Kazeminajafabadi and Imani *Cybersecurity (2023)6:22*
https://doi.org/10.1186/s42400-023-00155-y

Cybersecurity

RESEARCH

Open Access

Optimal monitoring and attack detection of networks modeled by Bayesian attack graphs

Armita Kazeminajafabadi[1*] and Mahdi Imani[2]

Abstract

Early attack detection is essential to ensure the security of complex networks, especially those in critical infrastructures. This is particularly crucial in networks with multi-stage attacks, where multiple nodes are connected to external sources, through which attacks could enter and quickly spread to other network elements. Bayesian attack graphs (BAGs) are powerful models for security risk assessment and mitigation in complex networks, which provide the probabilistic model of attackers' behavior and attack progression in the network. Most attack detection techniques developed for BAGs rely on the assumption that network compromises will be detected through routine monitoring, which is unrealistic given the ever-growing complexity of threats. This paper derives the optimal minimum mean square error (MMSE) attack detection and monitoring policy for the most general form of BAGs. By exploiting the structure of BAGs and their partial and imperfect monitoring capacity, the proposed detection policy achieves the MMSE optimality possible only for linear-Gaussian state space models using Kalman filtering. An adaptive resource monitoring policy is also introduced for monitoring nodes if the expected predictive error exceeds a user-defined value. Exact and efficient matrix-form computations of the proposed policies are provided, and their high performance is demonstrated in terms of the accuracy of attack detection and the most efficient use of available resources using synthetic Bayesian attack graphs with different topologies.

Keywords Multi-stage attacks, Bayesian attack graph, Attack detection, Optimal monitoring

Introduction

The increased connectivity of networks and smart devices allow for effective operations of complex networks while significantly weakening network security (Lallie et al. 2020; Ou et al. 2006; Wang et al. 2018; Al Ghazo et al. 2019; Al-Araji et al. 2022; Nguyen et al. 2017). In particular, the operation of critical infrastructures such as manufacturing, energy, communication, water, and transportation networks increasingly rely on networked devices, generating significant vulnerabilities in many areas of society.

*Correspondence:
Armita Kazeminajafabadi
kazeminajafabadi.a@northeastern.edu
[1] Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA
[2] Department of Electrical and Computer Engineering, Northeastern University, Boston, MA, USA

Attack graphs are a useful model to characterize the interactions and dependencies between vulnerabilities across the network components (Noel and Jajodia 2014; Singhal and Ou 2017; Stan et al. 2020; Noel and Jajodia 2017; Capobianco et al. 2019; Agmon et al. 2019; Malzahn et al. 2020; Albanese et al. 2012; Homer et al. 2013; Yu et al. 2015; Munoz Gonzalez and Lupu 2016). These graphs model how attackers can exploit combinations of vulnerabilities to penetrate networks. Bayesian attack graphs (BAGs) are extensions of attack graphs, where the Bayesian network probabilistically models attackers' behavior and progression of attacks across the network (Poolsappasit et al. 2011; Muñoz-González et al. 2017; Sembiring et al. 2015; Miehling et al. 2015; Hu et al. 2017; Matthews et al. 2020; Sahu and Davis 2021; Frigault et al. 2017; Chen et al. 2021; Chockalingam et al. 2017; Sun et al. 2018; Liu et al. 2019). BAGs are directed graphs consisting of nodes that represent the

status of compromises at various network components and edges that represent exploit probabilities among the components.

Most existing attack detection techniques developed for BAGs rely on the simplified assumption that the network's compromises are certainly detectable through routine monitoring (Li et al. 2020; Chadza et al. 2020; Holgado et al. 2017; Thanthrige et al. 2016; Ramaki et al. 2015). However, given the ever-growing types of attacks and the intelligence of attackers to hide the exploit, this assumption is unrealistic and leads to the unreliability of detection. Meanwhile, the existing detection methods often built upon heuristics (Poolsappasit et al. 2011; Alhomidi and Reed 2013; Husák et al. 2018) or approximations (Liu and Liu 2016; Wang et al. 2013; Ma et al. 2022). These methods do not yield the optimality expected for these structured graphs, such as minimum mean square error (MMSE) or component-wise optimality. This paper derives the exact optimal MMSE attack detection method for a general form of BAGs with partial and imperfect monitoring and arbitrary network vulnerabilities. The binary structure of the nodes on the graph (denoting the compromised status of network components) is taken into account to achieve the same MMSE optimality as the Kalman filter for the linear Gaussian state space model (Liang et al. 2019; Bai et al. 2017). We demonstrate that the proposed detection method also holds the component-wise maximum aposteriori optimality, which differs from the commonly used maximum aposteriori solution obtained for the entire nodes.

The second contribution of this paper is to derive an exact optimal policy to select a subset of monitoring nodes at any given time to enhance the performance of the detection process. In practice, a few nodes in the network can be routinely monitored due to resource limitations and reducing potential disruptions to network operations. Intelligent selection of these nodes plays a crucial role in accurately detecting attacks over the network. For instance, monitoring a fixed set of nodes could significantly degrade detection performance at unobserved components. Therefore, it is critical to sequentially and strategically select nodes for monitoring and making the best use of available resources.

Several monitoring approaches have been developed for Bayesian attack graphs, including Monte Carlo and probabilistic methods. The Monte Carlo or tree-based approaches (Noel and Jajodia 2008; Krisper et al. 2019; Poolsappasit et al. 2011) simulate the most likely attack paths and sequentially select monitoring nodes located on these paths. The probabilistic vulnerability assessment approaches (Dantu et al. 2004; Nipkow et al. 2012; Frigault and Wang 2008) measure the expected increase in the probability of compromise at various nodes and select those with the highest overall vulnerabilities. These methods mostly rely on heuristics for their selection or some simulated attack paths, which makes them inefficient in securing complex networks with uncertain monitoring and limited available resources. Meanwhile, existing techniques take into account the network vulnerability of nodes for selecting monitoring nodes rather than accurate detection and identifying invisible compromises in the network.

This paper presents an optimal monitoring policy that supports the optimal detection policy and ensures the selection of monitoring nodes that are most likely to be incorrectly detected. The proposed monitoring method selects the optimal subset of nodes for monitoring sequentially based on the highest expected predictive mean squared error (MSE). Instead of selecting nodes that are already compromised or uncompromised, we have developed fixed-resource and adaptive-resource monitoring policies that select a subset of nodes sequentially to ensure the best detectability of attacks across the entire network. Depending on the network's vulnerabilities or the sensitivity of its components, the appropriate monitoring policy can prioritize network detectability at specific parts of the network rather than all components. We introduce efficient and exact matrix-form solutions for attack detection and network monitoring policies and demonstrate the performance of the methods using several synthetic Bayesian attack graphs.

The article is organized as follows. First, the Bayesian attack graph model is briefly described. Then, the optimal attack detection and monitoring policies are derived, and their matrix-form implementations are introduced. Finally, the numerical examples and concluding remarks are provided.

Bayesian attack graphs (BAGs)

Bayesian attack graphs are a powerful class of models for the probabilistic representation of attackers' behavior and the progression of attacks on networks. The attackers aim to take over the entire network by exploiting reachable vulnerabilities, while each exploit only succeeds with a certain probability. A BAG is a directed graph where the nodes of the graph represent the compromises' status at each network component (i.e., 1 for compromised nodes and 0 for not compromised nodes), and edges represent the likelihood that a compromised node could successfully expose a neighboring component.

A BAG is defined as a tuple (Hu et al. 2020)

$$\mathcal{G} = (\mathcal{N}, \mathcal{T}, \mathcal{E}, \mathcal{P})$$

where $\mathcal{N} = \{1, \cdots, n\}$ represents n elements (nodes) of the network, $\mathcal{T}$ is the set of node types, $\mathcal{E}$ is the set of

directed edges between the nodes, and $\mathcal{P}$ is the set of exploit probabilities. The nodes are random variables taking in $\{0, 1\}$, where 0 and 1 indicate that a given component is not compromised and compromised, respectively. For simplicity and without loss of generality, each node is assumed to be one of the following two types: $\mathcal{T}_i \in \{\text{AND}, \text{OR}\}$, where $\mathcal{T}_i$ represents the type of the ith component. The edge $(i, j) \in \mathcal{E}$ represents if node j could be compromised through node i. $\mathcal{P}$ consists of the set of exploit probabilities associated with edges, where $\rho_{ij} \in \mathcal{P}$ represents the probability that the node j can be compromised through node i, given that node i is already compromised. These exploit probabilities are often computed according to the NIST's Common Vulnerability Scoring System (CVSS), which characterizes the severity of vulnerabilities through numerical scores (Radack et al. 2007).

Node i is an in-neighbor of node j if $(i, j) \in \mathcal{E}$. The in-neighbor set of node j can be formally defined as: $D_j = \{i \in \mathcal{N} | (i, j) \in \mathcal{E}\}$. The nodes connected to outside sources are susceptible to external attacks. The external attack on node j can be expressed in terms of the exploit probability ρ_j. As mentioned before, there are two types of nodes; an AND node (e.g., admin servers) could get compromised only if all of its in-neighbor nodes are compromised, while an OR node (e.g., SQL servers) could get compromised through a single (or more) compromised in-neighbor(s).

An example of the Bayesian attack graph is shown in Fig. 1. The graph consists of 20 nodes; the nodes that are exposed to external attacks include $\{2, 5, 6, 8\}$. AND nodes illustrated as double encircled nodes are $\{2, 3, 6, 9, 10, 13, 18, 19\}$, and OR nodes are $\{1, 4, 5, 7, 8, 11, 12, 14, 15, 16, 17, 20\}$. Exploit probabilities are labeled only for node 1 for simplicity.

