A foresight model for intrusion response management

Mohammad GhasemiGol a, Hassan Takabi b, Abbas Ghaemi-Bafghi a,∗

a Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad, Iran
b Department of Computer Science and Engineering, University of North Texas, Denton, TX, USA

ARTICLE INFO

Article history:
Received 22 April 2016
Accepted 27 June 2016
Available online 29 June 2016

Keywords:
Intrusion response system
Response management
Global network dependency graph
Uncertainty-aware attack graph
Multi-level response graph
Foresight cost estimating model
Response data model

ABSTRACT

Intrusion response system (IRS) is one of the most important components in the network security solution that selects appropriate countermeasures to handle the intrusion alerts. Recently, many techniques have been proposed in designing an automated IRS. However, one of the big challenges in intrusion response system which is not considered in the literature is the lack of standardization for intrusion responses. So, this paper investigates how to model and manage the intrusion responses. We present a multilevel response model that provides a high-level view of intrusion responses. We also propose a foresight model to estimate the response cost by considering IDS alerts, network dependencies, attack damage, response impact, and probability of potential attacks. Furthermore, a data model is defined to represent and exchange the intrusion response messages with a standard format.

© 2016 Elsevier Ltd. All rights reserved.

1. Introduction

Intrusion detection and response systems provide a way to protect networks from attacks by external or internal intruders. The intrusion detection system (IDS) generates many alerts as the result of detected attacks. Whereas, the IRS is the last phase of the defense life-cycle that selects appropriate countermeasures to handle malicious behaviors based on received IDS alerts. On the other hand, the number of alerts raised by IDSs is usually extremely high so that the manual responding is not practical.

Since the beginning of the 21st century, the automation of intrusion response systems has attracted increasing attention in the network security research (Carver et al., 2000; Fisch, 1996; Lee et al., 2002; Lewandowski et al., 2001; Mu and Li, 2010; Shameli-Sendi et al., 2014; Stakhanova et al., 2007a, 2012; Toth and Kruegel, 2002). An automated IRS has to assess the cost of responses, the severity of the attack damage, and the other factors for choosing the best responses. Any mistake in this process may lead to blocking the authorized accesses to the network services and reducing the network performance. In this situation, the network administrator prefers to disable the automated IRSs and apply the manual methods (Shameli-Sendi et al., 2014).

Designing an IRS poses several challenges that affects the performance of a network (Shameli-Sendi et al., 2014). The first challenge in IRS development is estimating the response cost that depends on many parameters, although in some literature it only computes according to the expert knowledge (Zhang et al., 2009). Choosing the optimum set of responses against the attacks is another significant difficulty in IRS. Recently, many researchers are interested in cost-sensitive models that compare the intrusion damage and response cost to resist the attacks with minimum cost. However, the lack of standardization in intrusion responses is still one of...
the biggest remaining challenges in IRS that has not been considered before.

Therefore, this paper focuses on the set of intrusion responses as the main input of the IRSs. We also believe that Intrusion Detection Message Exchange Format (IDMEF) (Debar et al., 2007) is not sufficient for intrusion responses and they need a separated standard format. The main contributions of this paper are as follows:

- We propose a response management subsystem to model and manage the intrusion responses.
- We define a multilevel model to categorize intrusion responses. The proposed model provides a high-level view of intrusion responses that helps us in estimating the response cost and selecting appropriate responses against the attacks.
- We present a foresight model to estimate the response cost by considering IDS alerts, network dependencies, attack damage, response impact, and probability of potential attacks.
- We propose a data model to represent and exchange the intrusion response messages with a standard format.

The remainder of this paper is organized as follows. Section 2 reviews the recent existing IRSs. Section 3 explains the proposed models for intrusion response management. Section 4 reports experiments. Finally, Section 5 summarizes the major findings.

2. Related work

There are several taxonomies in this field that provide a comprehensive insight on existing IRSs (Curtis and Carver, 2001; Fisch, 1996; Kanoun et al., 2013; Shameli-Sendi et al., 2014; Stakhanova et al., 2007a; Wang et al., 2006). According to these taxonomies, we can categorize the IRSs based on the following criteria:

