{i\in\boldsymbol{U}}\sum_{j\in\boldsymbol{N}_i}r_{ij}(w_{ij}+V\log\frac{(k-1)^{k-1}\omega_j}{k^k})=\sum_{i\in\boldsymbol{U}}\sum_{k=1}^{n_i}w_{ii_k}+V\log\frac{(k-1)^{k-1}\omega_{i_k}}{k^k}=\sum_{i\in\boldsymbol{U}}\sum_{k=1}^{n_i}w_{ii_k}+V\log\frac{\prod_{k=1}^{n_i}\omega_{i_k}}{k^k}=O_A$.
- $O_M\ge O_A$: Consider the dual problem of MWM, which assigns a non-negative price to each host and finds the minimum price vertex cover in $\mathcal{G}^{\text{Response}}$ [28]. Denote $g(k)=-V\log\frac{(k-1)^{k-1}}{k^k}$, satisfying $g(k)\ge 0$ and $g(k)<g(k-1)$. To avoid negative weights, we set each edge weight $W_{ij}'^k=W_{ij}^k+h_1+h_2\ge 0$, where adjustment factors $h_1=g(n_i)$, $h_2=g(B_i)+|\min_i\{w_{ij}+\log\omega_j\}|$. Hence each zombie computer $j$ is assigned a utility associated with host $i$, $J_{ij}=h_2+w_{ij}+\log\omega_j$, and each $u_i^k$ is assigned a price $\pi_i$, which equals $h_1$ if $k=1$, $h_1-g(k)$ if $1<k\le n_i$ and 0 if $k>n_i$. We obtain $J_{ij}\ge 0$, $\pi_i\ge 0$, $W_{ij}'^k\le J_{ij}+\pi_i$. For any assignment $\boldsymbol{R}$ on $\mathcal{G}^{\text{Response}}$, we have $\sum_{(u_i^k,v_j^l)\in\boldsymbol{R}}W_{ij}'^k\le\sum_{(u_i^k,v_j^l)\in\boldsymbol{R}}(\pi_i+J_{ij})\le\sum_{u_i^k\in\mathcal{U}'}\pi_i+\sum_{v_j^l\in\mathcal{V}'}J_{ij}=\sum_{i\in\boldsymbol{U}}(n_i(h_1+h_2)+\log\frac{\prod_{k=1}^{n_i}\omega_{i_k}}{n_i^{n_i}}+\sum_{k=1}^{n_i}w_{ii_k})$. Since $\mathcal{G}^{\text{Response}}$ is a complete bipartite graph with $n_i\le B_i$, we have $n_i$ edges in MWM. By removing adjustment factors on edge weights, $n_i(h_1+h_2)$, we get $\sum_{(u_i^k,v_j^l)\in\boldsymbol{R}}W_{ij}\le O_A$.

Above all, the equivalence proposition follows. □

## APPENDIX E
## PROOF OF THEOREM 1

*Proof:* Suppose the algorithm does not terminate. At each iteration initiated with one unassigned attack, the assigned host-attack pair number increases by one or remains constant. If termination does not occur, only the latter case can happen, where at least one price $\pi_i$ of VHost increases by $\varepsilon$ and at least one profit $\rho_{ij}$ of VZombie decreases by $\varepsilon$. Hence subset $\boldsymbol{U}_\infty=\{i\in\boldsymbol{U}|\lim_{k\to\infty}\pi_i^k=\infty\}$ of VHost nodes that receive an infinite number of bids, and subset $\boldsymbol{V}_\infty=\{j\in\boldsymbol{V}|\lim_{k\to\infty}\rho_{ij}^k=-\infty\}$ of VZombie nodes that bid infinite times are empty. Then prices $\pi_i$ of VHost nodes in $\boldsymbol{U}_\infty$ must tend to $\infty$. For VZombie node $j\in\boldsymbol{V}_\infty$, profit $\rho_{ij}=\max_{i\in\boldsymbol{N}_j}\{J_{ij}-\pi_i\}$ must tend to $-\infty$. Thus, $J_{ij}-\pi_i$ tends to $-\infty$ for $i\in\boldsymbol{N}_j$. We obtain $\boldsymbol{N}_j\subset\boldsymbol{U}_\infty,\forall j\in\boldsymbol{V}_\infty$, i.e., VZombie nodes in $\boldsymbol{V}_\infty$ can only be assigned to VHost nodes in $\boldsymbol{U}_\infty$. According to $\varepsilon$-CS bidding process, i.e., $W_{ij}-\pi_i\ge\max_{l\in\boldsymbol{N}_j}\{W_{lj}-\pi_l\}-\varepsilon$, $J_{ij}-\pi_i\ge\rho_{ij}-\varepsilon$ for every assigned pair, and each VHost node in $\boldsymbol{U}_\infty$ can only be assigned to a VZombie node from $\boldsymbol{V}_\infty$. In the case of no termination, at least one unassigned VZombie node from $\boldsymbol{V}_\infty$ will be assigned. The number of VZombie nodes in $\boldsymbol{V}_\infty$ is larger than the number of VHost nodes in $\boldsymbol{U}_\infty$, contradicting the existence of a feasible assignment under condition $\boldsymbol{N}_j\subset\boldsymbol{U}_\infty$. Thus, the algorithm must terminate. □