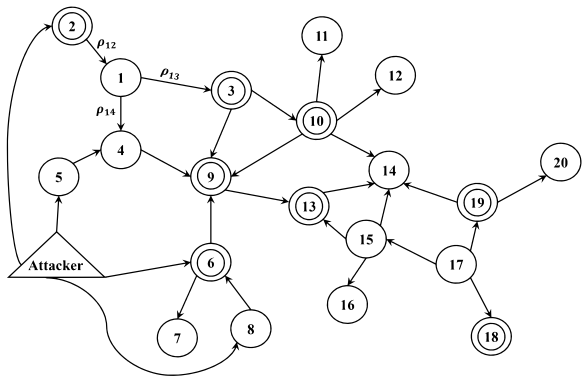

Fig. 1 An example of a Bayesian attack graph

Optimal attack detection for BAGs

Hidden Markov model (HMM) representation of BAG

The BAG can be seen as a special case of a hidden Markov model with binary state variables. The state vector consists of the status of compromises at all n nodes in the graph. This vector is represented by $\mathbf{x}_k = [\mathbf{x}_k(1), ..., \mathbf{x}_k(n)]$, where $\mathbf{x}_k(i)$ takes either 0 or 1; $\mathbf{x}_k(i) = 1$ indicates that the ith component is compromised at time step k, and reverse for $\mathbf{x}_k(i) = 0$. $\mathbf{x}_k = [0, 0, \cdots, 0]^T$ represents a network without any compromise, whereas $\mathbf{x}_k = [1, 1, \cdots, 1]^T$ represents network with all nodes being compromised. Therefore, the state vector can take 2^n different possible values, denoted by $\{\mathbf{x}^1, \cdots, \mathbf{x}^{2^n}\}$. The HMM representation of BAG, consisting of the state and observation processes, is described below.

State process The state process represents the probabilistic propagation of compromises at all nodes. This process can be expressed through the conditional probability distribution of states. The state process is governed by the probability of external attacks, exploit probabilities among nodes, and their types. For instance, the AND nodes are more robust against a single in-neighbor threat since the exploits at all in-neighbor nodes are required to give a chance for an AND node to be compromised. On the other hand, the OR nodes can be compromised if a single in-neighbor node is compromised. In the same way, large exploit and external attack probabilities increase the network's vulnerability.

The conditional probability that the jth node is compromised at time step k, given the nodes' state at time step $k-1$, i.e., $\mathbf{x}_{k-1}$, can be expressed for AND and OR nodes as:

- *AND nodes*:

$$P(\mathbf{x}_k(j) = 1 | \mathbf{x}_{k-1}) = \begin{cases} \rho_j + (1 - \rho_j) \prod\limits_{i \in D_j} 1_{\mathbf{x}_{k-1}(i)=1} \rho_{ij} & \text{if } \mathbf{x}_{k-1}(j) = 0, \\ 1 & \text{if } \mathbf{x}_{k-1}(j) = 1, \end{cases} \tag{1}$$

- *OR Nodes:*

$$P(\mathbf{x}_k(j) = 1 | \mathbf{x}_{k-1}) = \begin{cases} \rho_j + (1 - \rho_j) \left[1 - \prod\limits_{i \in D_j} (1 - 1_{\mathbf{x}_{k-1}(i)=1} \rho_{ij}) \right] & \text{if } \mathbf{x}_{k-1}(j) = 0, \\ 1 & \text{if } \mathbf{x}_{k-1}(j) = 1, \end{cases} \tag{2}$$

where $1_{b=1}$ returns 1 if $b = 1$, and 0 otherwise. Note that the conditional probabilities in (1) and (2) consider both the external (i.e., ρ_j) and internal (i.e., ρ_{ij}) attacks. Meanwhile, using the binary nature of each state

variable, the probability that the jth state variable is 0 can be computed as: $P(\mathbf{x}_k(j) = 0|\mathbf{x}_{k-1}) = 1 - P(\mathbf{x}_k(j) = 1|\mathbf{x}_{k-1})$.

Observation process: This process represents the way network components are monitored for potential threats. In practice, routine network monitoring is key in assuring network security and possibly detecting compromises in the network. The monitoring process is often labor-intensive, time-consuming, and costly, which might also interrupt or delay the network operations. Hence, a small subset of nodes can be monitored at any given time. Most available detection techniques for BAGs assume that possible network compromise at any given node is certainly identified if the node is selected for routine monitoring. However, given the complexity of attacks/attackers, this assumption is likely to be violated, resulting in significant security risks in detecting attacks. For instance, the monitoring might flag a node as not compromised while the node is compromised with an advanced difficult-to-detect attack.

Let $\mathbf{a}_{k-1} = \{i_1, ..., i_m\}$ be the indexes of m nodes to be monitored at time step k, where $\{i_1, ..., i_m\} \subset \mathcal{N}$ and $m < n$. As indicated in the subscripts, the nodes should be selected at time step $k-1$ for monitoring at time step k. The observation resulting from $\mathbf{a}_{k-1}$ is denoted by $\mathbf{y}_k$, where $\mathbf{y}_k(i)$ is the observation from node $\mathbf{a}_{k-1}(i)$.

We consider the following model for the observation process: (1) if the selected node for monitoring is not compromised, the observation will flag not compromised with a probability of 1; (2) if the selected node is compromised, the true compromised node will be detected with probability $(1-q)$ and will be flagged as not compromised with probability q, where $0 \leq q \leq 1$. Therefore, if the observation from a node is 1 (i.e., flagged as "compromised"), it is definitely intruded; however, observing 0 (i.e., flagged as "not compromised") does not provide certain information about the status of compromises in the monitored node. This stochastic observation model can significantly enhance the reliability and performance of attack detection. The observation process described above can be expressed at time step k as:

$$\mathbf{y}_k(i) = \begin{cases} 1 \text{ if } \mathbf{x}_k(a_{k-1}(i)) = 1 & \text{w.p. } 1-q \\ 0 \text{ if } \mathbf{x}_k(a_{k-1}(i)) = 1 & \text{w.p. } q \\ 0 \text{ if } \mathbf{x}_k(a_{k-1}(i)) = 0 & \text{w.p. } 1 \end{cases}, \tag{3}$$

for $i = 1, ..., m$. Small values of q model an advanced monitoring system where most threats can be identified, whereas larger values of q correspond to the less advanced monitoring systems or domains susceptible to more complex threats. It should be noted that the rest of the paper holds for any arbitrary observation process of form $\mathbf{y}_k \sim P(\mathbf{y}|\mathbf{x}_k, \mathbf{a}_{k-1})$, other than (3).

Optimal MMSE attack detection for BAGs

Accurate attack detection is crucial for effectively identifying compromises in network components and taking necessary steps to secure the network against potential threats. Attack detection is often challenging due to the probabilistic nature of attack progression and partial and imperfect monitoring of network components. The existing attack detection methods do not fully account for imperfect monitoring of networks and are built upon commonly used criteria for finite-state HMMs, such as maximum aposteriori or maximum likelihood (Liu and Liu 2016; Wang et al. 2013; Ma et al. 2022). Inspired by the Kalman filtering approach (Welch et al. 1995), which provides the exact optimal minimum mean square error (MMSE) state estimation solution for linear and additive-Gaussian state space models, this paper derives the exact optimal MMSE attack detection solution for the general form of BAGs with arbitrary distributions. It should be noted that the proposed detectors, described below, are the only exact MMSE detection techniques for the entire non-linear and non-Gaussian state space models (Särkkä 2013).

Let $\mathbf{a}_{0:k-1} = (\mathbf{a}_0, ..., \mathbf{a}_{k-1})$ be the selected monitoring nodes with associated observations $\mathbf{y}_{1:k} = (\mathbf{y}_1, , ...\mathbf{y}_k)$ between time step 1 to k. The attack detection problem consists of estimating the state values of all nodes at time step r given $\{\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}\}$. Note that depending on the objective, the detection time r can be the current (i.e., $r = k$), prior (i.e, $r < k$), or future (i.e., $r > k$) time step. A detected attack $\hat{\mathbf{x}}_{r|k} = [\hat{\mathbf{x}}_{r|k}(1), ..., \hat{\mathbf{x}}_{r|k}(n)]^T$ represents the estimated value of the true attacks (i.e., compromises) at all nodes $\mathbf{x}_r = [\mathbf{x}_r(1), ..., \mathbf{x}_r(n)]^T$ at time step k. The optimal attack detector can be obtained by minimizing the following mean squared error (MSE):

$$\hat{\mathbf{x}}^{\text{MS}}_{r|k} = \underset{\hat{\mathbf{x}}_{r|k} \in \Psi}{argmin}\, \mathbb{E}\big[\|\mathbf{x}_r - \hat{\mathbf{x}}_{r|k}\|_2 | \mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}\big], \tag{4}$$

where $\|.\|_2$ is the L_2 norm vector and $\Psi := \{0,1\}^n$ is the set of all 2^n possible compromise estimators.

Note that, for a Boolean vector $|\mathbf{z}|$, the L_1 and L_2 norms are the same, i.e., $\|\mathbf{z}\|_2 = \|\mathbf{z}\|_1 = \sum_{i=1}^{n} |\mathbf{z}(i)|$. Thus, the minimization in (4) can be written as:

$$\begin{aligned} \hat{\mathbf{x}}^{\text{MS}}_{r|k} &= \underset{\hat{\mathbf{x}}_{r|k} \in \Psi}{argmin}\, \mathbb{E}\big[\|\mathbf{x}_r - \hat{\mathbf{x}}_{r|k}\|_1 | \mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}\big] \\ &= \underset{\hat{\mathbf{x}}_{r|k} \in \Psi}{argmin} \sum_{i=1}^{n} \mathbb{E}\big[|\mathbf{x}_r(i) - \hat{\mathbf{x}}_{r|k}(i)| | \mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}\big]. \end{aligned} \tag{5}$$

where the last expression is obtained by exchanging the summation and expectation. Each term contains an independent estimator for a given node; thus, the optimal MMSE attack detector needs to minimize

$\mathbb{E}[|\mathbf{x}_r(i) - \hat{\mathbf{x}}_{r|k}(i)||\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}]$, for all $i = 1, \ldots, n$. Given the binary nature of each state variable, the minimizer can be computed as:

$$\begin{aligned}\hat{\mathbf{x}}^{\mathrm{MS}}_{r|k}(i) &= \begin{cases} 1, \text{ if } \mathbb{E}[\mathbf{x}_r(i)|\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}] > 1/2, \\ 0, \text{ otherwise,} \end{cases} \\ &= \overline{\mathbb{E}[\mathbf{x}_r(i)|\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}]},\end{aligned} \tag{6}$$

for $i = 1, \ldots, n$, where $\overline{\mathbf{v}}(i) = 1$ if $\mathbf{v}(i) > 1/2$ and 0 otherwise, for any vector $\mathbf{v} \in [0, 1]^n$ and $i = 1, \ldots, n$.

