An Active Learning Approach to Dynamic Alert Prioritization for Real-time Situational Awareness

Yeongwoo Kim and György Dán
Division of Network and Systems Engineering
School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology
Stockholm, Sweden
E-mail: {yeongwoo,gyuri}@kth.se

***Abstract*—Real-time situational awareness (SA) plays an essential role in accurate and timely incident response. Maintaining SA is, however, extremely costly due to excessive false alerts generated by intrusion detection systems, which require prioritization and manual investigation by security analysts. In this paper, we propose a novel approach to prioritizing alerts so as to maximize SA, by formulating the problem as that of active learning in a hidden Markov model (HMM). We propose to use the entropy of the belief of the security state as a proxy for the mean squared error (MSE) of the belief, and we develop two computationally tractable policies for choosing alerts to investigate that minimize the entropy, taking into account the potential uncertainty of the investigations' results. We use simulations to compare our policies to a variety of baseline policies. We find that our policies reduce the MSE of the belief of the security state by up to 50% compared to static baseline policies, and they are robust to high false alert rates and to the investigation errors.**

***Index Terms*—Situational awareness, intrusion detection, hidden Markov model, active learning**

I. Introduction

Accurate and timely incident response is essential for mitigating the impact of advanced persistent threat on networked systems, from critical infrastructures through financial institutions to governmental systems. The consequences of inaccurate incident response could be detrimental through limiting benign users' access to networked resources. Non-timely response actions, at the same time, could allow attackers ample time to achieve their objective.

A prerequisite for accurate and timely incident response is real-time situational awareness (SA), i.e., maintaining an accurate belief about potentially ongoing attacks and threats against the system. Maintaining SA in practice is, however, extremely challenging and resource intensive, as it has to be based on observing alerts from intrusion detection systems (IDSs), which typically generate a large amount of false alerts [1]. The investigation of alerts is typically done by highly skilled security analysts, and hence as the network size and the number of alerts grow, the investigation of all alerts becomes infeasible due to limited human resources.

Existing attempts to improve SA through improved alert investigation have developed schemes that prioritize alerts based on the level of importance of different types of alerts [2]–[4], and are thus static. Such schemes do not, however, take into account information available about the system, and in addition, sophisticated attackers could learn to bypass such static prioritization schemes [5]. Dynamic alert prioritization could address this issue, but since the defender cannot observe the security state of the system, it is unclear how to assign priorities to alerts dynamically so as to maximize SA in real-time, subject to a constraint in terms of the alerts.

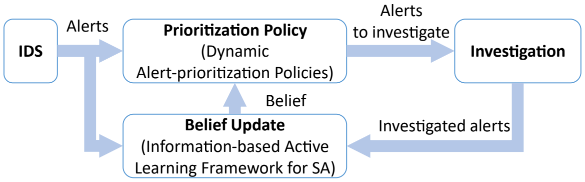

Fig. 1. Illustration of dynamic alert prioritization policies.

In this paper, we address the problem of dynamic alert prioritization to maximize SA, defined as the mean squared error (MSE) of the belief of the security state. We formulate the problem of SA as a hidden Markov model (HMM), where the security state is hidden from the defender and the defender's observations are noisy alerts. We then formulate alert prioritization as a novel variant of the active learning problem, where the defender can choose to investigate a limited number of alerts for reducing the observation noise, as illustrated in Fig. 1. Our main contributions are as follows:

- **Information-based Active Learning Framework for SA**: We propose to formulate real-time SA as a novel active learning problem for HMMs and to use the uncertainty of the defender's belief as a proxy for minimizing the MSE of the belief, as the uncertainty can be computed in real-time unlike the MSE. We provide belief update equations that account for our model of active learning, extending the traditional forward-backward algorithm.
- **Dynamic Alert-prioritization Policies**: We propose two policies for dynamic alert prioritization based on the entropy of the belief. The first policy chooses alerts whose investigation would lead to the highest reduction of the belief entropy. The second policy chooses the alerts with the lowest likelihood ratio, i.e., largest uncertainty, given the current belief.
- **Simulation-based Evaluation**: We use simulations to evaluate the proposed policies against a variety of base-

lines, including commonly used static policies. Our results show that the proposed policies lead to significantly lower MSE, are more robust to investigation errors and to false positives, and achieve a certain MSE reduction at the price of significantly less investigations.

The rest of the paper is organized as follows. We discuss the related works in Section II. Section III describes our system model and problem formulation, and Section IV details the belief update equations for active learning. In Section V, we introduce the proposed policies for minimizing the state estimation error. We evaluate the proposed policies in Section VI, and we conclude the paper in Section VII.

II. Related Work

A number of recent works consider the investigation of noisy alerts from IDSs by security analysts [1]–[4]. Authors of [2] formulated the problem of allocating alerts to security analysts as a game, and proposed heuristics for solving the resulting game. A zero-sum Markov game model was proposed in [3], and an approach based on dynamic programming and Q-maximin value iteration was developed for the optimal allocation of alerts to analysts. Shah et al. [4] used a reinforcement learning (RL) model to maximize the level of operational effectiveness while the security analysts shift every two weeks. Common to these works [2]–[4] is that the alert priorities are assumed to be known, and the focus is on assigning alerts to analysts [5]. On the contrary, in our work the focus is on prioritizing the alerts based on the real-time belief of the security state, so as to maximize SA.

A different line of works considers intrusion response systems (IRSs) coupled with the estimation of the security state (i.e., the attacker's progression) [6]–[11]. Authors of [6] used a partially observable Markov decision process (POMDP) to estimate the security state of the host and to choose defensive actions while minimizing the defender's cost. In [7], the authors proposed a hierarchy of local engines and global engines. Each local engine has an attack-response tree that calculates the security state of the host, and the global engine collects the security states from local engines and computes the defense actions. Iannucci et al. [8], [9] computed responses by considering the fact that a defense action may change the available exploits and may limit other attacks. Miehling et al. [10] modeled the security state by Bayesian attack graphs. Given the noisy alerts, the model calculates the belief regarding the attacker's privilege. The authors of [11] extended the model by including multiple dependencies for exploits and probabilistic alerts. These works either focus on a single host [6], assume that the system state is observable [8], [9], and learn about the security state through interaction with attacker [7], [10], [11], but they do not consider the prioritization of alerts and their investigation, which is the focus of our work.

Our methodology is closely related to active learning for HMMs [12], where the learner can decide what extra observations, called queries, to make about the system state and at what time. Contrary to this work, in our model queries can only concern existing alerts and their result could be noisy as well, which makes the learning problem fundamentally distinct from existing literature.

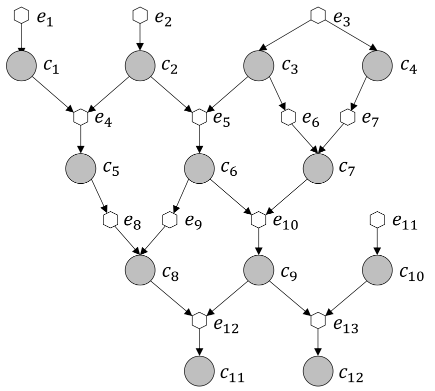

Fig. 2. Dependency graph with 12 conditions and 13 exploits in [11]. Each exploit e_x is connected to preconditions $\mathcal{N}_x^-$ and postconditions $\mathcal{N}_x^+$ (e.g., $\mathcal{N}_4^- = \{c_1, c_2\}$ and $\mathcal{N}_4^+ = \{c_5\}$, hence, $e_4 = (\{c_1, c_2\}, \{c_5\})$). There are four initial exploits, $\mathcal{E}_0 = \{e_1, e_2, e_3, e_{11}\}$ and two goal conditions $\mathcal{N}_g = \{c_{11}, c_{12}\}$.

III. System Model and Problem Formulation

In order to model the progression of an attacker in the system, we adopt an attack graph model in which *security states* are represented by nodes and *exploits* are represented by directed edges. This abstraction is often called a dependency graph [13]. In order to build the dependency graph, all possible security states are enumerated and interconnected by exploits. Dependency graphs can be constructed based on vulnerability information about the system components using network scanning tools like TVA [14].