## APPENDIX F
## PROOF OF THEOREM 2

*Proof:* When $t=0$, $Q_i[0]=0<Q_i^{\max}$. Suppose $Q_i[t]\le Q_i^{\max}$ for some slot $t$. We next show it also holds for slot $t+1$ from the following two cases: (1) *Case 1*: When $Q_i[t]\le\theta_i p^{\max}$, $Q_i[t+1]$ will increase at most $A^{\max}$, i.e., $Q_i[t+1]\le Q_i[t]+A^{\max}\le\theta_i p^{\max}+A^{\max}$. (2) *Case 2*: When $Q_i[t]>\theta_i p^{\max}$, $Q_i[t]-\theta_i p_{ij}[t]+H_i[t]p_{ij}[t]>0$. Through admission control policy, $a_{ij}[t]=0$ for all $j$ and no attack will

$$\begin{aligned}\Delta(\boldsymbol{\Theta}[t])=L\{\boldsymbol{\Theta}[t+1]\}-L\{\boldsymbol{\Theta}[t]\}\le&\sum_{i\in\boldsymbol{U}}\frac{(\sigma^{\max}B_i+d_i^{\max})^2+(A^{\max})^2}{2}+Q_i[t](a_i[t]-r_i[t]-d_i[t])\\&+\sum_{i\in\boldsymbol{U}}\frac{\max\{(\sigma^{\max}B_i+d_i^{\max})^2,{\epsilon_i}^2\}}{2}+Z_i[t](\epsilon_i-r_i[t]-d_i[t])\\&+\sum_{i\in\boldsymbol{U}}\frac{((\sigma^{\max}B_i+d_i^{\max})\sigma^{\max}\omega^{\max}e^{1+D_{\max}})^2+(A^{\max}\omega^{\max}e^{D_{\max}})^2}{2}\\&+\sum_{i\in\boldsymbol{U}}\frac{(Q_i^{\max}\omega^{\max}e^{D_{\max}})^2(e^2-e)}{2(\sigma^{\min})^2}+(1-e)\theta_i H_i[t]+\sum_{i\in\boldsymbol{U}}(H_i[t]-\theta_i)a_i'[t]+(\theta_i-eH_i[t])(r_i'[t]+d_i'[t])\end{aligned}$$

be admitted into $Q_i$. We obtain $Q_i[t+1] \le Q_i[t] \le Q_i^{\max}$. Hence $Q_i[t] \le \theta_i p^{\max} + A^{\max}$ for all $t$.

When $t = 0$, $Z_i[0] = 0 < Z_i^{\max}$. We next prove if the inequality holds for $Z_i[t]$, it will also hold for $Z_i[t+1]$. Consider the following two cases: (1) *Case 1*: When $Z_i[t] \le V\ \alpha^{\max} + e\theta_i p^{\max}$, we have $Z_i[t+1] \le Z_i[t] + \epsilon_i \le V\ \alpha^{\max} + e\theta_i p^{\max} + \epsilon_i$ since queue length of $Z_i$ can increase by at most $\epsilon_i$ at one slot. (2) *Case 2*: When $Z_i[t] > V\alpha^{\max} + e\theta_i p^{\max}$, $Z_i[t] > V\ \alpha_j + e\theta_i p_{ij}[t]$ for all $j$, suggesting objective of dropping problem (23) is positive. Then at most $d_i^{\max}$ attack loads are dropped. We get $Z_i[t+1] \le Z_i[t] - r_i[t] - d_i^{\max} + \epsilon_i \le Z_i[t] \le Z_i^{\max}$ since $\epsilon_i \le d_i^{\max}$. Hence $Z_i[t]$ is bounded by $V\alpha^{\max} + e\theta_i p^{\max} + \epsilon_i$ at slot $t$.