Substituting (6) into (5) leads to the following optimal MMSE attack detector at time step r:

$$\hat{\mathbf{x}}^{\mathrm{MS}}_{r|k} = \overline{\mathbb{E}[\mathbf{x}_r|\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}]}. \tag{7}$$

The expected error of the attack detector in terms of the MSE can be computed as:

$$\begin{aligned}C^{\mathrm{MS}}_{r|k} &= \sum_{i=1}^{n} \mathbb{E}\Big[|\mathbf{x}_r(i) - \hat{\mathbf{x}}^{\mathrm{MS}}_{r|k}(i)||\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}\Big] \\ &= \sum_{i=1}^{n} P\Big(\mathbf{x}_r(i) \neq \hat{\mathbf{x}}^{\mathrm{MS}}_{r|k}(i)|\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}\Big).\end{aligned} \tag{8}$$

The ith element in summation in the last line of equation (8) can be expressed as:

$$\begin{aligned}&P\Big(\hat{\mathbf{x}}^{\mathrm{MS}}_{r|k}(i) \neq \mathbf{x}_r(i)|\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}\Big) \\ &= \begin{cases} 1 - \mathbb{E}[\mathbf{x}_r(i)|\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}] & \text{if } \mathbb{E}[\mathbf{x}_r(i)|\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}] > 1/2, \\ \mathbb{E}[\mathbf{x}_r(i)|\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}] & \text{otherwise.} \end{cases}\end{aligned} \tag{9}$$

Now, substituting (9) into (8) leads to

$$C^{\mathrm{MS}}_{r|k} = \frac{n}{2} - \sum_{i=1}^{n} \left| \mathbb{E}[\mathbf{x}_r(i)|\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}] - \frac{1}{2} \right|, \tag{10}$$

where the last expression in (10) is obtained by using $\min\{a, 1 - a\} = 1/2 - |a - 1/2|$, for $0 \leq a \leq 1$. Note that the $0 \leq C^{\mathrm{MS}}_{r|k} \leq n/2$, where the values close to 0 correspond to a small expected error of optimal attack detector, whereas large values correspond to a less confident detection process (i.e., larger expected error).

The following theorem summarizes the results of the optimal MMSE attack detector for the general form of BAGs.

Theorem 1 *Let $\mathbf{a}_{0:k-1}$ be selected monitoring nodes with associated observation $\mathbf{y}_{1:k}$ between time step 1 to k from a Bayesian attack graph. The exact optimal MMSE attack detector at time step r can be achieved as*:

$$\hat{\mathbf{x}}^{\mathrm{MS}}_{r|k} = \overline{\mathbb{E}[\mathbf{x}_r|\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}]}, \tag{11}$$

with the normalized optimal expected MSE

$$C^{\mathrm{MS}}_{r|k} = \frac{n}{2} - \sum_{i=1}^{n} \left| \mathbb{E}[\mathbf{x}_r(i)|\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}] - \frac{1}{2} \right|. \tag{12}$$

As noted before, the theorem provides the optimal detection for past, current, and future, depending on whether $r < k$, $r = k$, or $r > k$. In the next section, we will describe how the optimal attack prediction can help monitor vulnerable components of the network.

Exact matrix-based computation of optimal MMSE attack detector

This section introduces an algorithm for the exact computation of the optimal MMSE attack detector for BAGs. We put all possible network compromises in a single $n \times 2^n$ matrix as:

$$A = [\mathbf{x}^1, \ldots, \mathbf{x}^{2^n}], \tag{13}$$

where $\mathbf{x}^1$ to $\mathbf{x}^{2^n}$ are arbitrary enumerations of possible network compromises, e.g., $\mathbf{x}^1 = [0, 0, 0, ..., 0]^T$, $\mathbf{x}^{2^n} = [1, 1, 1, ..., 1]^T$. Consider the following state conditional distribution vectors:

$$\begin{aligned}\mathbf{\Pi}_{k|k}(i) &= P\Big(\mathbf{x}_k = \mathbf{x}^i|\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}\Big), \\ \mathbf{\Pi}_{k|k-1}(i) &= P\Big(\mathbf{x}_k = \mathbf{x}^i|\mathbf{a}_{0:k-2}, \mathbf{y}_{1:k-1}\Big),\end{aligned} \tag{14}$$

for $i = 1, \ldots, 2^n$ and $k = 1, 2, \ldots$. Let $\mathbf{\Pi}_{0|0}$ be the initial attack distribution. This distribution depends on the last time the nodes in the network have been re-imaged or monitored; for instance, $\mathbf{\Pi}_{0|0} = [1, 0, ..., 0]^T$ can be used for networks with recently re-imaged nodes, and $\mathbf{\Pi}_{0|0} = [1/2^n, ..., 1/2^n]^T$ can be used if not enough information about compromises at various nodes exists (i.e., each node with 0.5 probability being compromised). Note that more complex initial distributions can be used, such as larger compromise probabilities for nodes exposed to direct external attacks.

Let the *transition matrix* M_k of size $2^n \times 2^n$ be the transition matrix of the Markov chain at time step k as:

$$\begin{aligned}(M_k)_{ij} &= P(\mathbf{x}_k = \mathbf{x}^i|\mathbf{x}_{k-1} = \mathbf{x}^j) \\ &= \prod_{l=1}^{n} \Big(\eta_l^{ij}\, 1_{\mathbf{x}^i(l)=1} + (1 - \eta_l^{ij})\, 1_{\mathbf{x}^i(l)=0}\Big),\end{aligned} \tag{15}$$

for $i, j = 1, \ldots, 2^n$; where

$$\eta_l^{ij} = 1_{\mathbf{x}^j(l)=0}\left[\rho_l + (1-\rho_l)\prod_{r\in D_l} 1_{\mathbf{x}^j(r)=1}\rho_{rl}\right]1_{\mathcal{N}_l=\text{AND}} + 1_{\mathbf{x}^j(l)=0}\left[\rho_l + (1-\rho_l)\left[1-\prod_{r\in D_l}(1-1_{\mathbf{x}^j(r)=1}\rho_{rl})\right]\right]1_{\mathcal{N}_l=\text{OR}} + 1_{\mathbf{x}^j(l)=1}. \tag{16}$$

Note that $1_{\mathcal{N}_l=\text{AND}}$ is 1 if node l is an AND node, and the transition probabilities in (15) and (16) are obtained according to the conditional probabilities for AND and OR nodes in (1) and (2), respectively. Meanwhile, the subscript k in M_k denotes that the transition matrix in (15) can be time-dependent in general, such as domains with changing exploit probabilities or network structure. Additionally, given that $\mathbf{y}_k$ is the observation vector obtained from nodes $\mathbf{a}_{k-1}$ at time k, we define the *update vector*, $T_k(\mathbf{y}_k, \mathbf{a}_{k-1})$, as:

$$\begin{aligned}(T_k(\mathbf{y}_k,\mathbf{a}_{k-1}))_i &= P\Big(\mathbf{y}_k|\mathbf{x}_k=\mathbf{x}^i,\mathbf{a}_{k-1}\Big)\\ &=\prod_{l=1}^{m} P\Big(\mathbf{y}_k(l)|\mathbf{x}_k=\mathbf{x}^i,\mathbf{a}_{k-1}\Big)\\ &=\prod_{l=1}^{m} P\Big(\mathbf{y}_k(l)|\mathbf{x}_k\big(\mathbf{a}_{k-1}(l)\big)=\mathbf{x}^i\big(\mathbf{a}_{k-1}(l)\big)\Big)\\ &=\prod_{l=1}^{m}\Big|(q-1)\mathbf{x}^i\big(\mathbf{a}_{k-1}(l)\big)-\mathbf{y}_k(l)+1\Big|,\end{aligned} \tag{17}$$

for $i = 1, \ldots, 2^n$, where the last expression in (17) is derived according to the observation process in (3).

The computation of the predictive posterior probability, $\mathbf{\Pi}_{k|k-1}$, can be achieved using the previous posterior probability $\mathbf{\Pi}_{k-1|k-1}$ and the transition matrix M_k through:

$$\mathbf{\Pi}_{k|k-1} = M_k\,\mathbf{\Pi}_{k-1|k-1}. \tag{18}$$

The posterior distribution of states, $\mathbf{\Pi}_{k|k}$, upon observing $\mathbf{y}_k$ at nodes $\mathbf{a}_{k-1}$ can be achieved through the following Bayesian recursion (Kumar and Varaiya 2015; Särkkä 2013):

$$\mathbf{\Pi}_{k|k} = \frac{T_k(\mathbf{y}_k,\mathbf{a}_{k-1})\circ\mathbf{\Pi}_{k|k-1}}{\|T_k(\mathbf{y}_k,\mathbf{a}_{k-1})\circ\mathbf{\Pi}_{k|k-1}\|_1}, \tag{19}$$

where $\circ$ is Hadamard product, and $T_k(\mathbf{y}_k, \mathbf{a}_{k-1})$ is defined in (17).