We model the dependency graph by a directed acyclic hypergraph $\mathcal{H} = (\mathcal{N}, \mathcal{E})$ shown in Fig. 2, where $\mathcal{N} = \{c_1, \ldots, c_{n_c}\}$ is the set of nodes (i.e., security conditions), and $\mathcal{E} = \{e_1, \ldots, e_{n_e}\}$ is the set of directed hyperedges (i.e., exploits), where $n_c = |\mathcal{N}|$ and $n_e = |\mathcal{E}|$. Each condition corresponds to a certain compromise of a system component (e.g., privilege escalation), and can be true or false. By denoting benign and malicious conditions by false and true respectively, we can increase the number of true conditions based on the attacker's progression. We refer to the progression as a state, and thus the initial state is where the attacker has not compromised any system component. The subset $\mathcal{N}^g \subseteq \mathcal{N}$ is the set of goal conditions and represents the attacker's final goal. The defender identifies the goal states based on their importance, e.g., corresponding to critical assets and data. For instance, gaining root privilege on a customer database can be the goal state of an attacker aiming at data exfiltration.

We define a hyperedge $e_x \in \mathcal{E}$ as an ordered pair of two sets such that $e_x = (\mathcal{N}_x^-, \mathcal{N}_x^+)$, where $\mathcal{N}_x^- \subseteq \mathcal{N}$ is the set of preconditions to execute exploit e_x, and $\mathcal{N}_x^+ \subseteq \mathcal{N}$ is the set of postconditions after the successful execution of exploit e_x. Thus, a hyperedge (i.e., an exploit) connects its *preconditions* to its *postconditions*. If all preconditions for exploit e_x are not satisfied, the attacker cannot perform the exploit. There are also exploits that do not require any preconditions such that $\mathcal{N}_x^- = \emptyset$. We denote by $\mathcal{E}_0 \subseteq \mathcal{E}$ the set of such exploits, and we refer to them as *initial exploits*, i.e., the attacker's entry points to the network. If the attacker succeeds with an exploit, the postconditions $\mathcal{N}_x^+$ become true. The gained postconditions may be preconditions for future exploits.

We define the security state of the system as the subset of conditions that are true, i.e., $s \subseteq \mathcal{N}$, and we denote by $S \subseteq 2^{\mathcal{N}}$ the set of all security states. The security state effectively models the attacker's progress in compromising components. We denote by $\mathcal{E}^+(s_i)$ the exploits that are available to the attacker in the state $s_i \in S$. We make the assumption that the successful use of an exploit does not affect the success of exploits already used. This assumption is referred to as *monotonicity* [11], allowing to eliminate the necessity of enumerating all combinations of security conditions for S.

A. Attacker Model

Time is slotted, and in every time step an attacker can choose to perform a set $\mathcal{E}_t \subseteq \mathcal{E}^+(s_i)$ of exploits so as to compromise additional components. We consider that there is a set Φ of attacker types, and the choice of exploits depends on the attacker type φ_l. An attacker type φ_l is characterized by the probability $\alpha_{e_x}(\varphi_l) > 0$ that the attacker chooses $e_x \in \mathcal{E}^+(s_i)$ (resp. $\alpha_{e_x}(\varphi_l) = 0$ for $e_x \in \mathcal{E} \setminus \mathcal{E}^+(s_i)$), by the probability $\beta_{e_x}(\varphi_l)$ that it succeeds with $e_x \in \mathcal{E}_t$ gaining the additional conditions ($s_{i'} = s_i \cup \mathcal{N}_x^+$), and by the probability $\delta_{xa}(\varphi_l)$ that the alert z_a will be triggered if the attacker attempts to use exploit $e_x \in \mathcal{E}_t$, where a is the index of an alert that may be potentially triggered by exploit e_x. These three probabilities allow us to model attackers with different tactics and skill levels. We assume the defender is not aware of the type of the attacker, but is aware of the different types of attackers.

B. Defender model

The defender can observe the alerts from an IDS. As anomaly-based IDSs, we consider that alerts are noisy, *i.e.* alerts may be generated for benign events by legitimate users, called false positives, and may not be generated for malicious events, called false negatives.

We denote by $\mathcal{Z} = \{z_1, z_2, \ldots, z_{n_z}\}$ the set of alerts that the IDS can generate and by $\mathcal{A} = \{1, 2, \ldots, n_z\}$ the set of alert indices where n_z is the number of alert types. We associate each exploit $e_x \in \mathcal{E}$ with a subset $\mathcal{Z}(e_x) = \{z_{\mathcal{A}_x(1)}, z_{\mathcal{A}_x(2)}, \ldots, z_{\mathcal{A}_x(n_{e_x})}\} \subseteq \mathcal{Z}$, where $\mathcal{A}_x \subseteq \mathcal{A}$, and n_{e_x} is the number of distinct alerts that exploit e_x may raise. Note that some exploits may not generate any alert, while different exploits can generate the same alert. We denote by ζ_a the probability of false alerts for alert z_a.

Let us denote by $y_t \in Y = \{0, 1\}^{n_z}$ the alert vector at time t, where 0 is an inactive alert, and 1 is an active alert. Given the observed alerts, at time t the defender can choose to investigate the alert(s) $V_t \subseteq \{a \mid y_t^a = 1, a \in \mathcal{A}\}$ where the superscript a is the index of an alert, and the number of alerts to investigate is limited by the investigation budget I such that $|V_t| \leq I$. Such an investigation corresponds to a security analyst looking at event logs that triggered the alert(s). Based on the outcome of the investigation the defender clears or confirms the alert(s) V_t. Our model accounts for the probability of the investigation error by defining the probability ω that the outcome of the investigation is incorrect (i.e., a false positive is confirmed or a true positive is cleared). We denote by $\hat{y}_t$ the alert vector after the investigation, and note that $\hat{y}_t$ can only differ from y_t in the investigated alert(s).

Thus, at time t the defender has access to the observed alerts $Y = \{y_0, y_1, \ldots, y_t\}$, the investigations $V = \{v_0, v_1, \ldots, v_t\}$, and the alerts after investigation $\hat{Y} = \{\hat{y}_0, \hat{y}_1, \ldots, \hat{y}_t\}$. These together constitute the history at time t as $h_t = (\pi_0, v_0, y_0, \hat{y}_0, \ldots, v_t, y_t, \hat{y}_t)$, which the defender can use for maintaining a belief π_t of the security state, based on its initial belief π_0. The defender's belief regarding the security state and the attacker's type at time step t is

$$\pi_t = \begin{bmatrix} \pi_t^{1,1} & \pi_t^{1,2} & \ldots & \pi_t^{1,n_\Phi} \\ \pi_t^{2,1} & \pi_t^{2,2} & \ldots & \pi_t^{2,n_\Phi} \\ \vdots & \vdots & \ddots & \vdots \\ \pi_t^{n_S,1} & \pi_t^{n_S,2} & \ldots & \pi_t^{n_S,n_\Phi} \end{bmatrix}, \tag{1}$$

where n_S is the number of possible states (i.e., $n_S = |S|$), and n_Φ is the number of attacker types (i.e., $n_\Phi = |\Phi|$). Thus, $\pi_t^{il} = P(S_t = s_i, \Phi_t = \varphi_l \mid H_t = h_t)$ is the probability that s_i is the true security state and φ_l is the true type, given the history h_t. The belief π_t is a doubly-stochastic matrix since each row and column is a probability mass function given the security state and the attacker's type, respectively.

C. Problem Formulation

The objective of the defender is to maximize its SA given the noisy alerts, i.e., the accuracy of its estimate of the security state of the system. A natural way to capture this objective is to minimize the MSE of the belief. Recall that the columns and rows of the belief matrix π_t stand for the attacker type and the states, hence the MSE can be expressed as

$$\begin{aligned} MSE(\pi_t, S_t) = \frac{1}{n_c} \sum_{j=1}^{n_c} \Big(& \mathbf{1}_{\{c_j \in S_t\}} \\ & - \sum_{i=1}^{n_S} \big(\mathbf{1}_{\{c_j \in s_i\}} \cdot \sum_{l=1}^{n_\Phi} \pi_t^{il} \big) \Big)^2, \end{aligned} \tag{2}$$

where $\mathbf{1}_{\{\cdot\}}$ is an indicator function that is 1 (resp. 0) if the condition is true (resp. false), $c_j \in \mathcal{N}$ is a condition, and S_t is the attacker's security state at time t. Considering an infinite time horizon, we define the operator's cost under policy κ as

$$J^\kappa = \lim_{T \to \infty} \frac{1}{T} \sum_{t=1}^{T} \gamma^t MSE(\pi_t^\kappa, S_t), \tag{3}$$

where $\gamma \in (0, 1)$ is the discount factor for the future uncertainty, ω is the the investigation error probability, and we are interested in finding a policy

$$\kappa^* \in \arg\min_{\kappa \in \mathcal{K}} J^\kappa. \tag{4}$$

The policy κ selects actions V_{t+1} given a belief Π_t and alert vector Y_{t+1} taking into account the investigation error probability ω, and is thus a mapping

$$\kappa : [0,1]^{n_\Phi \times n_S} \times \{0,1\}^{n_z} \rightarrow \{1,\dots,n_z\}^I. \tag{5}$$

The formulated problem is an active learning problem for an HMM, where queries are limited to a subset of existing observations, and we are interested in understanding the structure of near-optimal policies and factors that may affect their performance.