When $t = 0$, $H_i[0] = 0 < H_i^{\max}$. Suppose this inequality is true for some slot $t$. Consider queue length bound in the following two cases: (1) *Case 1*: When $H_i[t] \le \frac{e\theta_i \sigma^{\max} p^{\max} + V\alpha^{\max}}{e^2 \sigma^{\min} p^{\min}}$, $H_i[t+1] \le H_i[t] + \beta_i[t] + a_i'[t] \le \frac{e\theta_i \sigma^{\max} p^{\max} + V\alpha^{\max}}{e^2 \sigma^{\min} p^{\min}} + \beta_i^{\max} + (a_i')^{\max} = H_i^{\max}$. (2) *Case 2*: When $H_i[t] > \frac{e\theta_i \sigma^{\max} p^{\max} + V\alpha^{\max}}{e^2 \sigma^{\min} p^{\min}}$, $e^2 \sigma_j p_{ij}[t] H_i[t] \ge e^2 \sigma^{\min} p^{\min} H_i[t] > e\theta_i \sigma^{\max} p^{\max} + V\alpha^{\max} > e\theta_i \sigma_j p_{ij}[t] + V\alpha_j > e\theta_i \sigma_j p_{ij}[t] + V\sigma_j c_i[t]$, suggesting objectives of (24) and (23) are positive. The defender resolves at most $B_i$ attacks and drops at most $d_i^{\max}$ attack loads. Since $r_i'[t] + d_i'[t] = eB_i p_{ij}[t] + ed_i^{\max} p_{ij}[t] \ge \overline{r}_i' + \overline{d}_i'$, the maximum of $a_i'[t] + \beta_i[t]$ should satisfy stability constraints, i.e., $\overline{r}_i' + \overline{d}_i' \ge \overline{\beta}_i + \overline{a}_i'$. We get $H_i[t+1] = H_i[t] - (r_i'[t] + d_i'[t] - \beta_i[t] - a_i'[t]) \le H_i[t] \le H_i^{\max}$. Hence virtual queue length bound (29) holds at any slot. □

## Appendix G
## Proof of Theorem 3

*Proof:* For risk queue $H_i$, resolved risks $r_i'[t] \le eB_i p^{\max}$ and dropped risks $d_i'[t] \le ed_i^{\max} p^{\max}$. There exists no risk underflow if $H_i[t] \ge ep^{\max}(B_i + d_i^{\max})$. Suppose $H_i[t] \ge ep^{\max}(B_i + d_i^{\max})$ and consider the following two cases: (1) *Case 1*: When $H_i[t] \ge 2ep^{\max}(B_i + d_i^{\max})$, $H_i[t+1] \ge H_i[t] - r_i'[t] - d_i'[t] \ge ep^{\max}(B_i + d_i^{\max})$. Thus, underflow will not occur. (2) *Case 2*: When $ep^{\max}(B_i + d_i^{\max}) \le H_i[t] \le 2ep^{\max}(B_i + d_i^{\max})$, $eH_i[t] - \frac{V\sigma^{\min} c_i[t]}{ep^{\max}} \le 2e^2 p^{\max}(B_i + d_i^{\max}) - \frac{V\sigma^{\min} c_i[t]}{ep^{\max}}$. Since $\theta_i \ge 2e^2 p^{\max}(B_i + d_i^{\max}) - \frac{V\sigma^{\min} c_i[t]}{ep^{\max}}$ and $\alpha_j > \max\ \sigma_j c_i[t]$, we have $0 \ge ep^{\max}(eH_i[t] - \theta_i - \frac{V\sigma^{\min} c_i[t]}{ep^{\max}}) \ge ep_{ij}[t]\ (eH_i[t] - \theta_i - \frac{V\sigma^{\min} c_i[t]}{ep_{ij}[t]}) > ep_{ij}[t](eH_i[t] - \theta_i - \frac{V\alpha_j}{ep_{ij}[t]})$. By solutions to problems (23) and (24), no attack will be resolved or dropped, i.e, $r_i[t] = d_i[t] = 0$. Thus, $H_i[t+1] \ge H_i[t] - r_i'[t] - d_i'[t] = H_i[t] \ge ep^{\max}(B_i + d_i^{\max})$. Above all, the theorem follows. □

## Appendix H
## Proof of Theorem 4

*Proof:* Dropping decisions $d_{ij}[t] = 0, \forall j \in \boldsymbol{N}_i[t]$ if $\sigma_j(Q_i[t] + Z_i[t] + (eH_i[t] - \theta_i)ep_{ij}[t]) \le V\alpha_j$. Since $\sigma_j(Q_i[t] + Z_i[t] + e^2 p_{ij}[t] H_i[t] - e\theta_i p_{ij}[t]) \le W_{ij}[t] = \sigma_j(Q_i[t] + Z_i[t] + e^2 p_{ij}[t] H_i[t] - e\theta_i p_{ij}[t] - Vc_i[t]) + V\log p_{ij}[t]$, no dropping condition holds only if $V\alpha_j \ge W_{ij}[t]$. Next prove $W_{ij}[t]$ is upper bounded. Construct a new queue corresponding to host $i$ with queue backlog $W_{ij}[t]$. At each slot, the input of queue is no larger than $\max\ \sigma_j(a_i[t] + \epsilon_i + e^2 p_{ij}[t](\beta_i[t] + a_i'[t])) + V\log e = \sigma^{\max}(A^{\max} + \epsilon^{\max} + \frac{V\log e}{\sigma^{\max}} + (ep^{\max})^2(\frac{e-1}{\sigma^{\min}} Q^{\max} + A^{\max}))$. The output is no smaller than $\min\ \sigma_j(2r_i[t] + e^2 p_{ij}[t] r_i'[t] + e(e-1)\theta_i p_{ij}[t]) = \sigma^{\min}((2 + e^3 (p^{\min})^2) r_i[t] + e(e-1)\theta^{\min} p^{\min})$. Under assignment algorithm, attacks with larger weight $W_{ij}[t]$ are given priority. Such assignment is a variation of MaxWeight algorithm proposed by Maguluri *et. al.* [42]. The constructed queue is stable if total arrival rate is smaller than $\frac{\Gamma - \sigma^{\max}}{\Gamma}$ fraction of total response rate, i.e., $|\boldsymbol{U}|\,\sigma^{\max}(A^{\max} + \epsilon^{\max} + \frac{V\log e}{\sigma^{\max}} + (ep^{\max})^2(\frac{e-1}{\sigma^{\min}} Q^{\max} + A^{\max})) \le \frac{\Gamma - \sigma^{\max}}{\Gamma}\sigma^{\min}((2 + e^3 (p^{\min})^2) \sum_{i \in \boldsymbol{U}} r_i[t] + e(e-1)\theta^{\min} p^{\min})$. Under stability constraint, queue backlog $W_{ij}[t]$ is bounded. Hence the theorem follows. □