Using (13) and (14), one can write:

$$A\,\mathbf{\Pi}_{k|k} = \begin{bmatrix}\mathbb{E}[\mathbf{x}_r(1)|\mathbf{a}_{0:k-1},\mathbf{y}_{1:k}]\\ \vdots\\ \mathbb{E}[\mathbf{x}_r(n)|\mathbf{a}_{0:k-1},\mathbf{y}_{1:k}]\end{bmatrix}$$

The optimal MMSE attack detector in (11) for $r = k$ can be computed as:

$$\hat{\mathbf{x}}^{\text{MS}}_{k|k} = \overline{A\mathbf{\Pi}_{k|k}}.$$

with the expected error of the optimal detection according to (12) as:

$$C^{\text{MS}}_{k|k} = \frac{n}{2} - \sum_{i=1}^{n}\left|(A\mathbf{\Pi}_{k|k})_i - \frac{1}{2}\right|. \tag{20}$$

Optimal monitoring policy for BAGs

The proposed attack detection policy in the previous section provides the optimal MMSE solution for detecting network compromises. However, detection accuracy is highly dependent on the available information, i.e., the monitored nodes and the observations. Given the complexity and partial observability of network compromises, accurate detection requires the best use of available monitoring resources. In fact, monitoring should provide the most valuable information about network compromises to enhance the accuracy of detection, especially in sensitive domains where inaccurate attack detection could put the network at risk. It is worth mentioning that the selected monitoring nodes not only provide information about the compromised status at those nodes but also valuable information about all neighboring nodes and the nodes with a feasible path to the currently selected nodes. Therefore, a holistic network-based approach for selecting monitoring nodes is essential. Toward this, the proposed monitoring policy, described below, is derived to optimally support the proposed attack detection policy's performance.

This paper proposes a systematic and optimal solution to enhance the detection accuracy by sequentially monitoring the network components. Let $\{\mathbf{a}_{0:k-1}, \mathbf{y}_k\}$ be the selected monitoring nodes and observations up to time step k. The goal is to select the best m nodes, i.e., $a_k \subset \mathcal{N}$, that maximize the attack detection accuracy in the next step. This can be expressed using the prediction capability of the optimal MMSE attack detection discussed in Theorem 1. Let $\hat{\mathbf{x}}^{\text{MS}}_{k+1|k}$ be the optimal MMSE attack predictor at time step $k+1$ given the observation up to time step k. Then, the optimal subset of nodes yielding the highest attack

prediction error can be formulated through the following optimization problem:

$$\mathbf{a}_k = \underset{\mathbf{a}=\{i_1,..,i_m\}\subset\mathcal{N}}{argmax} \sum_{i\in\mathbf{a}} \mathbb{E}\left[|\mathbf{x}_{k+1}(i) - \hat{\mathbf{x}}^{\mathrm{MS}}_{k+1|k}(i)\|\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}\right], \tag{21}$$

where the expectation is with respect to unobserved state $\mathbf{x}_{k+1}$. The solution to the optimization in (21) guarantees to achieve the minimum expected MSE error (or the highest detection accuracy) in the next time step. Meanwhile, the policy in (21) can also be interpreted as monitoring a subset of nodes most likely to be miss-detected in the next step. This assures optimal use of available resources for monitoring the vulnerable parts of networks given the latest information. The optimal MMSE predictor $\hat{\mathbf{x}}^{\mathrm{MS}}_{k+1|k}$ can be obtained according to Theorem 1 as: $\hat{\mathbf{x}}^{\mathrm{MS}}_{k+1|k} = \overline{\mathbb{E}[\mathbf{x}_{k+1}(i)|\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}]}$. Using Theorem 1, the expression in (21) can also be further simplified as:

$$\begin{aligned}
\mathbf{a}_k &= \underset{\mathbf{a}=\{i_1,..,i_m\}\subset\mathcal{N}}{argmax} \sum_{i\in\mathbf{a}} \mathbb{E}\left[|\mathbf{x}_{k+1}(i) - \hat{\mathbf{x}}^{\mathrm{MS}}_{k+1|k}(i)||\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}\right] \\
&= \underset{\mathbf{a}=\{i_1,..,i_m\}\subset\mathcal{N}}{argmax} \frac{m}{2} - \sum_{i\in\mathbf{a}} \left|\mathbb{E}[\mathbf{x}_{k+1}(i)|\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}] - \frac{1}{2}\right| \\
&= \underset{\mathbf{a}=\{i_1,..,i_m\}\subset\mathcal{N}}{argmin} \sum_{i\in\mathbf{a}} \left|\mathbb{E}[\mathbf{x}_{k+1}(i)|\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}] - \frac{1}{2}\right|.
\end{aligned} \tag{22}$$

The last expression can be interpreted as selecting nodes with the expected predictive value closer to 1/2. The minimum value of $\left|\mathbb{E}[\mathbf{x}_{k+1}(i)|\mathbf{a}_{0:k-1}, \mathbf{y}_{1:k}] - \frac{1}{2}\right|$ is 0, which represents scenarios that the attack detection error is predicted to be the largest at node i in the next time step.

Using the current posterior distribution as $\mathbf{\Pi}_{k|k}$, the exact vector-form computation of the last expression in (22) can be expressed as:

$$\mathbf{a}_k = \underset{\mathbf{a}=\{i_1,..,i_m\}\subset\mathcal{N}}{argmin} \sum_{i\in\mathbf{a}} \left|(A\,\mathbf{\Pi}_{k+1|k})_i - \frac{1}{2}\right|. \tag{23}$$

Regarding the computational complexity of the policy in (23), one should note that the search space in the argument of *argmin* does not demand searching over all m combinations of n nodes. In fact, one can compute the expected predictive error for all nodes as: $s_i = |(A\mathbf{\Pi}_{k+1|k})_i - \frac{1}{2}|$, for $i = 1, .., n$; then, m nodes with the minimum s_i can be selected for monitoring purpose. Meanwhile, the predictive posterior probability $\mathbf{\Pi}_{k+1|k}$ can be simply computed through current posterior probability $\mathbf{\Pi}_{k|k}$ in real-time.

For domains with flexible available resources, the number of nodes for monitoring can be selected adaptively at any given time. In this scenario, the size of $\mathbf{a}_{k-1}$ could be set according to the extent of network vulnerabilities and the targeted detection accuracy. Assuming the objective is to keep the miss-detection rate for all nodes below $100\alpha\%$, where $0 \leq \alpha \leq 0.5$. This can be achieved by monitoring all nodes that their expected predictive errors exceed α as:

$$\mathbf{a}_k = \underset{\mathbf{a}=\{i_1,..,i_m\}\subset\mathcal{N}}{argmax} \sum_{i\in\mathbf{a}} \left(\frac{1}{2} - \left|(A\mathbf{\Pi}_{k+1|k})_i - \frac{1}{2}\right|\right) > \alpha. \tag{24}$$

If the expected predictive errors for all nodes fall below α, the monitoring can be skipped in the next step; however, if expected predictive errors for several nodes are higher than α, up to m of those nodes should be monitored in the next time step. The expected predictive error for each node takes a value between 0 and 1/2; thus, a smaller value of α employs more extensive monitoring to assure accurate detectability of the entire network. Furthermore, if accurate detection is necessary at certain parts of the network, a smaller α can be used for corresponding nodes.

The detailed steps of the proposed optimal MMSE attack detection and monitoring policy for BAGs are provided in Algorithm 1. The algorithm progresses sequentially; a new monitoring set is selected, and the corresponding observations are used for detection in the next step. The algorithm's computational complexity is of order $O(2^{2n})$ due to the transition matrix involved in updating the attack posterior distribution. The size of the transition matrix grows exponentially with the number of components in the network. As a result, it is not possible to compute the attack posterior distribution exactly, preventing the applicability of the proposed monitoring and detection policies in large BAGs. Therefore, our future work will focus on developing scalable particle filtering approaches capable of approximating these optimal monitoring and detection policies. The binary structure of the state variables in BAGs will be exploited to achieve approximate MMSE optimality while remaining computationally efficient.

Algorithm 1 Optimal MMSE Monitoring and Attack Detection for Bayesian Attack Graphs

Inputs: Initial attack distribution, $\mathbf{\Pi}_{1|0}$; fixed monitoring nodes, m; maximum desired detection error, α.

for $k = 1, 2, \ldots$ **do**
  if Fixed Resource Monitoring **then**
    Fixed Resource Monitoring: $\mathbf{a}_{k-1} = \underset{\mathbf{a}=\{i_1,..,i_m\}\subset\mathcal{N}}{\operatorname{argmin}} \sum_{i\in\mathbf{a}} \left| (A\mathbf{\Pi}_{k|k-1})_i - \frac{1}{2} \right|$.
  else
    Adaptive Resource Monitoring: $\mathbf{a}_{k-1} = \underset{\mathbf{a}=\{i_1,..,i_m\}\subset\mathcal{N}}{\operatorname{argmax}} \sum_{i\in\mathbf{a}} \left(\frac{1}{2} - \left| (A\mathbf{\Pi}_{k|k-1})_i - \frac{1}{2} \right| \right) > \alpha$.
  end if
  Observe the network subset $\mathbf{a}_{k-1}$ to get the observation $\mathbf{y}_k$.
  Update: $\mathbf{\Pi}_{k|k} = \frac{T_k(\mathbf{y}_k, \mathbf{a}_{k-1}) \circ \mathbf{\Pi}_{k|k-1}}{\|T_k(\mathbf{y}_k, \mathbf{a}_{k-1}) \circ \mathbf{\Pi}_{k|k-1}\|_1}$.
  Prediction: $\mathbf{\Pi}_{k+1|k} = M_{k+1}\, \mathbf{\Pi}_{k|k}$.
  MMSE Attack Detector: $\hat{\mathbf{x}}_{k|k}^{\mathrm{MS}} = \overline{A\mathbf{\Pi}_{k|k}}$.
  Optimal MSE: $C_{k|k}^{\mathrm{MS}} = \frac{n}{2} - \sum_{i=1}^{n} \left| (A\mathbf{\Pi}_{k|k})_i - \frac{1}{2} \right|$.
end for