IV. Belief Update with Active Learning

We start with describing the belief update, assuming that a policy for choosing alerts to investigate exists. The defender updates its belief as new observations y_{t+1} and the result $\hat{y}_{t+1}$ of the investigation v_{t+1} become available. For a raw alert vector $y_{t+1} = y_n$, investigation $v_{t+1} = v$ and investigated alert vector $\hat{y}_{t+1} = \hat{y}_k$ the belief update is $\pi_{t+1} = \mathcal{T}_{i'l'}(\pi_t, \hat{y}_k, y_n, v)_{\varphi_{i'} \in \Phi, s_{l'} \in S}$, where $\mathcal{T}_{i'l'}(\pi_t, \hat{y}_k, y_n, v)$ is the update function for the i'th state and the l'th attacker type. The update for each entry of the belief matrix can be obtained using Bayes' theorem,

$$\begin{aligned}\pi_{t+1}^{i'l'} &= \mathcal{T}_{i'l'}(\pi_t, \hat{y}_k, y_n, v)\\ &= P(S_{t+1} = s_{i'}, \Phi_{t+1} = \varphi_{l'} \mid \hat{Y}_{t+1} = \hat{y}_k, V_{t+1} = v,\\ &\quad Y_{t+1} = y_n, \Pi_t = \pi_t)\\ &= \frac{p_{i'l'}^n(\pi_t) r_{i'nkl'}^v(\pi_t)}{\sigma(\pi_t, \hat{y}_k, y_n, v)}.\end{aligned} \tag{6}$$

The above terms are defined as

$$\begin{aligned}p_{i'l'}^n(\pi_t) &= P(S_{t+1} = s_{i'}, \Phi_{t+1} = \varphi_{l'}, Y_{t+1} = y_n \mid\\ &\quad V_{t+1} = v, \Pi_t = \pi_t)\\ &= \sum_{s_i \in S, \varphi_l \in \Phi} \pi_t^{il} p_{ii'l}^n q_{ll'}\end{aligned} \tag{7}$$

$$\begin{aligned}r_{i'nkl'}^v(\pi_t) &= P(\hat{Y}_{t+1} = \hat{y}_k \mid Y_{t+1} = y_n, S_{t+1} = s_{i'},\\ &\quad \Phi_{t+1} = \varphi_{l'}, V_{t+1} = v, \Pi_t = \pi_t)\\ &= \sum_{s_i \in S, \varphi_l \in \Phi} \pi_t^{il} r_{ii'nkll'}^v\end{aligned} \tag{8}$$

$$\begin{aligned}\sigma(\pi_t, \hat{y}_k, y_n, v) &= P(\hat{Y}_{t+1} = \hat{y}_k, Y_{t+1} = y_n \mid V_{t+1} = v,\\ &\quad \Pi_t = \pi_t)\\ &= \sum_{s_{i'} \in S, \varphi_{l'} \in \Phi} r_{i'nkl'}^v(\pi_t) p_{i'l'}(\pi_t),\end{aligned} \tag{9}$$

where $p_{ii'l}^n = P(S_{t+1} = s_{i'}, Y_{t+1} = y_n \mid V_{t+1} = v, S_t = s_i, \Phi_t = \varphi_l)$, $r_{ii'nkll'}^v = P(\hat{Y}_{t+1} = \hat{y}_k \mid V_{t+1} = v, Y_{t+1} = y_n, S_{t+1} = s_{i'}, \Phi_{t+1} = \varphi_{l'}, S_t = s_i, \Phi_t = \varphi_l)$, and $q_{ll'} = P(\Phi_{t+1} = \varphi_{l'} \mid \Phi_t = \varphi_l)$. For $q_{ll'}$, we assume that the attacker type does not change during its attack such that $q_{ll'} = 1$ (resp. 0) for $l = l'$ (resp. $l \neq l'$).

In eqn. (7), $p_{ii'l}^n = P(S_{t+1} = s_{i'}, Y_{t+1} = y_n | V_{t+1} = v, S_t = s_i, \Phi_t = \varphi_l)$ where the joint probability of the state $s_{i'}$ and the alert vector y_n is independent of investigation v. Thus, we can write

$$\begin{aligned}p_{ii'l}^n &= P(S_{t+1} = s_{i'}, Y_{t+1} = y_n \mid S_t = s_i, \Phi_t = \varphi_l)\\ &= \sum_{\mathcal{E}_t \in \mathscr{P}(\mathcal{E}^+(s_i))} P(S_{t+1} = s_{i'} \mid Y_{t+1} = y_n, E_t = \mathcal{E}_t, S_t = s_i,\\ &\qquad \Phi_t = \varphi_l)\\ &\qquad \cdot P(Y_{t+1} = y_n \mid E_t = \mathcal{E}_t, S_t = s_i, \Phi_t = \varphi_l)\\ &\qquad \cdot P(E_t = \mathcal{E}_t \mid S_t = s_i, \Phi_t = \varphi_l)\\ &= \sum_{\mathcal{E}_t \in \mathscr{P}(\mathcal{E}^+(s_i))} P(S_{t+1} = s_{i'} \mid E_t = \mathcal{E}_t, S_t = s_i, \Phi_t = \varphi_l)\\ &\qquad \cdot P(Y_{t+1} = y_n \mid E_t = \mathcal{E}_t, \Phi_t = \varphi_l)\\ &\qquad \cdot P(E_t = \mathcal{E}_t \mid S_t = s_i, \Phi_t = \varphi_l),\end{aligned} \tag{10}$$

where $\mathcal{E}^+(s_i)$ is the set of available exploits in the state s_i, and $\mathscr{P}(\mathcal{E}^+(s_i))$ is the power set of the available exploits. The state $s_{i'}$ is independent of the alert vector y_n given the exploits $\mathcal{E}_t$, the state s_i, and the attacker type φ_l, and that the alert vector y_n is independent of the state s_i given the exploits $\mathcal{E}_t$. The terms in eqn. (10) can be expressed as

$$\begin{aligned}&P(S_{t+1} = s_{i'} \mid E_t = \mathcal{E}_t, S_t = s_i, \Phi_t = \varphi_l)\\ &\quad = \sum_{o \in \mathcal{F}(s_i, s_{i'}, \varphi_l, \mathcal{E}_t)} \prod_{\{e_x \mid e_x \in o, e_x = 1\}} \beta_{e_x}(\varphi_l)\\ &\qquad \cdot \prod_{\{e_x \mid e_x \in o, e_x = 0\}} (1 - \beta_{e_x}(\varphi_l)),\end{aligned} \tag{11}$$

$$\begin{aligned}&P(Y_{t+1} = y_n \mid E_t = \mathcal{E}_t, \Phi_t = \varphi_l)\\ &\quad = \prod_{a \in \mathcal{A}} P(Y_{t+1}^a = y_n^a \mid E_t = \mathcal{E}_t, \Phi_t = \varphi_l),\end{aligned} \tag{12}$$

$$\begin{aligned}&P(E_t = \mathcal{E}_t \mid S_t = s_i, \Phi_t = \varphi_l)\\ &\quad = \prod_{e_x \in \mathcal{E}_t \cap \mathcal{E}^+(s_i)} \alpha_{e_x}(\varphi_l) \cdot \prod_{e_x \in \mathcal{E}^+(s_i) \setminus \mathcal{E}_t} (1 - \alpha_{e_x}(\varphi_l)),\end{aligned} \tag{13}$$

where $\mathcal{F}(s_i, s_{i'}, \varphi_l, v, \mathcal{E}_t)$ is the outcome (i.e., success or failure) of attempted exploits causing the state transition from s_i to $s_{i'}$ under attacker type φ_l and exploits $\mathcal{E}_t$; outcome o is the set of exploits, and each exploit e_x is either 1 (i.e., success) or 0 (i.e., failure). Note that eqn. (12) makes use of conditional independence of the individual alerts $z_a \in \mathcal{Z}$ given the exploits $\mathcal{E}_t$. The alert probabilities are given by

$$\begin{aligned}&P(Y_{t+1}^a = y_n^a \mid E_t = \mathcal{E}_t, \Phi_t = \varphi_l)\\ &\quad = \begin{cases} (1 - \zeta_a) \prod_{e_x \in \mathcal{E}_t \cap \mathcal{E}(z_a)} (1 - \delta_{xa}(\varphi_l)) \\ \hfill \text{if } y_n^a = 0 \\ 1 - \Big((1 - \zeta_a) \prod_{e_x \in \mathcal{E}_t \cap \mathcal{E}(z_a)} (1 - \delta_{xa}(\varphi_l)) \Big) \\ \hfill \text{if } y_n^a = 1, \end{cases}\end{aligned} \tag{14}$$

where $\mathcal{E}(z_a)$ is the set of exploits that may raise alert z_a, $\delta_{xa}(\varphi_l)$ is the probability that the attacker type φ_l triggers the alert z_a by using the exploit e_x, and ζ_a is the probability of the alert z_a being a false alert.