## Appendix I
## Proof of Theorem 5

*Proof:* Under no attack dropping condition, we have $d_{ij}[t] = 0$. There exists an offline optimal algorithm that achieves the optimal system profit under any supportable attack arrival rate vector $\lambda_i$, which is within $\frac{\Gamma - \sigma^{\max}}{\Gamma}$ fraction of capacity region $\boldsymbol{\Omega}^i$. There exists $\delta \ge 0$ such that $\frac{(1+\delta)\Gamma}{\Gamma - \sigma^{\max}}\lambda_i \in \boldsymbol{\Omega}^i$. Let $a_{ij}^*[t]$, $r_{ij}^*[t]$ and $n_i^*[t]$ denote the decisions achieving the offline optimal time average profit $f^{\frac{(1+\delta)\Gamma}{\Gamma - \sigma^{\max}}}$. Then $\frac{\Gamma - \sigma^{\max}}{\Gamma} \sum_{j \in \boldsymbol{N}_i[n\Gamma]} \sigma_j \mathbb{E}\{r_{ij}^*[n\Gamma]\} \ge \frac{1+\delta}{\Gamma} \sum_{t=n\Gamma}^{(n+1)\Gamma-1} \sum_{j \in V} \mathbb{E}\{a_{ij}^*[t]\}$. Summing both sides in drift plus penalty (20) for $t$ from $n\Gamma$ to $(n+1)\Gamma - 1$, we get

$$
\begin{aligned}
&\mathbb{E}\{L(\boldsymbol{\Theta}[(n+1)\Gamma]) - L(\boldsymbol{\Theta}[n\Gamma]) - V \sum_{t=n\Gamma}^{(n+1)\Gamma-1} f[t] | \boldsymbol{\Theta}[n\Gamma]\} \\
&\le \Gamma B_1 + \sum_{t=0}^{\Gamma - \sigma^{\max} - 1} t B_3 + \sum_{i \in \boldsymbol{U}} \sum_{t=0}^{\Gamma-1} t(\epsilon_i^2 + (A^{\max})^2 \\
&\quad + \sum_{i \in \boldsymbol{U}} \sum_{t=0}^{\Gamma-1} t\, p^{\max}((1-e)\theta_i + p^{\max} A^{\max})(\frac{e-1}{\sigma^{\min}} Q^{\max} + A^{\max})) \\
&\quad + \sum_{i \in \boldsymbol{U}} \sum_{t=0}^{\Gamma - \sigma^{\max} - 1} t\sigma^{\max} B_i V(c_i^{\max} - c_i^{\min})) \\
&\quad + \sum_{i \in \boldsymbol{U}} B_i (\sigma^{\max})^2 (V c_i^{\max} + e p^{\max} \theta_i + V \log \omega^{\max}) \\
&\quad + \sum_{i \in \boldsymbol{U}} \sum_{j \in \boldsymbol{N}_i[n\Gamma]} \sigma_j e\theta_i (\Gamma - \sigma^{\max}) p_{ij}[n\Gamma] r_{ij}^*[n\Gamma] \\
&\quad + \sum_{i \in \boldsymbol{U}} \sum_{j \in \boldsymbol{N}_i[n\Gamma]} V(\Gamma - \sigma^{\max}) r_{ij}^*[n\Gamma](\sigma_j c_i[n\Gamma] - \log \frac{\omega_j}{n_i^*[n\Gamma]}) \\
&\quad - \sum_{i \in \boldsymbol{U}} \sum_{t=n\Gamma}^{(n+1)\Gamma-1} \sum_{j \in \boldsymbol{N}_i[t]} V \mathbb{E}\{r_{ij}^*[t] \log \frac{\omega_j}{n_i^*[t]} | \boldsymbol{\Theta}[n\Gamma]\} \\
&\quad + \sum_{i \in \boldsymbol{U}} \sum_{t=n\Gamma}^{(n+1)\Gamma-1} \sum_{j \in \boldsymbol{V}} Q_i[n\Gamma] \mathbb{E}\{a_{ij}^*[t] | \boldsymbol{\Theta}[n\Gamma]\} \\
&\quad - \sum_{i \in \boldsymbol{U}} \sum_{j \in \boldsymbol{N}_i[n\Gamma]} Q_i[n\Gamma](\Gamma - \sigma^{\max}) \sigma_j r_{ij}^*[n\Gamma] \\
&\quad - \sum_{i \in \boldsymbol{U}} \sum_{j \in \boldsymbol{N}_i[n\Gamma]} Z_i[n\Gamma](\Gamma - \sigma^{\max}) \sigma_j r_{ij}^*[n\Gamma] - Z_i[n\Gamma]\Gamma\epsilon_i \\
&\quad + \sum_{i \in \boldsymbol{U}} \sum_{t=n\Gamma}^{(n+1)\Gamma-1} \sum_{j \in \boldsymbol{V}} (H_i[n\Gamma] - \theta_i) p_{ij}[t] \mathbb{E}\{a_{ij}^*[t] | \boldsymbol{\Theta}[n\Gamma]\}
\end{aligned}
$$