Numerical experiments

The numerical experiments in this section evaluate the performance of the proposed attack detection and monitoring policies. The five methods considered for our comparison are: (1) All nodes monitoring, (2) Proposed Adaptive Resource Monitoring; (3) Proposed Fixed Resource Monitoring; (4) Random Monitoring, and (5) Fixed Nodes Monitoring. The first algorithm represents the baseline results, where all nodes are monitored at all time steps. The results obtained by this method specify the lower bound error and higher bound accuracy achievable by other methods with limited monitoring resources. For the third, fourth, and fifth methods, the number of monitoring nodes is m at any given time, whereas, for the second method, the maximum number of monitoring nodes is set to be m. In the fixed node monitoring policy, a fixed set of random nodes are used for monitoring purposes throughout the process. In the random policy, a random subset of m nodes is selected at each time step for monitoring purposes. All the results represented in the numerical experiments are averaged over 100 independent runs obtained for trajectories of length 10. Three important metrics used for performance assessments are average accuracy, error, and total error of attack detection, which can be expressed as:

Average accuracy of attack detection at time k :

$$\frac{1}{100}\sum_{t=1}^{100} 1_{\mathbf{x}_k^t = \hat{\mathbf{x}}_k^t},$$

Average error of attack detection at time k :

$$\frac{1}{100}\sum_{t=1}^{100} \|\mathbf{x}_k^t - \hat{\mathbf{x}}_k^t\|_1$$

Total average error of attack detection :

$$\frac{1}{100}\sum_{t=1}^{10}\sum_{k=1}^{10} \|\mathbf{x}_k^t - \hat{\mathbf{x}}_k^t\|_1,$$

where $\mathbf{x}_k^t$ and $\hat{\mathbf{x}}_k^t$ are the true and detected compromises at time step k in the t-th trajectory respectively.

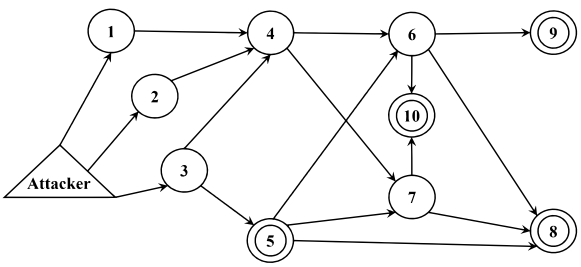

Fig. 2 The 10-node BAG used for the first set of experiments

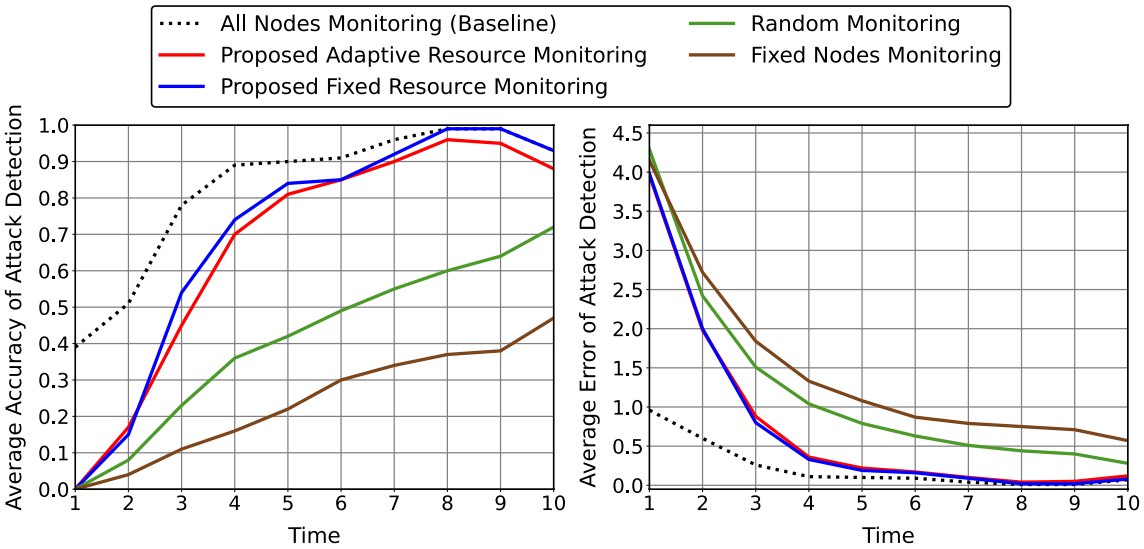

Fig. 3 The average attack detection accuracy and error obtained for 10-node BAG by various policies

Experiment 1—10-Nodes BAG

In this part of the experiments, we consider detecting attacks in the network used in Hu et al. (2020) and shown in Fig. 2. The BAG consists of 10 nodes, resulting in $2^{10} = 1{,}024$ different possible states (i.e., network compromises). A uniform prior is considered for the initial network compromise, i.e., $\mathbf{\Pi}_{1|0}(i) = 1/2^{10}, i = 1, ..., 2^{10}$. The measurement noise is set as $q = 0.2$, and the maximum desired detection error is set as $\alpha = 0.15$. The network vulnerabilities indicated by ρ_{ij} can be represented by:

$\rho_{14} = 0.5700$, $\rho_{24} = 0.5700$, $\rho_{34} = 0.5700$, $\rho_{35} = 0.4329$, $\rho_{46} = 0.8054$, $\rho_{47} = 0.7722$, $\rho_{56} = 0.8054$, $\rho_{57} = 0.7722$, $\rho_{58} = 0.3549$, $\rho_{68} = 0.3549$, $\rho_{69} = 0.3400$, $\rho_{6\,10} = 0.3811$, $\rho_{78} = 0.3549$, $\rho_{7\,10} = 0.3811$.

Three nodes are susceptible to external attacks, represented through the following parameters:

$\rho_1 = 0.6900$, $\rho_2 = 0.6200$, $\rho_3 = 0.5300$. In the first experiment, the number of monitoring nodes m is set as 2. The average detection accuracy and error are shown in Fig. 3. As expected, the highest accuracy rate is obtained by the baseline method, where all nodes are monitored at all time steps. The accuracy of the proposed adaptive resource and the proposed fixed resource monitoring policies are closer to the baseline and empirically converge to the baseline as time progresses. The results of the fixed nodes and random node monitoring policies are significantly lower than the proposed methods, which demonstrates the importance of intelligent node monitoring for enhancing attack detection accuracy. In particular, after 10 time-steps, the average accuracy of attack detection by the proposed policies is above 86%, which is much higher than 72% obtained by the random, and 46% obtained by the fixed nodes monitoring policies. Similar results can be seen in Fig. 3b in terms of the average error of attack detection obtained by various methods.

The average number of monitored nodes under various policies is shown in Fig. 4. For the random monitoring policy, all nodes are almost monitored equally, whereas, under the proposed policies, nodes 8, 10, 9, and 5 have been monitored more often. These imbalanced monitoring of nodes come with better accuracy of detection, represented in Fig. 3. One can see a significantly less number of monitored nodes under the proposed adaptive resource monitoring policy compared to the fixed

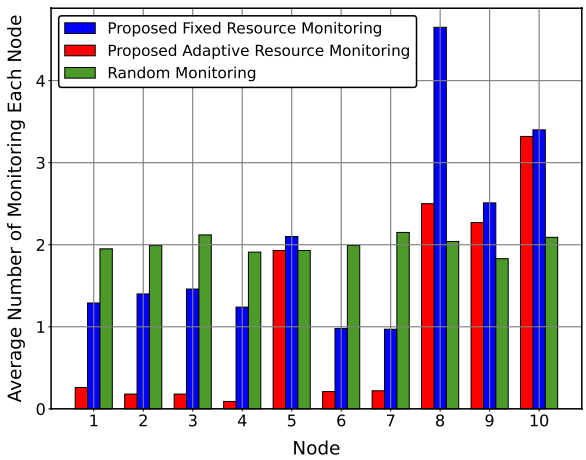

Fig. 4 The average number of monitoring each node obtained for 10-node BAG by various policies

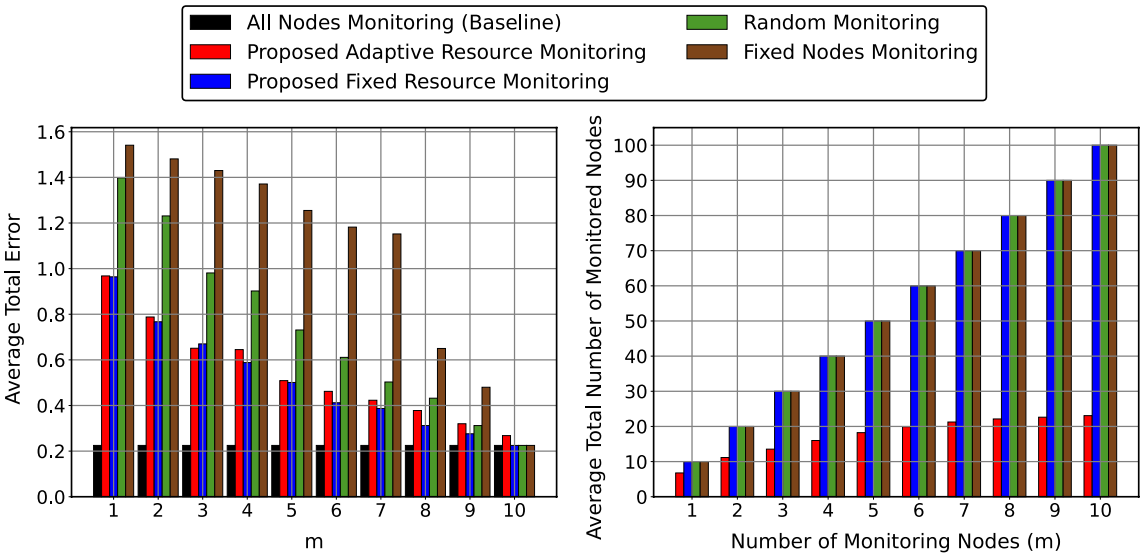

Fig. 5 The average attack detection error and the total monitored nodes with respect to available resource m obtained for 10-node BAG by various policies

resource monitoring policy. In particular, nodes 1, 2, 3, 4, 6, and 7 are selected significantly less under the proposed adaptive resource policy. Despite much lower monitoring under this policy, similar performance is obtained by the proposed adaptive resource monitoring policy compared to the fixed resource monitoring policy (see Fig. 3). The results imply that the proposed monitoring policies can monitor nodes that enhance the detection of the entire network's compromises.