- Level of automation: an IRS can be classified as notification, manual, or automated system.
- Activity of triggered response: according to the attempt of minimizing the attack damage, there are active or passive IRSs.
- Cooperation capabilities: the IRS can work in autonomous or cooperative mode to respond to an intrusion.
- Response cost: there are three types of response cost models in the literature: (1) static and static evaluated response cost model (Kanoun et al., 2010; Mu et al., 2010; Papadaki and Furnell, 2006; Stakhanova et al., 2007b, 2012; Strasburg et al., 2009a; Tanachaiwiwat et al., 2002), and (2) dynamic evaluated cost model (Balepin et al., 2003; Shameli-Sendi and Dagenais, 2015; Toth and Kruegel, 2002).
- Response time: the IRSs can be classified into delayed and proactive approaches.
- Adjustment ability: non-adaptive and adaptive are two types of adjustment models for IRSs.
- Response selection: there are three types of response selection model: (1) static model that applies a predefined mapping table between alerts and responses (Bowen et al., 2000; Locasto et al., 2005; Musman and Flesher, 2000; Somayaji and Forrest, 2000; Uppuluri and Sekar, 2001); (2) dynamic model that uses some attack and system factors to apply the appropriate responses (Carver et al., 2000; Lewandowski et al., 2001; Porras and Neumann, 1997; Ragsdale et al., 2000; Schnackenberg et al., 2001; Wang et al., 2001; White et al., 1996); (3) cost-sensitive model that attempts to balance the response cost according to the attack damage (Balepin et al., 2003; Foo et al., 2005; Haslum et al., 2007; Jahnke et al., 2007; Kanoun et al., 2010; Kheir et al., 2010; Lee et al., 2002; Mateos et al., 2012; Mu and Li, 2010; Papadaki and Furnell, 2006; Shameli-Sendi, 2013; Stakhanova et al., 2007b, 2012; Strasburg et al., 2009b; Tanachaiwiwat et al., 2002; Toth and Kruegel, 2002; Zhang et al., 2009).
- Applying location: there are different points in the network to apply the responses such as the start point, firewall, mid-point, and the end point. However, most IRSs apply responses either on the attacked machine or the intruder’s machine.
- Deactivation ability: some IRSs have ability to deactivate the responses after the risk of network is coming down.

During the last decade, some automated techniques have been proposed in IRSs. However, there is no IRS to apply the optimum responses during the attack time. The main reason for this degradation is that many researchers just focus on the response selection methodologies, whereas there is not any standard model for intrusion response management. Therefore, in this paper, we focus on the set of intrusion responses, and we believe that it will be a starting point for standardization of intrusion responses.

3. Intrusion response management

One of the big challenges in IRS is the lack of standardization for intrusion responses that imposes a negative impact to select appropriate responses. Hence, we present a new subsystem for IRS that is responsible for managing and analyzing the intrusion responses before feeding the IRS (see Fig. 1). In this
subsystem, we present a multilevel model to categorize the intrusion responses. We also define a foresight model to estimate the response cost by considering IDS alerts, network dependencies, attack damage, response impact, and probability of potential attacks. Moreover, a data model is proposed by Extensible Markup Language (XML) to represent and exchange the intrusion response messages.

Fig. 2 shows the proposed intrusion response management subsystem for IRS. We describe the components of this model in the following subsections.

3.1. Global network dependency graph (GNDG)

Here, we define a global network dependency graph by considering relationships between all of the network elements in terms of confidentiality, integrity, or availability (CIA) parameters.

Definition 1. Global network dependency graph is a directed graph $GNDG = (H, E, L)$ representing dependencies of services, processes, programs, files, users, and vulnerabilities toward each other.

- $H = \{\eta_1, \eta_2, \ldots, \eta_n\}$ is the set of GNDG nodes including services, processes, programs, files, users, and vulnerabilities.
- $E_{\text{di}} = \{e_{1}, \ldots, e_{\text{all}}\}$ is the set of directed edges that displays the all of dependencies between network elements.
- $L = \{l, \ldots, l_{\text{all}}\}$ is a set of labels where $l \in \{\text{confidentiality, integrity, availability}\}$ indicates the label of edge $e$.

We can analyze the global network dependency graph to generate a dependency matrix.

Definition 2. $DM^{\text{GNDG}}$ is an $m \times m$ matrix which indicates dependencies between nodes in the global network dependency graph where:

- $M \in \{C, I, A\}$ determines the kind of relations in terms of CIA parameters,
- $DM^{\text{GNDG}}(\eta_i, \eta_j) = 0$ if $\eta_i$ and $\eta_j$ are independent from each other,
- $DM^{\text{GNDG}}(\eta_i, \eta_j) = 1$ if $\eta_i$ is dependent on $\eta_j$ regarding CIA parameter $M$.

3.2. Uncertainty-aware attack graph (UAG)

We define the concept of the uncertainty-aware attack graph to handle the uncertainty of attack probabilities (GhasemiGol et al., 2016). The uncertainty arises from measuring probability of vulnerability exploitation. The formal definition of the uncertainty-aware attack graph is given below.

Definition 3. An uncertainty-aware attack graph is a 6-tuple $UAG = (N, E_u, D, Pr, C, G)$, where:

- $N = \{n_1, n_2, \ldots, n_v\}$ is the set of UAG nodes.
- $E_u$ is the set of uncertainty-aware attack graph edges that shows the relationships between vulnerabilities.
- $D$ is a set of pairs $d_{\text{leaf and or}} = \{n, d\}$ where $d \in \{\text{LEAF, AND, OR}\}$ denotes the type of node $n$.
- $Pr = \{P(n_1), P(n_2), \ldots, P(n_v)\}$ is the set of imprecise probabilities, where $P(n_i) = (\hat{P}(n_i), \check{P}(n_i))$, $\hat{P}(n_i) = \sup\{P(n_i) : P \in \rho\}$ indicates the lower probability and $\check{P}(n_i) = \inf\{P(n_i) : P \in \rho\}$ shows the upper probability of each node in the graph and $\rho$ is the set of probability distributions. In the classical case of probability theory, the lower bound is always equal to the upper bound.
- $C$ is a set of constraints on the probability of nodes. Some constraints can be easily extracted from the structure of the current attack graph while the other constraints can be defined by expert’s knowledge. Hence, we have the following constraints for each node $n_i$:
  - If $\langle n, d \rangle \in D$, $d \in \{\text{LEAF}\}$ then $\hat{P}(n_i) = (1, 1)$. 