$$-\sum_{i\in\boldsymbol{U}}\sum_{j\in\boldsymbol{N}_i[n\Gamma]} H_i[n\Gamma]((\Gamma-\sigma^{\max})\times\sigma_j e^2 p_{ij}[n\Gamma]r^*_{ij}[n\Gamma]-(1-e)\Gamma\theta_i)$$

Taking expectations of the above, summing resulting telescoping series for $n$ from 0 to $\eta-1$, dividing by $\eta V\Gamma$ yields

$$\begin{aligned}
&\frac{\mathbb{E}\{L(\boldsymbol{\Theta}[\eta\Gamma])\}}{\eta V\Gamma}-\frac{\mathbb{E}\{L(\boldsymbol{\Theta}\ [0])\}}{\eta V\Gamma}-\frac{1}{\eta\Gamma}\sum_{n=0}^{\eta-1}\sum_{t=n\Gamma}^{(n+1)\Gamma-1}\mathbb{E}\{f[t]\}\\
&\le \frac{B_1}{V}+\frac{(\Gamma-\sigma^{\max})(\Gamma-\sigma^{\max}-1)B_3}{2\Gamma V}\\
&\quad+\frac{(\Gamma-\sigma^{\max})(\Gamma-\sigma^{\max}-1)}{2\Gamma}\sum_{i\in\boldsymbol{U}}\sigma^{\max}B_i(c_i^{\max}-c_i^{\min})\\
&\quad+\frac{\Gamma-1}{2V}\sum_{i\in\boldsymbol{U}}(\epsilon_i^2+(A^{\max})^2\\
&\quad+p^{\max}(\frac{(e-1)Q^{\max}}{\sigma^{\min}}+A^{\max})(\theta_i(1-e)+p^{\max}A^{\max}))\\
&\quad+\frac{(\sigma^{\max})^2}{\Gamma}\sum_{i\in\boldsymbol{U}}B_i(c_i^{\max}+\log\omega^{\max}+\frac{ep^{\max}\theta_i}{V})\\
&\quad-\frac{1}{\eta\Gamma}\sum_{n=0}^{\eta-1}\sum_{t=n\Gamma}^{(n+1)\Gamma-1}\mathbb{E}\{\sum_{i\in\boldsymbol{U}}\sum_{j\in\boldsymbol{N}_i[t]}r^*_{ij}[t]\log\frac{\omega_j}{n_i^*[t]}\}+\frac{\Gamma-\sigma^{\max}}{\eta\Gamma}\\
&\quad\cdot\sum_{n=0}^{\eta-1}\mathbb{E}\{\sum_{i\in\boldsymbol{U}}\sum_{j\in\boldsymbol{N}_i[n\Gamma]}r^*_{ij}[n\Gamma](\sigma_j c_i[n\Gamma]-\log\frac{\omega_j}{n_i^*[n\Gamma]})\}-\frac{\delta}{\eta V\Gamma}\\
&\quad\cdot\sum_{n=0}^{\eta-1}\sum_{i\in\boldsymbol{U}}\mathbb{E}\{Q_i[n\Gamma]\}\sum_{t=n\Gamma}^{(n+1)\Gamma-1}\sum_{j\in\boldsymbol{V}}\mathbb{E}\left\{a^*_{ij}[t]\right\}-\frac{\delta}{\eta V}\sum_{n=0}^{\eta-1}\sum_{i\in\boldsymbol{U}}\epsilon_i\\
&\quad\cdot\mathbb{E}\{Z_i[n\Gamma]\}+\frac{(e-1)\theta_i}{\eta V}\sum_{n=0}^{\eta-1}\sum_{i\in\boldsymbol{U}}\mathbb{E}\{H_i[n\Gamma]\}-\frac{\Gamma-\sigma^{\max}}{\eta V\Gamma}\frac{1-e^2}{1+\delta}\\
&\quad\cdot\sum_{n=0}^{\eta-1}\sum_{i\in\boldsymbol{U}}(\mathbb{E}\{H_i[n\Gamma]\}-\theta_i)\sum_{j\in\boldsymbol{N}_i[n\Gamma]}\sigma_j p_{ij}[n\Gamma]\mathbb{E}\{r^*_{ij}[n\Gamma]\}
\end{aligned}$$