In this part of the experiment, we analyze the impact of the number of monitoring nodes on the performance of the proposed policies. Figure 5a represents the average total error obtained by various methods with respect to the available number of monitoring resources, i.e., m. The minimum average error is obtained by the proposed policies in all conditions. The total error decreases for all methods as more monitoring resources are available; in particular, for $m = 10$, the error of all methods becomes

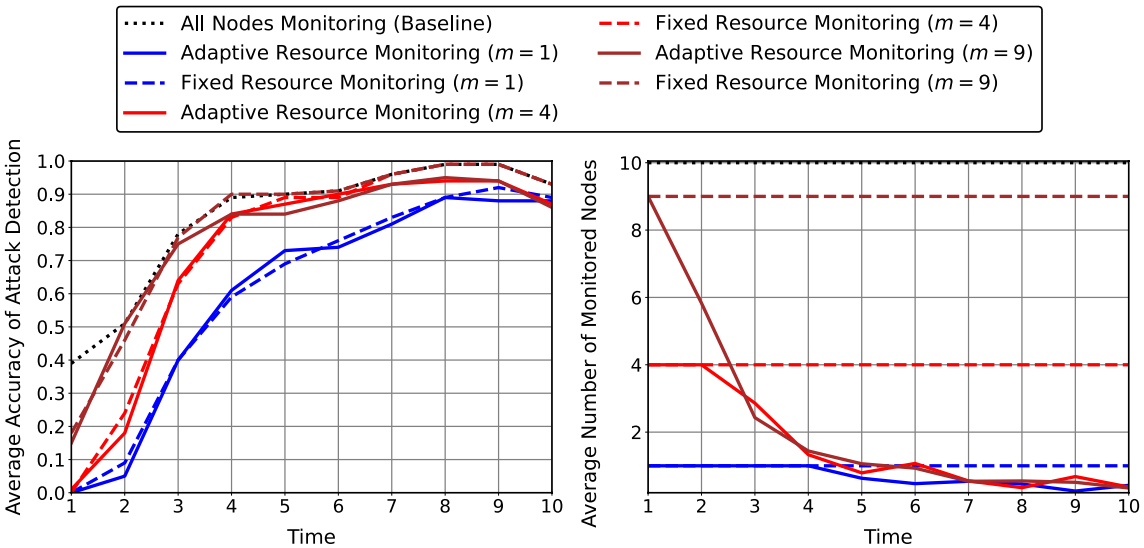

Fig. 6 The average attack detection error and the number of monitoring nodes for $m = 1, 4$ and 9 with respect to the time obtained by the proposed monitoring policies

the same, as all nodes can be monitored given the available resource (except for the adaptive resource monitoring that might use fewer monitoring nodes). Figure 5b represents the used resources (i.e., the total number of monitored nodes) by all policies. As more resources become available, the average number of monitoring nodes increases for all methods. However, the proposed adaptive resource monitoring policy uses significantly fewer monitoring resources while yielding the same average error as the proposed fixed resource monitoring policy and a much lower average error than the other two policies. This comes from the capability of the proposed adaptive resource monitoring policy to properly use available resources if the expected detection error exceeds the desired detection error $\alpha = 0.15$. Therefore, considering average error and used resources, the best results are obtained for the proposed adaptive resource monitoring policy.

To better analyze the proposed method's efficiency in using available resources, we represent the average accuracy and monitored nodes with respect to the time step. Fig 6 contains the results of proposed adaptive resource and fixed resource monitoring policies for $m = 1, 4$, and 9. As shown in Fig 6(a), the average detection accuracy is similar for both policies for any given m and increases as more information becomes available. When larger resources are available (i.e., larger m), the performance of both policies converges to the baseline approach (all nodes monitored). Figure 6b compares the average number of monitored nodes employed by both policies for various m values. The proposed fixed resource policy monitors a fixed number of m nodes at any given time, whereas the number of monitored nodes decreases significantly as more information becomes available. A larger number of nodes monitored in the first step comes from the uniform prior distribution of compromises; however, as time progresses and more information is acquired, the monitored nodes significantly reduce and converge to an average of 1.2 in all conditions. Therefore, comparing the accuracy and the employed resources on the left and right side of Fig. 6, one can see that the adaptive resource policy reduces resource consumption without significantly impacting detection quality.

The impact of the monitoring or measurement noise on the performance of the proposed policies is analyzed in this section. The measurement noise represents the likelihood of miss identifying compromises in the network. A larger measurement noise models a less-advanced monitoring process or the existence of new or difficult-to-detect attacks. Figure 7a represents the average detection error obtained for 100 trajectories of length 10 with respect to measurement noise. As expected, the average error increases for all methods as the level of noise increases. This is due to the inaccuracy of identifying potential attacks during monitoring, which degrades detection accuracy. As expected, the minimum average error is obtained by the baseline policy. For a specific case of $q = 1$, which represents the extreme case of miss-monitoring all compromises in the network, the maximum total error is achieved for all methods. However, for smaller values of measurement noise, the proposed fixed resource and adaptive resource policies yield

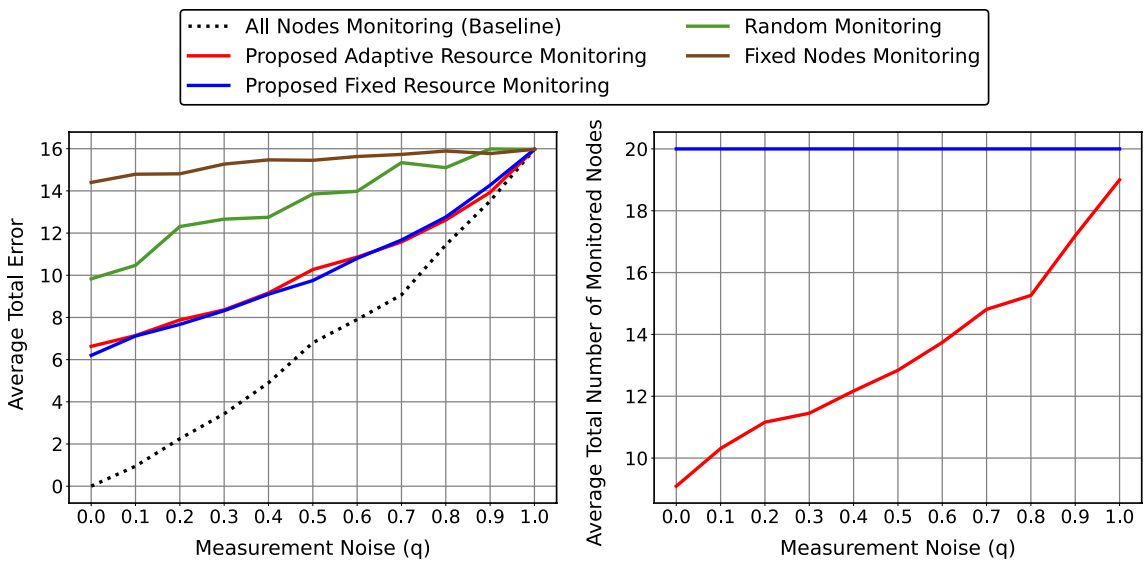

Fig. 7 The average error of attack detection and the number of monitored nodes with respect to the measurement noise q obtained by various policies

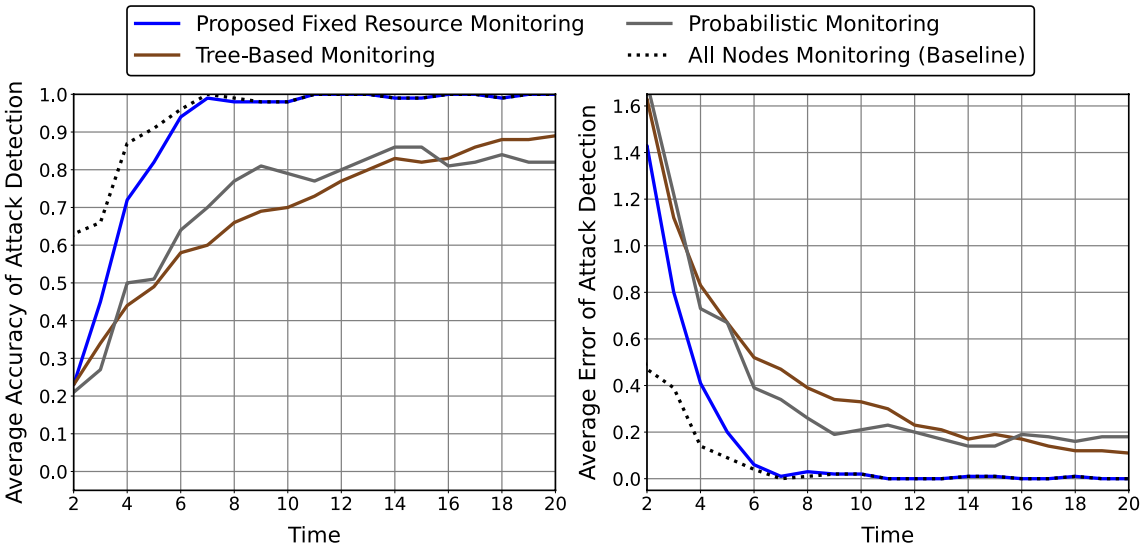

Fig. 8 The average attack detection accuracy and error obtained for 10-node BAG by proposed monitoring policy and two other monitoring policies