We can further express $r^v_{ii'nkll'}$ as

$$\begin{aligned} r^v_{ii'nkll'} &= P(\hat{Y}_{t+1} = \hat{y}_k \mid V_{t+1} = v, Y_{t+1} = y_n, S_{t+1} = s_{i'}, \\ &\qquad \Phi_{t+1} = \varphi_{l'}, S_t = s_i, \Phi_t = \varphi_l) \\ &= \frac{\begin{array}{c} P(\hat{Y}_{t+1} = \hat{y}_k, \Phi_{t+1} = \varphi_{l'} \mid V_{t+1} = v, \\ Y_{t+1} = y_n, S_{t+1} = s_{i'}, S_t = s_i, \Phi_t = \varphi_l) \end{array}}{\begin{array}{c} P(\Phi_{t+1} = \varphi_{l'} \mid V_{t+1} = v, Y_{t+1} = y_n, \\ S_{t+1} = s_{i'}, S_t = s_i, \Phi_t = \varphi_l) \end{array}} \\ &= \frac{\begin{array}{c} P(\hat{Y}_{t+1} = \hat{y}_k \mid V_{t+1} = v, Y_{t+1} = y_n, S_{t+1} = s_{i'}, \\ S_t = s_i, \Phi_t = \varphi_l) \cdot (\Phi_{t+1} = \varphi_{l'} \mid \Phi_t = \varphi_l) \end{array}}{P(\Phi_{t+1} = \varphi_{l'} \mid \Phi_t = \varphi_l)}, \end{aligned} \tag{15}$$

where $P(\Phi_{t+1} = \varphi_{l'} \mid V_{t+1} = v, Y_{t+1} = y_n, S_{t+1} = s_{i'}, S_t = s_i, \Phi_t = \varphi_l) = P(\Phi_{t+1} = \varphi_{l'} \mid \Phi_t = \varphi_l)$ due to the assumption on the attacker type, and $P(\Phi_{t+1} = \varphi_{l'} \mid \Phi_t = \varphi_l) = q_{ll'}$.

The first term in the numerator of eqn. (15) can further be expressed as

$$\begin{aligned} &P(\hat{Y}_{t+1} = \hat{y}_k \mid V_{t+1} = v, Y_{t+1} = y_n, S_{t+1} = s_{i'}, S_t = s_i, \Phi_t = \varphi_l) \\ &\quad = \sum_{\mathcal{E}_t \in \mathscr{P}(\mathcal{E}^+(s_i))} P(\hat{Y}_{t+1} = \hat{y}_k \mid E_t = \mathcal{E}_t, V_{t+1} = v, \\ &\qquad\qquad Y_{t+1} = y_n, S_{t+1} = s_{i'}, S_t = s_i, \Phi_t = \varphi_l) \\ &\qquad\qquad \cdot P(E_t = \mathcal{E}_t \mid V_{t+1} = v, Y_{t+1} = y_n, \\ &\qquad\qquad S_{t+1} = s_{i'}, S_t = s_i, \Phi_t = \varphi_l), \end{aligned} \tag{16}$$

$$\begin{aligned} &\quad = \sum_{\mathcal{E}_t \in \mathscr{P}(\mathcal{E}^+(s_i))} P(\hat{Y}_{t+1} = \hat{y}_k \mid E_t = \mathcal{E}_t, V_{t+1} = v, \\ &\qquad\qquad Y_{t+1} = y_n, \Phi_t = \varphi_l) \\ &\qquad\qquad \cdot P(E_t = \mathcal{E}_t \mid Y_{t+1} = y_n, S_{t+1} = s_{i'}, \\ &\qquad\qquad S_t = s_i, \Phi_t = \varphi_l), \end{aligned} \tag{17}$$

where eqn. (17) is due to the fact that the investigated alert vector $\hat{y}_k$ is independent of the states s_i and $s_{i'}$ given the exploits $\mathcal{E}_t$, the investigation v, the raw alert vector y_n, and the attacker type φ_l, and that the choices of exploits $\mathcal{E}_t$ are independent of the investigation v. Consider now the result of the investigation

$$\begin{aligned} &P(\hat{Y}_{t+1} = \hat{y}_k \mid E_t = \mathcal{E}_t, V_{t+1} = v, Y_{t+1} = y_n, \Phi_t = \varphi_l) \\ &\quad = \prod_{a \in \mathcal{A}} P(\hat{Y}^a_{t+1} = \hat{y}^a_k \mid E_t = \mathcal{E}_t, V_{t+1} = v, Y^a_{t+1} = y^a_n, \\ &\qquad\qquad \Phi_t = \varphi_l), \end{aligned} \tag{18}$$

where we make use of the conditional independence of individual investigations given the set of attempted exploits and the corresponding alert. Clearly, $P(\hat{Y}^a_{t+1} = Y^a_{t+1}) = 1$ for $a \notin V_{t+1}$, and $a \in V_{t+1}$ implies $Y^a_{t+1} = 1$, hence using that $\mathcal{E}(z_a)$ is the set of exploits that may trigger alert z_a, we can express the terms in eqn. (18) as

$$\begin{aligned} &P(\hat{Y}^a_{t+1} = \hat{y}^a_k \mid E_t = \mathcal{E}_t, V_{t+1} = v, Y^a_{t+1} = y^a_n, \Phi_t = \varphi_l) \\ &= \begin{cases} 1 \\ \qquad\qquad\qquad \text{if } a \notin v, y^a_n = \hat{y}^a_k \\ \dfrac{\begin{array}{l} \zeta_a \prod_{e_x \in \mathcal{E}_t \cap \mathcal{E}(z_a)} (1 - \delta_{xa}(\varphi_l))(1 - \omega) \\ \quad + (1 - \prod_{e_x \in \mathcal{E}_t \cap \mathcal{E}(z_a)} (1 - \delta_{xa}(\varphi_l)))\omega \end{array}}{1 - \left((1-\zeta_a) \prod_{e_x \in \mathcal{E}_t \cap \mathcal{E}(z_a)} (1 - \delta_{xa}(\varphi_l)) \right)} \\ \qquad\qquad\qquad \text{if } a \in v, y^a_n = 1,\ \hat{y}^a_k = 0 \\ \dfrac{\begin{array}{l} \zeta_a \prod_{e_x \in \mathcal{E}_t \cap \mathcal{E}(z_a)} (1 - \delta_{xa}(\varphi_l))\omega \\ \quad + (1 - \prod_{e_x \in \mathcal{E}_t \cap \mathcal{E}(z_a)} (1 - \delta_{xa}(\varphi_l)))(1 - \omega) \end{array}}{1 - \left((1-\zeta_a) \prod_{e_x \in \mathcal{E}_t \cap \mathcal{E}(z_a)} (1 - \delta_{xa}(\varphi_l)) \right)} \\ \qquad\qquad\qquad \text{if } a \in v, y^a_n = 1, \hat{y}^a_k = 1, \end{cases} \end{aligned} \tag{19}$$

where ω is the probability that the defender's investigation incorrectly identifies the cause of the alert. The terms $\zeta_a \prod_{e_x \in \mathcal{E}_t \cap \mathcal{E}(z_a)} (1 - \delta_{xa}(\varphi_l))$ and $(1 - \prod_{e_x \in \mathcal{E}_t \cap \mathcal{E}(z_a)} (1 - \delta_{xa}(\varphi_l)))$ in eqn. (19) stand for the probability of the alert being a false positive and the probability of the alert being a true positive, respectively.