Using the facts that $\lim_{\eta\to\infty}\frac{1}{\eta\Gamma}\sum_{n=0}^{\eta-1}\sum_{t=n\Gamma}^{(n+1)\Gamma-1}\mathbb{E}\{\sum_{i\in\boldsymbol{U}}\sum_{j\in\boldsymbol{N}_i[t]}r^*_{ij}[t]\log\frac{\omega_j}{n_i^*[t]}\}-\lim_{\eta\to\infty}\frac{\Gamma-\sigma^{\max}}{\eta\Gamma}\sum_{n=0}^{\eta-1}\mathbb{E}\{\sum_{i\in\boldsymbol{U}}\sum_{j\in\boldsymbol{N}_i[n\Gamma]}r^*_{ij}[n\Gamma](\sigma_j\cdot c_i[n\Gamma]-\log\frac{\omega_j}{n_i^*[n\Gamma]})\}\ge f^{\frac{(1+\delta)\Gamma}{\Gamma-\sigma^{\max}}}$.Taking limits for both sides of the above as $\eta\to\infty$, the theorem thus follows. □

## Appendix J
## Proof of Theorem 6

*Proof:* Denote $e_i^p[t]=\hat{p}_{ij}[t]-p_{ij}[t]$, and $e_i^X[t]=\hat{X}_i[t]-X_i[t]$ for $Q_i[t]$, $Z_i[t]$, $H_i[t]$. Since $|e_i^Q[t]|$, $|e_i^Z[t]|$, $|e_i^H[t]|\le\xi^Q$, $|e_i^p[t]|=|e_i^\omega[t]e^{t-t_j}|\le\xi^\omega e^{D_{\max}}$, and $|a_i[t]-r_i[t]-d_i[t]|\le A^{\max}+B_i\sigma^{\max}+d_i^{\max}$, $|a_i'[t]-r_i'[t]-d_i'[t]|\le p^{\max}(A^{\max}+eB_i+ed_i^{\max})$, we have

$$\begin{aligned}
&\mathbb{E}\{L(\hat{\boldsymbol{\Theta}}[t+1])-L(\hat{\boldsymbol{\Theta}}[t])-Vf^\dagger[t]|\boldsymbol{\Theta}[t]\}\\
&\le G^*+\sum_{i\in\boldsymbol{U}}(\xi^Q(\epsilon_i+\theta_i(1-e)+A^{\max}(1+p^{\max}))\\
&\quad+B_i(2\sigma^{\max}+ep^{\max})+d_i^{\max}(2+ep^{\max}))\\
&\quad+\xi^\omega e^{D_{\max}}(A^{\max}(H_i^{\max}+\xi^Q+\theta_i)\\
&\quad+(B_i\sigma^{\max}+d_i^{\max})(e^2H_i^{\max}+e^2\xi^Q+e\theta_i))
\end{aligned}$$

where $G^*$ is the minimum of $\mathbb{E}\{L(\boldsymbol{\Theta}[t+1])-L(\boldsymbol{\Theta}[t])-Vf[t]|\boldsymbol{\Theta}[t]\}$, $a_{ij}^\dagger[t]$, $n_i^\dagger[t]$, $r_{ij}^\dagger[t]$, $d_{ij}^\dagger[t]$ are actions taken based on $\hat{\boldsymbol{\Theta}}[t]$. This shows (20) will still hold with $\boldsymbol{\Theta}[t]$ replaced by $\hat{\boldsymbol{\Theta}}[t]$, and $B_1$ replaced by $B_1'=B_1+\sum_{i\in\boldsymbol{U}}(\xi^Q(\epsilon_i+\theta_i(1-e)+A^{\max}(1+p^{\max}))+B_i(2\sigma^{\max}+ep^{\max})+d_i^{\max}(2+ep^{\max}))+\xi^\omega e^{D_{\max}}(A^{\max}(H_i^{\max}+\xi^Q+\theta_i)+(B_i\sigma^{\max}+d_i^{\max})(e^2H_i^{\max}+e^2\xi^Q+e\theta_i))$. Thus, the rest of the proof follows similarly as the proof of Theorem 5. □

## References


[1] "Annual security report," Cisco, San Jose, CA, USA, White Paper 1, Feb. 2017.

[2] "Advanced persistent threat awareness," ISACA, White Paper 3, Oct. 2015.

[3] "Worldwide infrastructure security report," Arbor Networks, Burlington, MA, USA, White Paper 12, Jan. 2017.