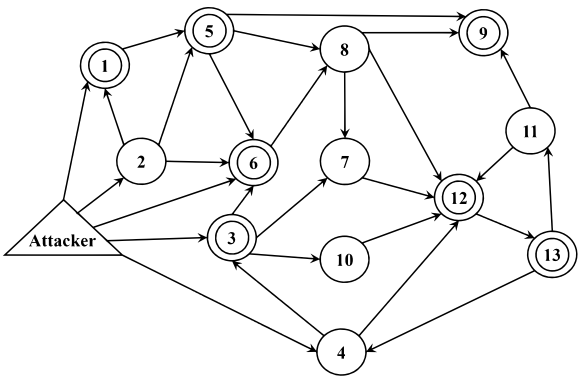

Fig. 9 The 13-node BAG used for the second set of experiments

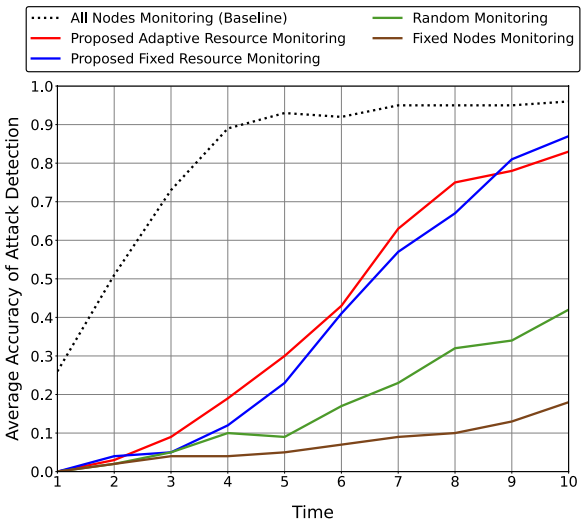

Fig. 10 The average attack detection accuracy with respect to the time step obtained by various policies

significantly smaller average errors than other policies. This again demonstrates the capability of the proposed policies in effectively monitoring nodes under easy-to-detect and difficult-to-detect attacks. Figure 7b demonstrate the average total number of monitored nodes for the proposed methods. Similar to previous results, the average number of monitored nodes is much smaller by the proposed adaptive resource policy. The reduction becomes less visible for larger measurement noise since more monitoring is needed to achieve the desired detection accuracy.

In this part of the experiment, we compare the performance of the proposed monitoring policy with that of the tree-based monitoring approach (Noel and Jajodia 2008) and the probabilistic vulnerability assessment approach (Dantu et al. 2004). The tree-based approach simulates the most probable attack path and selects monitoring nodes with the highest vulnerabilities on the path. The probabilistic vulnerability assessment approach selects nodes with the highest expected increase in the compromise probability, which represents the most vulnerable nodes in the network. Figure 8 shows the performance of attack detection under various monitoring policies for $m = 2$ and $q = 0.2$. The proposed monitoring policy outperforms the other methods and achieves the highest detection accuracy and minimum detection error. Both the tree-based and probabilistic monitoring policies cannot fully detect the system even under larger data. This is due to the fact that these methods aim to select nodes with the highest vulnerability, whereas the proposed monitoring

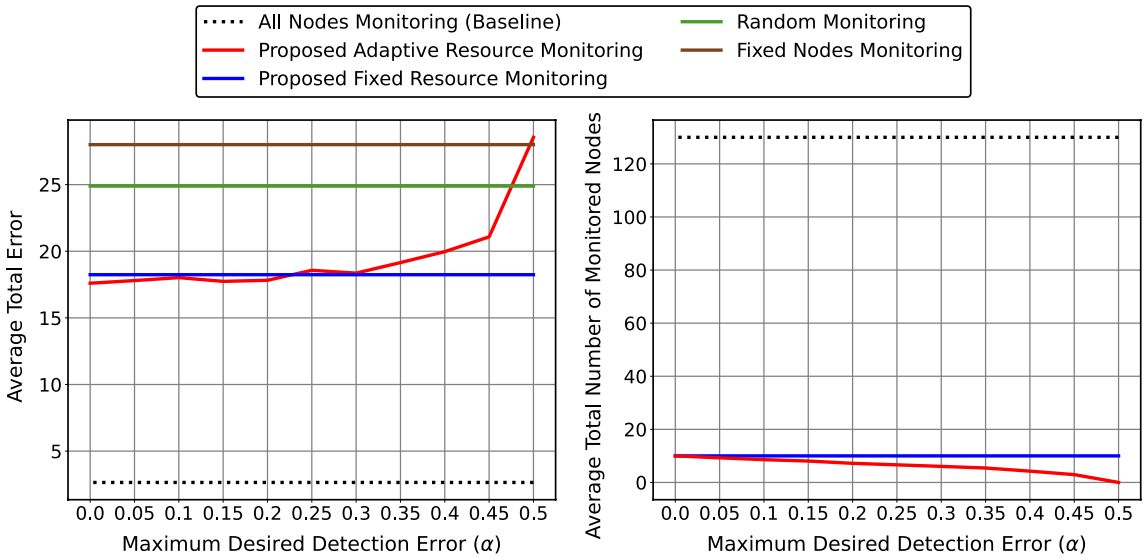

Fig. 11 The average attack detection error and the average number of monitored nodes with respect to the maximum desired detection error α

policy can optimally allocate resources by monitoring nodes that are most likely to be miss-detected in the next time step.

Experiment 2—13-Nodes BAG

For the second part of our experiments, we analyze the attack detection and monitoring for a network depicted in Fig. 9. This network consists of 13 nodes, leading to $2^{13} = 8{,}192$ possible compromise states. A uniform prior state distribution is considered for our experiments with $q = 0.2$, $\alpha = 0.15$, and $m = 1$. The network consists of five external attacks with the following parameters: $\rho_1 = 0.60$, $\rho_2 = 0.50$, $\rho_3 = 0.40$, $\rho_4 = 0.70$, $\rho_6 = 0.30$. The network internal vulnerabilities ρ_{ij} can be represented by:

$\rho_{15} = 0.50$, $\rho_{21} = 0.60$, $\rho_{25} = 0.50$, $\rho_{26} = 0.30$, $\rho_{36} = 0.30$,
$\rho_{37} = 0.20$, $\rho_{3\,10} = 0.10$, $\rho_{43} = 0.40$, $\rho_{4\,12} = 0.40$, $\rho_{56} = 0.30$,
$\rho_{58} = 0.30$, $\rho_{59} = 0.20$, $\rho_{68} = 0.30$, $\rho_{7\,12} = 0.40$, $\rho_{87} = 0.20$
$\rho_{89} = 0.20$, $\rho_{8\,12} = 0.40$, $\rho_{10\,12} = 0.40$, $\rho_{11\,9} = 0.20$, $\rho_{11\,12} = 0.40$,
$\rho_{12\,13} = 0.80$, $\rho_{13\,4} = 0.70$, $\rho_{13\,11} = 0.60$.

The average detection accuracy with respect to the time step obtained by various policies is shown in Fig. 10. The highest accuracy is obtained by the proposed policies, which ultimately converges to the baseline method as more data becomes available. It should be noted that the maximum number of monitoring nodes is set to be $m = 1$ in this network with 13 nodes. Therefore, the convergence of the proposed policies' average accuracy to the baseline policy (with all nodes monitored) represents the capability of the proposed policies in the intelligent node selection. Finally, by comparing the results for fixed nodes and random monitoring policies, one can understand that non-systematic monitoring does not reveal network vulnerabilities and can lead to a huge error in attack detection.

The impact of the maximum desired detection error in the adaptive resource monitoring policy is analyzed here. The parameter α indicates the maximum acceptable detection error for any given node. The proposed policy monitors up to m nodes with the expected predictive error exceeding the α value. The average result for $m = 1$ and α ranging between 0 and 0.5 is presented in Fig 11. As shown in Fig 11a, the average detection error increases as the value of α increases. The reason is that a larger α value represents a more acceptable detection error, which consequently appears in terms of a larger detection error. The results of all monitoring policies (except adaptive resource monitoring) are shown as horizontal lines in Fig. 11a. These policies do not rely on α.

Figure 11b compares the average number of monitored nodes obtained by both policies. The proposed adaptive resource monitoring uses smaller resources than the fixed resource monitoring policy. For very small values of α, the average number of monitored nodes by both policies are similar, but as the value of α increases, the average number of monitored nodes decreases significantly for the proposed adaptive monitoring policy. Finally, as shown in the results obtained over the 10-node BAG, selecting a reasonable value for α according to the sensitivity of the miss-detection (e.g., $\alpha = 0.15$) often leads to a good balance between the accuracy and the use of

available resources. Finally, one could choose α specific for any given nodes in domains where detecting attack at specific nodes has higher priority over other nodes.

Conclusion

In this paper, we developed optimal monitoring and attack detection methods for the general form of Bayesian attack graphs (BAGs). Our approach takes into account sparse and imperfect monitoring techniques, which differ from most existing attack detection techniques. The proposed policies yield the exact minimum mean square error (MMSE) optimality by exploiting the binary structure of nodes in the graph. Optimal sequential monitoring is achieved by selecting a subset of nodes that lead to the highest detectability of network compromises or, equivalently, the least network vulnerability. The exact matrix-form algorithms for the proposed monitoring and detection policies were introduced in this paper. The performance of the proposed methods was demonstrated using comprehensive numerical experiments. Our future work will focus on scaling the proposed attack detection and network monitoring policies to large networks and deriving policies for intelligently defending the network against potential attacks.

Abbreviations

BAG	Bayesian attack graph
MSE	Mean squared error
MMSE	Minimum mean square error
HMM	Hidden Markov model

Acknowledgements

Not applicable.