Let us consider now the last term in eqn. (17). For attacker type φ_l, the probability of exploits $\mathcal{E}_t$ given the transition from s_i to $s_{i'}$ and the alert vector y_n can be expressed as

$$\begin{aligned} &P(E_t = \mathcal{E}_t \mid Y_{t+1} = y_n, S_{t+1} = s_{i'}, S_t = s_i, \Phi_t = \varphi_l) \\ &\quad = \frac{P(E_t = \mathcal{E}_t, Y_{t+1} = y_n \mid S_{t+1} = s_{i'}, S_t = s_i, \Phi_t = \varphi_l)}{P(Y_{t+1} = y_n \mid S_{t+1} = s_{i'}, S_t = s_i, \Phi_t = \varphi_l)} \\ &\quad = \frac{P(Y_{t+1} = y_n, E_t = \mathcal{E}_t \mid S_{t+1} = s_{i'}, S_t = s_i, \Phi_t = \varphi_l)}{\begin{array}{r} \sum_{\mathcal{E}'_t \in \mathscr{P}(\mathcal{E}(s_i, s_{i'}))} P(Y_{t+1} = y_n, E_t = \mathcal{E}'_t \mid S_{t+1} = s_{i'}, \\ S_t = s_i, \Phi_t = \varphi_l) \end{array}} \\ &\quad = \frac{\begin{array}{c} P(Y_{t+1} = y_n \mid E_t = \mathcal{E}_t, \Phi_t = \varphi_l) \\ \cdot P(E_t = \mathcal{E}_t \mid S_{t+1} = s_{i'}, S_t = s_i, \Phi_t = \varphi_l) \end{array}}{\begin{array}{r} \sum_{\mathcal{E}'_t \in \mathscr{P}(\mathcal{E}(s_i, s_{i'}))} P(Y_{t+1} = y_n \mid E_t = \mathcal{E}'_t, \Phi_t = \varphi_l) \\ \cdot P(E_t = \mathcal{E}'_t \mid S_{t+1} = s_{i'}, S_t = s_i, \Phi_t = \varphi_l) \end{array}}, \end{aligned} \tag{20}$$

where we use the fact that the alert vector is independent of the states given the attacker type and the exploits, $\mathcal{E}(s_i, s_{i'})$ is the exploits that result in the transition from s_i to $s_{i'}$, and $\mathscr{P}(\mathcal{E}(s_i, s_{i'}))$ is the power set of the exploits causing the transition from s_i to $s_{i'}$. $P(Y_{t+1} = y_n \mid E_t = \mathcal{E}_t, \Phi_t = \varphi_l)$ is given by eqn. (12), and $P(E_t = \mathcal{E}_t \mid S_{t+1} = s_{i'}, S_t = s_i, \Phi_t = \varphi_l)$ is the probability of exploits given the transitions as

$$\begin{aligned} &P(E_t = \mathcal{E}_t \mid S_{t+1} = s_{i'}, S_t = s_i, \Phi_t = \varphi_l) \\ &\quad = \frac{\begin{array}{c} P(S_{t+1} = s_{i'} \mid E_t = \mathcal{E}_t, S_t = s_i, \Phi_t = \varphi_l) \\ \cdot P(E_t = \mathcal{E}_t \mid S_t = s_i, \Phi_t = \varphi_l) \end{array}}{P(S_{t+1} = s_{i'} \mid S_t = s_i, \Phi_t = \varphi_l)}, \end{aligned} \tag{21}$$

where $P(S_{t+1} = s_{i'} \mid E_t = \mathcal{E}_t, S_t = s_i, \Phi_t = \varphi_l)$ is given by eqn. (11), $P(E_t = \mathcal{E}_t \mid S_t = s_i, \Phi_t = \varphi_l)$ is given by eqn. (13), and $P(S_{t+1} = s_{i'} \mid S_t = s_i, \Phi_t = \varphi_l)$ is expressed as

$$
\begin{aligned}
&P(S_{t+1} = s_{i'} \mid S_t = s_i, \Phi_t = \varphi_l) \\
&\quad = \sum_{o \in \mathcal{F}(s_i, s_{i'}, \varphi_l)} \prod_{\{e_x \mid e_x \in o, e_x = 1\}} \alpha_{e_x}(\varphi_l) \beta_{e_x}(\varphi_l) \\
&\qquad\qquad \cdot \prod_{\{e_x \mid e_x \in o, e_x = 0\}} (1 - \alpha_{e_x}(\varphi_l) \beta_{e_x}(\varphi_l)),
\end{aligned} \tag{22}
$$

where $\mathcal{F}(s_i, s_{i'}, \varphi_l)$ is the outcome (i.e., success or failure) of attempted exploits causing the state transition from s_i to $s_{i'}$ under attacker type φ_l.

V. Design of Defender Policies

A fundamental challenge in the considered problem is that the type of the attacker and the current state are unknown to the defender, hence it is not possible to formulate defender policies that directly minimize the MSE. To circumvent this issue, we propose to use (un)certainty as a substitute metric for the MSE in formulating defender policies, motivated by the observation that accurate SA corresponds to a low uncertainty in the belief about the system's security state. The most common way to quantify (un)certainty is through the entropy of the operator's belief, defined at time step t as

$$
H(\pi_t) = -\sum_{i=1}^{n_S} \sum_{l=1}^{n_\Phi} \pi_t^{il} \log(\pi_t^{il}). \tag{23}
$$

As an alternative we also explore the likelihood ratio of the observations, called the Bayes factor. In what follows we present two policies based on these metrics of uncertainty for maximizing situational awareness. We present the policies for the case $I = 1$, we then discuss how to use them for $I > 1$.

A. MaxEntropy policy

The MaxEntropy policy aims to choose alerts $z_a \in \mathcal{Z}$ that, after investigation, would provide the highest reduction of the entropy[1] of the belief. The policy requires the computation of $\mathcal{T}_{i'l'}(\pi_t, \hat{y}_k, y_n, v)$ for the potentially resulting $2|\{a \mid y_n^a = 1\}|$ investigated alert vectors $\hat{y}_k$, and for each such alert vector the probability $P(S_{t+1} = s_{i'}, \Phi_{t+1} = \varphi_{l'}, \hat{Y}_{t+1}^a = \hat{y}_k^a \mid V_{t+1} = a, Y_{t+1}^a = 1, \Pi_t = \pi_t)$ of its occurrence, which can be computed by conditioning on the set $\mathcal{E}_t$ of exploits used by the attacker. In order to obtain the result of an investigation for the state $s_{i'}$ and the attacker type $\varphi_{l'}$, we marginalize eqn. (19) and eqn. (11) over the power set of exploits $\mathscr{P}(\mathcal{E}^+(s_i))$ as

$$
\begin{aligned}
&P(S_{t+1} = s_{i'}, \hat{Y}_{t+1}^a = \hat{y}_k^a \mid V_{t+1} = a, Y_{t+1}^a = 1, S_t = s_i, \\
&\quad \Phi_t = \varphi_l) \\
&\quad = \sum_{\mathcal{E}_t \in \mathscr{P}(\mathcal{E}^+(s_i))} P(\hat{Y}_{t+1}^a = \hat{y}_k^a \mid E_t = \mathcal{E}_t, V_{t+1} = a, \\
&\qquad\qquad Y_{t+1}^a = 1, \Phi_t = \varphi_l) \\
&\qquad\qquad \cdot P(E_t = \mathcal{E}_t \mid Y_{t+1}^a = 1, S_t = s_i, \Phi_t = \varphi_l) \\
&\qquad\qquad \cdot P(S_{t+1} = s_{i'} \mid E_t = \mathcal{E}_t, S_t = s_i, \Phi_t = \varphi_l),
\end{aligned} \tag{24}
$$

where $P(\hat{Y}_{t+1}^a = \hat{y}_k^a \mid E_t = \mathcal{E}_t, V_{t+1} = a, Y_{t+1}^a = y_n^a, \Phi_t = \varphi_l)$ is given by eqn. (19), and $P(S_{t+1} = s_{i'} | E_t = \mathcal{E}_t, S_t = s_i, \Phi_t = \varphi_l)$ is given by eqn. (11). The probability of exploits is