[4] Wikipedia. (Jan. 2018). *Advanced Persistent Threat*. [Online]. Available: https://en.wikipedia.org/wiki/Advanced_persistent_threat

[5] Cloud Security Alliance, "Big data analytics for security intelligence," Big Data Working Group, Res. Rep. 1, Sep. 2013.

[6] T.-F. Yen *et al.*, "Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks," in *Proc. ACM ACSAC*, Dec. 2013, pp. 199–208.

[7] K. Veeramachaneni, I. Arnaldo, V. Korrapati, C. Bassias, and K. Li, "AI$^2$: Training a big data machine to defend," in *Proc. IEEE Int. Conf. Big Data Secur.*, Apr. 2016, pp. 49–54.

[8] "Annual security report," Cisco, San Jose, CA, USA, White Paper 1, Feb. 2014.

[9] *iSIGHT Threat Intelligence*, Data Sheet No. 1, FireEy, Milpitas, CA, USA, Sep. 2017.

[10] W. Tong and S. Zhong, "A unified resource allocation framework for defending against pollution attacks in wireless network coding systems," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 10, pp. 2255–2267, Oct. 2016.

[11] S. Wang and N. Shroff, "Security game with non-additive utilities and multiple attacker resources," *ACM Meas. Anal. Comput. Syst.*, vol. 1, no. 1, Jun. 2017, Art. no. 13.

[12] L. Xiao, D. Xu, C. Xie, N. B. Mandayam, and H. V. Poor, "Cloud storage defense against advanced persistent threats: A prospect theoretic study," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 3, pp. 534–544, Mar. 2017.

[13] M. Zhang, Z. Zheng, and N. B. Shroff, "A game theoretic model for defending against stealthy attacks with limited resources," in *Proc. GameSec*, Nov. 2015, pp. 93–112.

[14] A. Garnaev, M. Baykal-Gursoy, and H. V. Poor, "Incorporating attack-type uncertainty into network protection," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 8, pp. 1278–1287, Aug. 2014.

[15] M. Rezvani, V. Sekulic, A. Ignjatovic, E. Bertino, and S. Jha, "Interdependent security risk analysis of hosts and flows," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 11, pp. 2325–2339, Nov. 2015.

[16] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic security risk management using Bayesian attack graphs," *IEEE Trans. Dependable Secure Comput.*, vol. 9, no. 1, pp. 61–74, Jan. 2012.

[17] Z. Zhan, M. Xu, and S. Xu, "Predicting cyber attack rates with extreme values," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 8, pp. 1666–1677, Aug. 2015.

[18] Z. Zhan, M. Xu, and S. Xu, "Characterizing honeypot-captured cyber attacks: Statistical framework and case study," *IEEE Trans. Inf. Forensics Security*, vol. 8, no. 11, pp. 1775–1789, Nov. 2013.

[19] (Jan. 2018). *Cyberxingan*. [Online]. Available: http://www.cyberxingan.com

[20] M. van Dijk, A. Juels, A. Oprea, and R. L. Rivest, "FlipIt: The game of 'stealthy takeover,"' *J. Cryptol.*, vol. 26, no. 4, pp. 655–713, 2013.

[21] J. Zhao, H. Li, C. Wu, Z. Li, Z. Zhang, and F. C. M. Lau, "Dynamic pricing and profit maximization for the cloud with geo-distributed data centers," in *Proc. IEEE INFOCOM*, Apr. 2014, pp. 118–126.

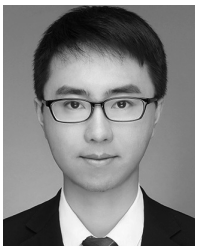
[22] M. J. Neely, *Stochastic Network Optimization With Application to Communication and Queueing Systems*. San Rafael, CA, USA: Morgan & Claypool, 2010.

[23] X. Wang, R. Jia, X. Tian, and X. Gan, "Dynamic task assignment in crowdsensing with location awareness and location diversity," in *Proc. IEEE INFOCOM*, Apr. 2018, pp. 1–9.

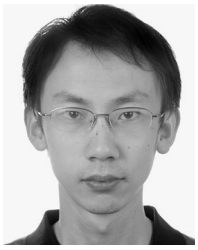
[24] F. Kelly, "Charging and rate control for elastic traffic," *Eur. Trans. Telecommun.*, vol. 8, no. 1, pp. 33–37, 1997.