Author contributions

AK developed the proposed detection and monitoring policies, performed the experiments, and wrote the manuscript. MI proposed the initial idea of the proposed policies and oversaw the research. Both authors have read and approved the final manuscript.

Funding

This work has been supported in part by the National Science Foundation award IIS-2202395, ARMY Research Office award W911NF2110299, and Oracle Cloud credits and related resources provided by the Oracle for Research program.

Availability of data and materials

Not applicable.

Declarations

Competing interests

The authors declare that they have no competing interests.

Received: 18 January 2023 Accepted: 29 March 2023
Published online: 01 September 2023

References

Agmon N, Shabtai A, Puzis R (2019) Deployment optimization of IoT devices through attack graph analysis. In: Proceedings of the 12th conference on security and privacy in wireless and mobile networks, pp 192–202

Al Ghazo AT, Ibrahim M, Ren H, Kumar R (2019) A2G2V: automatic attack graph generation and visualization and its applications to computer and SCADA networks. IEEE Trans Syst Man Cybern: Syst 50(10):3488–3498

Al-Araji Z, Syed Ahmad SS, Abdullah RS et al (2022) Attack prediction to enhance attack path discovery using improved attack graph. Karbala Int J Mod Sci 8(3):313–329

Albanese M, Jajodia S, Noel S (2012) Time-efficient and cost-effective network hardening using attack graphs. In: IEEE/IFIP international conference on dependable systems and networks (DSN 2012). IEEE, pp 1–12

Alhomidi M, Reed M (2013) Risk assessment and analysis through population-based attack graph modelling. In: World Congress on Internet Security (WorldCIS-2013). IEEE, pp 19–24

Bai C-Z, Gupta V, Pasqualetti F (2017) On Kalman filtering with compromised sensors: Attack stealthiness and performance bounds. IEEE Trans Autom Control 62(12):6641–6648

Capobianco F, George R, Huang K, Jaeger T, Krishnamurthy S, Qian Z, Payer M, Yu P (2019) Employing attack graphs for intrusion detection. In: Proceedings of the new security paradigms workshop, pp 16–30

Chadza T, Kyriakopoulos KG, Lambotharan S (2020) Analysis of hidden Markov model learning algorithms for the detection and prediction of multi-stage network attacks. Futur Gener Comput Syst 108:636–649

Chen YY, Xu B, Long J (2021) Information security assessment of wireless sensor networks based on Bayesian attack graphs. J Intell Fuzzy Syst 41(3):4511–4517

Chockalingam S, Pieters W, Teixeira A, Gelder Pv (2017) Bayesian network models in cyber security: a systematic review. In: Nordic conference on secure IT systems. Springer, pp 105–122

Dantu R, Loper K, Kolan P (2004) Risk management using behavior based attack graphs. In: International Conference on Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004., 1:445–449. IEEE

Frigault M, Wang L (2008) Measuring network security using Bayesian network-based attack graphs. In: 2008 32nd Annual IEEE International Computer Software and Applications Conference, pp. 698–703. IEEE

Frigault M, Wang L, Jajodia S, Singhal A (2017) Measuring the overall network security by combining CVSS scores based on attack graphs and Bayesian networks 1–23

Holgado P, Villagrá VA, Vazquez L (2017) Real-time multistep attack prediction based on hidden Markov models. IEEE Trans Dependable Secure Comput 17(1):134–147

Homer J, Zhang S, Ou X, Schmidt D, Du Y, Rajagopalan SR, Singhal A (2013) Aggregating vulnerability metrics in enterprise networks using attack graphs. J Comput Secur 21(4):561–597

Hu Z, Zhu M, Liu P (2020) Adaptive cyber defense against multi-stage attacks using learning-based POMDP. ACM Transactions on Privacy and Security (TOPS) 24(1):1–25

Husák M, Komárková J, Bou-Harb E, Čeleda P (2018) Survey of attack projection, prediction, and forecasting in cyber security. IEEE Commun Surv Tutor 21(1):640–660

Hu Z, Zhu M, Liu P (2017) Online algorithms for adaptive cyber defense on Bayesian attack graphs. In: Proceedings of the 2017 workshop on moving target defense, pp 99–109

Krisper M, Dobaj J, Macher G, Schmittner C (2019) Riskee: a risk-tree based method for assessing risk in cyber security. In: Systems, Software and Services Process Improvement: 26th European Conference, EuroSPI 2019, Edinburgh, UK, September 18–20, 2019, Proceedings 26, pp. 45–56. Springer

Kumar PR, Varaiya P (2015) Stochastic systems: Estimation, identification, and adaptive control. SIAM

Lallie HS, Debattista K, Bal J (2020) A review of attack graph and attack tree visual syntax in cyber security. Comput Sci Rev 35:100219

Li T, Liu Y, Liu Y, Xiao Y, Nguyen NA (2020) Attack plan recognition using hidden Markov and probabilistic inference. Comput Secur 97:101974

Liang C, Wen F, Wang Z (2019) Trust-based distributed Kalman filtering for target tracking under malicious cyber attacks. Information Fusion 46:44–50

Liu S-c, Liu Y (2016) Network security risk assessment method based on HMM and attack graph model. In: 2016 17th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/distributed Computing (SNPD), pp. 517–522. IEEE
Liu J, Liu B, Zhang R, Wang C (2019) Multi-step attack scenarios mining based on neural network and Bayesian network attack graph. In: International conference on artificial intelligence and security. Springer, pp 62–74
Ma Y, Wu Y, Yu D, Ding L, Chen Y (2022) Vulnerability association evaluation of internet of thing devices based on attack graph. Int J Distrib Sens Netw 18(5):15501329221097816
Malzahn D, Birnbaum Z, Wright-Hamor C (2020) Automated vulnerability testing via executable attack graphs. In: 2020 international conference on cyber security and protection of digital services (Cyber Security). IEEE, pp 1–10
Matthews I, Mace J, Soudjani S, van Moorsel A (2020) Cyclic Bayesian attack graphs: a systematic computational approach. In: 2020 IEEE 19th international conference on trust, security and privacy in computing and communications (TrustCom). IEEE, pp 129–136
Miehling E, Rasouli M, Teneketzis D (2015) Optimal defense policies for partially observable spreading processes on Bayesian attack graphs. In: Proceedings of the second ACM workshop on moving target defense, pp 67–76
Munoz Gonzalez L, Lupu E (2016) Bayesian attack graphs for security risk assessment. IST-153 Workshop on Cyber Resilience
Muñoz-González L, Sgandurra D, Barrère M, Lupu EC (2017) Exact inference techniques for the analysis of Bayesian attack graphs. IEEE Trans Dependable Secure Comput 16(2):231–244
Nguyen HH, Palani K, Nicol DM (2017) An approach to incorporating uncertainty in network security analysis. In: Proceedings of the hot topics in science of security: symposium and bootcamp, pp 74–84
Nipkow T et al (2012) Advances in probabilistic model checking. Software Safety and Security: Tools for Analysis and Verification 33(126)
Noel S, Jajodia S (2008) Optimal ids sensor placement and alert prioritization using attack graphs. J Netw Syst Manage 16:259–275
Noel S, Jajodia S (2014) Metrics suite for network attack graph analytics. In: Proceedings of the 9th Annual Cyber and Information Security Research Conference, pp 5–8
Noel S, Jajodia S (2017) A suite of metrics for network attack graph analytics. Network Security Metrics 141–176
Ou X, Boyer WF, McQueen MA (2006) A scalable approach to attack graph generation. In: Proceedings of the 13th ACM Conference on Computer and Communications Security, pp 336–345
Poolsappasit N, Dewri R, Ray I (2011) Dynamic security risk management using Bayesian attack graphs. IEEE Trans Dependable Secure Comput 9(1):61–74
Radack SM et al (2007) The Common Vulnerability Scoring System (CVSS)
Ramaki AA, Khosravi-Farmad M, Bafghi AG (2015) Real time alert correlation and prediction using Bayesian networks. In: 2015 12th International Iranian Society of Cryptology Conference on Information Security and Cryptology (ISCISC), pp 98–103. IEEE
Sahu A, Davis K (2021) Structural learning techniques for Bayesian attack graphs in cyber physical power systems. In: 2021 IEEE Texas power and energy conference (TPEC). IEEE, pp 1–6
Särkkä S (2013) Bayesian filtering and smoothing. Cambridge university press (3)
Sembiring J, Ramadhan M, Gondokaryono YS, Arman AA (2015) Network security risk analysis using improved MulVAL Bayesian attack graphs. Int J Electr Eng Inform 7(4):735
Singhal A, Ou X (2017) Security risk analysis of enterprise networks using probabilistic attack graphs. Network Security Metrics 53–73
Stan O, Bitton R, Ezrets M, Dadon M, Inokuchi M, Yoshinobu O, Tomohiko Y, Elovici Y, Shabtai A (2020) Extending attack graphs to represent cyber-attacks in communication protocols and modern it networks. IEEE Trans Depend Secure Comput
Sun X, Dai J, Liu P, Singhal A, Yen J (2018) Using Bayesian networks for probabilistic identification of zero-day attack paths. IEEE Trans Inf Forensics Secur 13(10):2506–2521
Thanthrige USK, Samarabandu J, Wang X (2016) Intrusion alert prediction using a hidden Markov model. arXiv:1610.07276
Wang X, Cheng M, Eaton J, Hsieh C-J, Wu F (2018) Attack graph convolutional networks by adding fake nodes. arXiv:1810.10751
Wang S, Zhang Z, Kadobayashi Y (2013) Exploring attack graph for cost-benefit security hardening: a probabilistic approach. Comput Secur 32:158–169
Welch G, Bishop G et al (1995) An introduction to the Kalman filter
Yu T, Sekar V, Seshan S, Agarwal Y, Xu C (2015) Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things. In: Proceedings of the 14th ACM workshop on hot topics in networks, pp 1–7

Publisher's Note