$$
\begin{aligned}
&P(E_t = \mathcal{E}_t \mid Y_{t+1}^a = 1, S_t = s_i, \Phi_t = \varphi_l) \\
&\quad = \frac{\begin{aligned} &P(E_t = \mathcal{E}_t \mid S_t = s_i, \Phi_t = \varphi_l) \\ &\cdot P(Y_{t+1}^a = 1 \mid E_t = \mathcal{E}_t, \Phi_t = \varphi_l) \end{aligned}}{\begin{aligned} \sum\nolimits_{\mathcal{E}'_t \in \mathscr{P}(\mathcal{E}^+(s_i))} &P(E_t = \mathcal{E}'_t \mid S_t = s_i, \Phi_t = \varphi_l) \\ &\cdot P(Y_{t+1}^a = 1 \mid E_t = \mathcal{E}_t, \Phi_t = \varphi_l) \end{aligned}},
\end{aligned} \tag{25}
$$

where $P(E_t = \mathcal{E}_t | S_t = s_i, \Phi_t = \varphi_l)$ is given by eqn. (13), and $P(Y_{t+1}^a = 1 \mid E_t = \mathcal{E}_t, \Phi_t = \varphi_l)$ is given by eqn. (14). Then, since the next attacker type is only dependent on the previous attacker type, we obtain

$$
\begin{aligned}
&P(S_{t+1} = s_{i'}, \Phi_{t+1} = \varphi_{i'}, \hat{Y}_{t+1}^a = \hat{y}_k^a \mid V_{t+1} = a, Y_{t+1}^a = 1, \\
&\quad S_t = s_i, \Phi_t = \varphi_l) \\
&\quad = P(S_{t+1} = s_{i'}, \hat{Y}_{t+1}^a = \hat{y}_k^a \mid V_{t+1} = a, Y_{t+1}^a = 1, \\
&\qquad S_t = s_i, \Phi_t = \varphi_l) \cdot P(\Phi_{t+1} = \varphi_{l'} | \Phi_t = \varphi_l).
\end{aligned} \tag{26}
$$

The updated belief taking into account the outcome of the investigation can then be expressed as

$$
\begin{aligned}
&P(S_{t+1} = s_{i'}, \Phi_{t+1} = \varphi_{i'}, \hat{Y}_{t+1}^a = \hat{y}_k^a \mid V_{t+1} = a, Y_{t+1}^a = 1, \\
&\quad \Pi_t = \pi_t) \\
&\quad = \sum_{s_i \in S_t, \varphi_l \in \Phi_t} P(S_{t+1} = s_{i'}, \Phi_{t+1} = \varphi_{i'}, \hat{Y}_{t+1}^a = \hat{y}_k^a \mid \\
&\qquad\qquad V_{t+1} = a, Y_{t+1}^a = 1, S_t = s_i, \Phi_t = \varphi_l) \\
&\qquad\qquad \cdot P(S_t = s_i, \Phi_t = \varphi_l \mid \Pi_t = \pi_t),
\end{aligned} \tag{27}
$$

where $P(S_t = s_i, \Phi_t = \varphi_l \mid \Pi_t = \pi_t) = \pi_t^{il}$. The MaxEntropy policy κ^M then chooses the I alerts that, if investigated, would lead to the highest reduction of the entropy of the belief. For this, the policy ranks the alerts in decreasing order of $H(S_{t+1}, \Phi_{t+1}, \hat{Y}_{t+1}^a \mid Y_{t+1}^{a'} = 1, V_{t+1} = a', \Pi_t = \pi_t)$ and greedily chooses the first I alerts.

B. Bayes factor policy

For an alert vector y_n, the Bayes factor policy computes for every $a \in \{a' | y_n^{a'} = 1\}$ the likelihood ratio of the alert being

[1] For a discrete random variable $X \in \mathcal{D}$ the entropy is defined as $H(X) = \sum_{x \in \mathcal{D}} p_x \log p_x$, where $p_x = P(X = x)$.

a true positive under the hypothesis that $\zeta_a = 0$ (i.e., no false positives) vs. it being a false positive, i.e.,

$$K^a = \frac{P(Y^a_{t+1} = 1 \mid Y^{-a}_{t+1} = y^{-a}_n, \Pi_t = \pi_t)|_{\zeta_a=0}}{\zeta_a}, \tag{28}$$

where $-a$ is the set of alert indices except for the alert index a (i.e., $-a = \mathcal{A} \setminus \{a\}$), Y^{-a}_{t+1} is the alert vector except for the alert index a, and $P(Y^a_{t+1} = 1 \mid Y^{-a}_{t+1} = y^{-a}_n, \Pi_t = \pi_t)|_{\zeta_a=0} = \sum_{s_i \in S_t, \varphi_l \in \Phi_t} P(Y^a_{t+1} = 1 \mid Y^{-a}_{t+1} = y^{-a}_n, S_t = s_i, \Phi_t = \varphi_l)|_{\zeta_a=0} \cdot P(S_t = s_i, \Phi_t = \varphi_l \mid \Pi_t = \pi_t)$. Consider the probability of an alert given other alerts and current state

$$\begin{aligned} &P(Y^a_{t+1} = 1 \mid Y^{-a}_{t+1} = y^{-a}_n, S_t = s_i, \Phi_t = \varphi_l)|_{\zeta_a=0} \\ &= \frac{P(Y^a_{t+1} = 1, Y^{-a}_{t+1} = y^{-a}_n \mid S_t = s_i, \Phi_t = \varphi_l)|_{\zeta_a=0}}{P(Y^{-a}_{t+1} = y^{-a}_n \mid S_t = s_i, \Phi_t = \varphi_l)} \\ &= \frac{\begin{array}{r}\sum_{\mathcal{E}_t \in \mathscr{P}(\mathcal{E}^+(s_i))} P(Y_{t+1} = y_n \mid E_t = \mathcal{E}_t, \Phi_t = \varphi_l)|_{\zeta_a=0} \\ \cdot P(E_t = \mathcal{E}_t \mid S_t = s_i, \Phi_t = \varphi_l)\end{array}}{\begin{array}{r}\sum_{\mathcal{E}_t \in \mathscr{P}(\mathcal{E}^+(s_i))} P(Y^{-a}_{t+1} = y^{-a}_n \mid E_t = \mathcal{E}_t, \Phi_t = \varphi_l) \\ \cdot P(E_t = \mathcal{E}_t \mid S_t = s_i, \Phi_t = \varphi_l)\end{array}}, \end{aligned} \tag{29}$$

where a is the index of a raised alert such that $a \in \{a' | y^{a'}_n = 1\}$, and $P(E_t = \mathcal{E}_t \mid S_t = s_i, \Phi_t = \varphi_l)$ is given by eqn. (13).

Considering that each alert y^a_n is independent of other alerts y^{-a}_n given the exploits $\mathcal{E}_t$ and the attacker type φ_l, we can express the above probabilities as

$$\begin{aligned} &P(Y^{-a}_{t+1} = y^{-a}_n \mid E_t = \mathcal{E}_t, \Phi_t = \varphi_l) \\ &\quad = \prod_{a' \in \mathcal{A} \setminus \{a\}} P(Y^{a'}_{t+1} = y^{a'}_n \mid E_t = \mathcal{E}_t, \Phi_t = \varphi_l), \end{aligned} \tag{30}$$

$$\begin{aligned} &P(Y_{t+1} = y_n \mid E_t = \mathcal{E}_t, \Phi_t = \varphi_l)|_{\zeta_a=0} \\ &\quad = P(Y^a_{t+1} = 1 \mid E_t = \mathcal{E}_t, \Phi_t = \varphi_l)|_{\zeta_a=0} \\ &\qquad \cdot\ P(Y^{-a}_{t+1} = y^{-a}_n \mid E_t = \mathcal{E}_t, \Phi_t = \varphi_l) \\ &\quad = \Big(1 - \prod_{e_x \in \mathcal{E}_t \cap \mathcal{E}(z_a)} (1 - \delta_{xa}(\varphi_l))\Big) \\ &\qquad \cdot \prod_{a' \in \mathcal{A} \setminus \{a\}} P(Y^{a'}_{t+1} = y^{a'}_n \mid E_t = \mathcal{E}_t, \Phi_t = \varphi_l), \end{aligned} \tag{31}$$

where each term in eqn. (30) and eqn. (31) is given by eqn. (14). Note that $K^a = 1$ stands for the highest uncertainty of an alert. Then, the Bayes factor policy κ^B ranks the alerts in increasing order of $(K^a - 1)^2$ and chooses the first I alerts.