[25] Y. Zhao *et al.*, "SmartSource: A mobile Q&A middleware powered by crowdsourcing," in *Proc. IEEE MDM*, Jun. 2015, pp. 145–156.
[26] R. Burkard and M. Dell'Amico and S. Martello, *Assignment Problems*. Philadelphia, PA, USA: SIAM, 2009.
[27] W. Wang, X. Wu, L. Xie, and S. Lu, "Femto-matching: Efficient traffic offloading in heterogeneous cellular networks," in *Proc. IEEE INFOCOM*, Apr. 2015, pp. 325–333.
[28] L. Gao, Y. Xu, and X. Wang, "MAP: Multiauctioneer progressive auction for dynamic spectrum access," *IEEE Trans. Mobile Comput.*, vol. 10, no. 8, pp. 1144–1161, Aug. 2011.
[29] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Proc. ACM SIGCOMM*, Aug. 2005, pp. 217–228.
[30] A. Almalawi, A. Fahad, Z. Tari, A. Alamri, R. AlGhamdi, and A. Y. Zomaya, "An efficient data-driven clustering technique to detect attacks in SCADA systems," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 5, pp. 893–906, May 2016.
[31] H. Ringberg, A. Soule, J. Rexford, and C. Diot, "Sensitivity of PCA for traffic anomaly detection," in *Proc. ACM SIGMETRICS*, Jun. 2017, pp. 109–120.
[32] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, Jun. 2014.
[33] S. Jamali and G. Shaker, "Defense against SYN flooding attacks: A scheduling approach," *J. Inf. Syst. Telecommun.*, vol. 2, no. 1, pp. 55–61, Jan. 2014.
[34] S. Xu, W. Lu, and H. Li, "A stochastic model of active cyber defense dynamics," *Internet Math.*, vol. 11, no. 1, pp. 23–61, 2015.
[35] K. Kaynar and F. Sivrikaya, "Distributed attack graph generation," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 5, pp. 519–532, Sep./Oct. 2016.
[36] H. S. Lallie, K. Debattista, and J. Bal, "An empirical evaluation of the effectiveness of attack graphs and fault trees in cyber-attack perception," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1110–1122, May 2018.
[37] Z. Zheng, N. B. Shroff, and P. Mohapatra, "When to reset your keys: Optimal timing of security updates via learning," in *Proc. AAAI*, Feb. 2017, pp. 3679–3685.
[38] W. Wei, H. Song, H. Wang, and X. Fan, "Research and simulation of queue management algorithms in ad hoc networks under DDoS attack," *IEEE Access*, vol. 5, pp. 27810–27817, Mar. 2017.
[39] N. C. Luong, D. T. Hoang, P. Wang, D. Niyato, and Z. Han, "Applications of economic and pricing models for wireless network security: A survey," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2735–2767, 4th Quart., 2017.
[40] P. Hu, H. Li, H. Fu, D. Cansever, and P. Mohapatra, "Dynamic defense strategy against advanced persistent threat with insiders," in *Proc. IEEE INFOCOM*, Apr. 2015, pp. 747–755.
[41] X. Wang, W. Huang, S. Wang, J. Zhang, and C. Hu, "Delay and capacity tradeoff analysis for motioncast," *IEEE/ACM Trans. Netw.*, vol. 19, no. 5, pp. 1354–1367, Oct. 2011.
[42] S. T. Maguluri, R. Srikant, and L. Ying, "Stochastic models of load balancing and scheduling in cloud computing clusters," in *Proc. IEEE INFOCOM*, Mar. 2013, pp. 702–710.

**Yuqing Li** received the B.S. degree in communication engineering from Xidian University, Xi'an, China, in 2014. She is currently pursuing the Ph.D. degree in electronic engineering with Shanghai Jiao Tong University, Shanghai, China. Her current research interests include network economics, social aware networks, edge computing system, and network security.



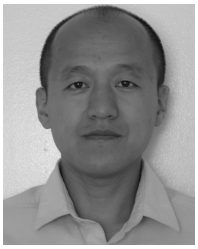
**Wenkuan Dai** received the B.S. degree in communication engineering from Xidian University, Xi'an, China, in 2016. He is currently pursuing the M.S. degree with the Department of Electronic Engineering, Shanghai Jiao Tong University. His research interests include charging station planning, and electric vehicles grid integration.



**Jie Bai** received the B.S. degree in computer science and technology from Hebei Agricultural University, Baoding, China, in 2007. His research interests include machine learning and recommendation system.

**Xiaoying Gan** received the Ph.D. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2006. From 2009 to 2010, she was a Visiting Researcher with the California Institute for Telecommunications and Information, University of California at San Diego, San Diego, CA, USA. She is currently an Associate Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University. Her current research interests include network economics, social aware networks, edge computing, multiuser multi-channel access, and dynamic resource management.

**Jingchao Wang** received the Ph.D. degree in electronics engineering from Tsinghua University, Beijing, China. His research interests include space information networks and satellite communications.



**Xinbing Wang** received the B.S. degree (Hons.) from the Department of Automation, Shanghai Jiao Tong University, Shanghai, China, in 1998, and the M.S. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2001, and the Ph.D. degree (major from the Department of Electrical and Computer Engineering and minor from the Department of Mathematics) from the North Carolina State University, Raleigh, NC, USA, in 2006. He is currently a Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University. He has been an Associate Editor of the IEEE/ACM TRANSACTIONS ON NETWORKING and the IEEE TRANSACTIONS ON MOBILE COMPUTING, and a member of the technical program committees of several conferences, including the ACM MobiCom 2012, the ACM MobiHoc 2012 and 2013, and the IEEE INFOCOM from 2009 to 2014.