C. Complexity analysis

Recall that we denote by n_z the number of alerts and by n_S the number of possible states. Also, let us denote by N_E the maximum number of available exploits among all states (i.e., $N_E = \max_{s \in S} |\mathcal{E}^+(s)|$). The Bayes factor policy considers all possible exploits in each state. Thus, given a state, the policy computes the power set of exploits to consider all possible exploits and require $\mathcal{O}(2^{N_E})$ computations. We repeat the computation for all states and all alerts and thus obtain $\mathcal{O}(n_z n_S 2^{N_E})$. In addition, MaxEntropy factors all combinations of the successes and failures of exploits $\mathcal{E}_t \subseteq \mathcal{E}^+(s)$ by using the power set. Thus, the worst case is $\mathcal{E}_t = \mathcal{E}^+(s)$, and the complexity of MaxEntropy is $\mathcal{O}(n_z n_S 2^{2N_E})$.

TABLE I
TRUE AND FALSE ALERT RATES USED FOR THE EVALUATION.

True Alert Rate ($\delta_{xa}(\varphi_l)$)								
Exploit	**Alert Index (a)**							
Index (x)	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	0.8	0	0	0	0	0	0	0
2	0.1	0.6	0	0	0	0	0	0
3	0	0.8	0	0	0	0	0	0
4	0	0	0.5	0	0	0	0	0
5	0	0	0	0.7	0	0	0	0
6	0	0	0.6	0	0	0	0	0
7	0	0	0.1	0.7	0	0	0	0
8	0	0	0	0	0.7	0	0	0
9	0	0	0	0	0.6	0.4	0	0
10	0	0	0	0	0	0.7	0	0
11	0.4	0.6	0	0	0	0	0	0
12	0	0	0	0	0	0	0.7	0
13	0	0	0	0	0	0	0	0.8
False Alert Rate (ζ_a)								
	0.25	0.3	0.25	0.3	0.25	0.4	0.35	0.3

The belief update in eqn. (6) consists of $p^n_{i'l'}(\pi_t)$, $r^v_{i'nkl'}(\pi_t)$, and $\sigma(\pi_t, \hat{y}_k, y_n, v)$. However, we derive $\sigma(\pi_t, \hat{y}_k, y_n, v)$ by summing the numerator. Thus, we focus on the terms in the numerator. The terms $p^n_{i'l'}(\pi_t)$ and $r^v_{i'nkl'}(\pi_t)$ consider all possible successes and failures of exploits $\mathcal{E}_t \subseteq \mathcal{E}^+(s)$. In the worst case $\mathcal{E}_t = \mathcal{E}^+(s)$, and we consider all possible successes and failures of N_E exploits and all possible exploits given a state, at complexity $\mathcal{O}(2^{2N_E})$. This computation is repeated for all states, resulting in $\mathcal{O}(n_S 2^{2N_E})$ computations. Each term in the numerator requires such a computation; hence the belief update requires $\mathcal{O}(n_S 2^{N_E+1})$ computations. Since $2^{N_E+1} \gg n_z$, we can state the computational burden of the Bayes factor policy is relatively low compared to that of the belief update.

VI. NUMERICAL RESULTS

We evaluated the proposed policies on the dependency graph shown in Fig. 2. For the evaluation, we consider two attacker types (i.e., $|\Phi| = 2$), where the first type and second type are a benign user and a malicious attacker, respectively. The benign user does not use any exploits ($\alpha_{e_x}(\varphi_1) = 0 \ \ \forall e_x \in \mathcal{E}$), while the attacker chooses exploits with $\alpha_{e_x}(\varphi_2) = 0.3$ for $e_x \in \mathcal{E}_0$ and $\alpha_{e_x}(\varphi_2) = 0.2$ for $e_x \in \mathcal{E} \setminus \mathcal{E}_0$. After choosing the exploits, the exploits succeed with probability $\beta_{e_x}(\varphi_l) = 0.3$ for $e_x \in \mathcal{E}_0$ and $\beta_{e_x}(\varphi_l) = 0.2$ for $e_x \in \mathcal{E} \setminus \mathcal{E}_0$ and for all attacker types $\varphi_l \in \Phi$. The exploits trigger alerts as shown in Table I regardless of the attacker type; the alert rates are as in [11]. To simulate the defender's limited resources (e.g., the limited number of security analysts and time), we define an investigation budget $I = 1$ which stands for the number of alerts the defender investigate in each step. Given the experimental settings, we observed that the MSE without any investigation reaches its maximum at $t = 43$ on average. Thus, by rounding the time step up, we performed each attack simulation for 50 time steps, and the results shown are the

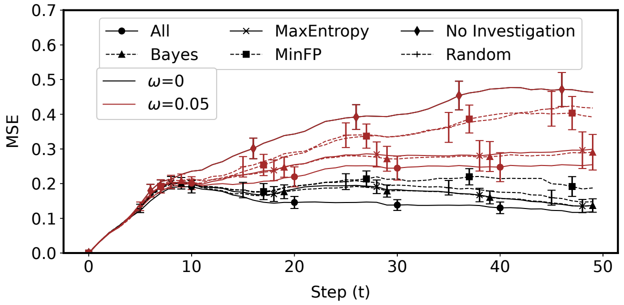

Fig. 3. MSE of attack state belief as a function of time ($0 \leq t \leq 50$) for $I = 1$ with 95% confidence intervals.

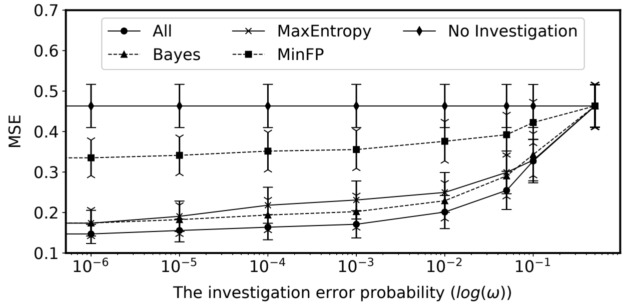

Fig. 4. MSE of attack state belief as the investigation error probability ω at $t = 50$, $I = 1$ with 95% confidence intervals.

averages of 100 simulations. In each simulation, the defender's initial belief π_0 has the uniform distribution over attacker types, assuming the network starts with the safe environment $s_1 = \emptyset$. Also, we focus on the simulation with a malicious attacker φ_2 since we observed low MSEs (i.e., less than 0.08) from the simulations with the benign user φ_1 using different investigation policies.

As baselines for comparison, we consider four policies. Using the first policy, the defender investigates all alerts, providing a lower bound on the achievable MSE. We refer to this as *All*. Using the second policy, the defender investigates the alert with the lowest false positive rate, referred to as *MinFP*. Using the third policy, the defender investigates an alert chosen at random, referred to as *Random*. Finally, under the *No Investigation* policy, the attacker does not investigate any alerts, providing an upper bound on the MSE. For a numerical comparison of policies, we use the reduction of normalized mean absolute error (RNMAE) to measure the reduction of MSE of a policy κ,

$$\text{RNMAE}_t^{\kappa} = \frac{MSE(\pi_t^{NI}, S_t) - MSE(\pi_t^{\kappa}, S_t)}{MSE(\pi_t^{NI}, S_t) - MSE(\pi_t^{All}, S_t)}, \tag{32}$$

where π_t^{κ} is the belief at time step t, and *NI* and *All* stand for the *No Investigation* and the *All* policy, respectively.

Figure 3 shows the MSE as a function of time for the considered policies for two values of the investigation error probability ω. The figure shows that the MSE increases sharply during the first few time steps for all policies, which we attribute to the confusion matrix of the IDS shown in Table I, where alerts $a \in \{1, 2\}$ can be shared by multiple initial exploits $e_x \in \mathcal{E}_0$, hence it is hard initially to infer the hidden state based on the alerts. The results are, however, very distinct after $t = 8$. Without investigation, the MSE continues to increase throughout the simulation. On the contrary, using the proposed policies, the MSE remains nearly constant or decreases slightly depending on the investigation error probability ω. Overall, the *MaxEntropy* and *Bayes* policies outperform the *Random* and the *MinFP* policy significantly, especially when $\omega > 0$.

TABLE II
RNMAE OF POLICIES AT $t = 50$.

	Bayes	MaxEntropy	MinFP	Random
$\omega = 0$	94.81%	94.83%	80.11%	91.63%
$\omega = 0.05$	82.84%	78.67%	30.06%	21.72%

Table II shows the RNMAE of the different policies computed using eqn. (32) at $t = 50$. The table shows that the RNMAE achieved by *MaxEntropy* and *Bayes* in comparison to *MinFP* and *Random* is significantly higher, up to 50% for $\omega > 0$, which indicates the importance of the dynamic prioritization of alerts and the superior performance of the proposed policies compared to static prioritization (*MinFP*). We can also observe that *MinFP* performs worse than *Random* for $\omega = 0$, highlighting the potential danger of static prioritization. Overall, *Random* and *MinFP* show similar performance, hence in what follows we use *MinFP* as the baseline.

Figure 4 shows the MSEs at $t = 50$ as a function of the investigation error probability ω. Note that $\omega = 0.5$ implies the investigation adds no information, and is thus equivalent to *No Investigation*. We observe that the MSE of the *All* policy, which serves as a lower bound, increases smoothly with the increase in the error probability ω, and so does the MSE of the proposed policies. On the one hand, the figure highlights the importance of the defender's certainty regarding the investigation results. On the other hand, it also shows that investigation error probability causes a graceful degradation of the state estimate. Table III illustrates the RNMAE achieved by the proposed policies (i.e., *MaxEntropy* and *Bayes*) compared to *No Investigation* for different values of the investigation error probability ω. The results show that *Bayes* mostly achieves a higher RNMAE than *MaxEntropy*. Since its computation burden is lower as well, we argue that *Bayes* should be the preferred method for dynamic alert prioritization if the investigation error probability is non-negligible.

Figure 5 shows the MSEs at $t = 50$ as a function of the investigation budget I. To obtain meaningful results for high values of I, we increased the false positive (FP) rate ζ_a of all alerts by 0.2 for the evaluation, hence there are 4 false alerts on average per time step. While it is no surpise that *No Investigation* is insensitive to the investigation budget, we observe that the proposed policies require significantly lower investigation budget ($I = 2$ instead of $I = 5$ and $I = 6$ for $\omega = 0$ and $\omega = 0.05$, respectively) for performing as well as the *All* policy than *MinFP*, which shows that our policies can

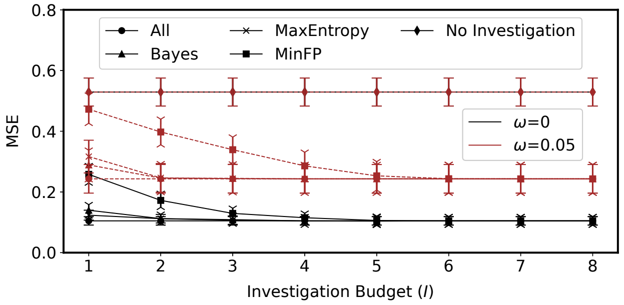

Fig. 5. MSEs of the belief as a function of the investigation budget I at $t = 50$ with 95% confidence intervals.

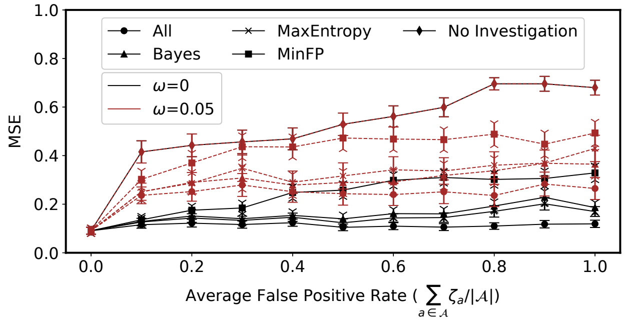

Fig. 6. MSEs of the belief as a function of the false positive rate for $I = 1$ at $t = 50$ with 95% confidence intervals.

identify the most informative alerts to investigate.

Figure 6 shows the MSEs at $t = 50$ as a function of the average false positive rates. We set the average false positive rate by decreasing/increasing the false positive rates shown in Table I uniformly (the default average FP rate is 0.3). We observe a significant degradation of the belief's accuracy when using *MinFP* as the FP rate increases, while *MaxEntropy* and *Bayes* exhibit a performance relatively close to *All*. We can thus conclude that our policies can efficiently choose the most informative alert, as a function of the system state, despite high FP rates, and doing so is sufficient to perform almost as well as when investigating all alerts, at much lower cost.

VII. Conclusion

In this paper, we considered the problem of dynamic alert prioritization for maintaining SA based on noisy alerts from an IDS. We proposed a novel formulation of the problem, in the form of active learning for estimating the state of a HMM, taking into account that queries may be error prone. We proposed two policies that rely on the uncertainty of the belief for minimizing the state estimation error. Our simulations showed that the proposed policies achieve significant improvements compared to static baseline policies, at moderate computational overhead. Interesting directions of future work include learning a succinct representation of the dependency graph to improve scaling while maintaining explainability, and incorporating containment and eradication actions.

Acknowledgement

The work was partly funded by the Swedish Foundation for Strategic Research through the CLAS project (grant RIT17-0046), the Swedish Research Council through project 2020-03860, and by KTH Digital Futures through the CLAIRE project.

TABLE III
RNMAE of policies for different ω at $t = 50$, $I = 1$.

ω	0.0	1×10^{-6}	1×10^{-5}	1×10^{-4}
Bayes	94.81%	91.27%	89.95%	89.30%
MaxEntropy	94.83%	91.56%	88.61%	81.89%
ω	1×10^{-3}	1×10^{-2}	5×10^{-2}	1×10^{-1}
Bayes	89.30%	89.36%	82.84%	88.34%
MaxEntropy	79.48%	81.49%	78.67%	98.50%

References

[1] R. A. Sadek, M. S. Soliman, and H. S. Elsayed, "Effective anomaly intrusion detection system based on neural network with indicator variable and rough set reduction," *International Journal of Computer Science Issues (IJCSI)*, vol. 10, no. 6, p. 227, 2013.

[2] A. Schlenker, H. Xu, M. Guirguis, C. Kiekintveld, A. Sinha, M. Tambe, S. Sonya, D. Balderas, and N. Dunstatter, "Don't bury your head in warnings: A game-theoretic approach for intelligent allocation of cyber-security alerts," in *Proc. of IJCAI*, 2017.

[3] N. Dunstatter, M. Guirguis, and A. Tahsini, "Allocating security analysts to cyber alerts using markov games," in *National Cyber Summit (NCS)*. IEEE, 2018, pp. 16–23.

[4] A. Shah, R. Ganesan, S. Jajodia, and H. Cam, "Dynamic optimization of the level of operational effectiveness of a csoc under adverse conditions," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 5, pp. 1–20, 2018.

[5] W. Tounsi and H. Rais, "A survey on technical threat intelligence in the age of sophisticated cyber attacks," *Computers & Security*, vol. 72, pp. 212–233, 2018.

[6] O. P. Kreidl and T. M. Frazier, "Feedback control applied to survivability: a host-based autonomic defense system," *IEEE Transactions on Reliability*, vol. 53, no. 1, pp. 148–166, 2004.

[7] S. A. Zonouz, H. Khurana, W. H. Sanders, and T. M. Yardley, "Rre: A game-theoretic intrusion response and recovery engine," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 395–406, 2013.

[8] S. Iannucci, Q. Chen, and S. Abdelwahed, "High-performance intrusion response planning on many-core architectures," in *International Conference on Computer Communication and Networks (ICCCN)*, 2016, pp. 1–6.

[9] S. Iannucci and S. Abdelwahed, "A probabilistic approach to autonomic security management," in *Proc. of IEEE International Conference on Autonomic Computing (ICAC)*, 2016, pp. 157–166.

[10] E. Miehling, M. Rasouli, and D. Teneketzis, "Optimal defense policies for partially observable spreading processes on bayesian attack graphs," in *Proc. of ACM Workshop on Moving Target Defense*, 2015, pp. 67–76.

[11] ——, "A pomdp approach to the dynamic defense of large-scale cyber networks," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2490–2505, 2018.

[12] B. Anderson and M. Andrew, "Active learning for hidden markov models: Objective functions and algorithms," in *Proc. of Intl. Conf. on Machine Learning (ICML)*, 2005.

[13] P. Ammann, D. Wijesekera, and S. Kaushik, "Scalable, graph-based network vulnerability analysis," in *Proc. of ACM Conference on Computer and Communications Security (CCS)*, 2002, pp. 217–224.

[14] S. Jajodia, S. Noel, and B. O'berry, "Topological analysis of network attack vulnerability," in *Managing cyber threats*. Springer, 2005, pp. 247–266.