Fig. 2 – The proposed foresight model for intrusion response management in IRS.
\[ \text{Cost}^{(i)} = \bar{C}^{(i)}(R_i) + \sum_{M \in \{C,A\}} \sum_{j=1}^{m} \alpha^{(i)}(M, \eta_j). \bar{C}^{(m)}(M, R_i, \eta_j) \]
We define Algorithms 2 and 3 to estimate $\tilde{C}_{M}^{i}(M, R, \eta)$ as the total negative impact of responses. In Algorithm 2, the direct impact of each response on CIA parameter of nodes in the global network dependency graph can be computed easily by comparing the feature of MRG nodes and GNDG nodes. The outcome of this algorithm is $R^{MH}$ as an $n \times m$ matrix which indicates the direct impact of responses on each CIA parameter of nodes in the global network dependency graph.

**Algorithm 2 - Computing the direct impact of applying responses on CIA parameters**

**Input:** GNDG, MRG, the number of responses ($n$), the number of nodes in GNDG ($m$)

**Output:** $R^{MH}$ (The direct impact of applying responses on each CIA parameter of nodes in GNDG)

1. Extract IP, servID, vulID, fileID, userID from each $\eta_j$ and $R_i$
2. for each CIA parameter $M = \{C, I, A\}$
3. for each $R_i$
4. for each $\eta_j$
5. $R^{MH}(i, j) = 0$
6. if $R_i$ is a host-level response then
7. if $(R_i.IP == \eta_j.IP)$ then
8. $R^{MH}(i, j) = 1$
9. end if
10. end if
11. if $R_i$ is a notification/attacker/service/vulnerability/file/user-level response then
12. if $(R_i.IP == \eta_j.IP) \& (R_i.ServID|vulID|fileID|userID) == \eta_j.|ServID|vulID|fileID|userID)$ then
13. $R^{MH}(i, j) = 1$
14. end if
15. end if
16. if $R_i$ is an unclassified-level response then
17. $R^{MH}(i, j) = \varepsilon$
18. end if
19. end for
20. end for
In addition, we must consider the indirect impact which comes from relationships between nodes in the global network dependency graph to estimate the total impact of responses. The procedure of computing the total negative impact of responses is shown in Algorithm 3. At the first step of this algorithm, we use Table 1 to compute \( C_{im}(M, R_i) \) as the predefined impact of each response \( R_i \) on CIA parameter \( M \) based on the response category. We also define a new matrix operation shown by \( \otimes \) to estimate \( R\text{Impact}^{(M)}(R, H) \) as the negative impact of responses by using \( RH^{(M)} \) and \( DM^{(M)} \) matrices. At the last step, the total negative impact of each response \( R_i \) can be computed by multiplying \( C_{im}(M, R_i) \) and \( R\text{Impact}^{(M)}(R_i, \eta_j) \).

**Algorithm 3 – Computing the total negative impact of applying responses**

**Input:** \( DM^{(M)}, RH^{(M)}, \) the number of responses \( n \), the number of nodes in \( GNDG \) \( m \).

**Output:** \( \bar{C}_{im}^{(i)}(M, R_i, \eta_j) \) (The total negative impact of applying responses on each CIA parameter of nodes in the global network dependency graph)

1: Define the class of each response and compute its impact on CIA parameters \( \bar{C}_{im}^{(i)}(M, R_i) \) (see Table 1)

2: Compute the total negative impact of each response on CIA parameter of nodes in the global network dependency graph where:

   a. \( \otimes : RH^{(M)}(\text{norm}) \times DM^{(M)}(\text{norm}) \rightarrow R\text{Impact}^{(M)}(\text{norm}) \)

   b. \( R\text{Impact}^{(M)}(R, H) = RH^{(M)}(\text{norm}) \otimes DM^{(M)}(\text{norm}) \)

   c. \( R\text{Impact}^{(M)}(R_i, H) = R\text{Impact}^{(M)}(R_i, H) + \sum_q \text{temp}^{(i)}(1, q) \sum_q \text{temp}^{(i)}(2, q), ..., \sum_q \text{temp}^{(i)}(m, q) \)

   where:
   - \( \text{temp}^{(i)}(r, q) = RH^{(M)}(i, q).DMF^{(M)}(r, q) \), \( i = 1, ..., n \), \( q = 1, ..., m \)
   - \( DMF^{(M)} = DM^{(M)} + I \)

3: Compute the total negative impact of applying responses

   a. \( \bar{C}_{im}^{(i)}(M, R_i, \eta_j) = \min \{ \bar{C}_{im}^{(i)}(M, R_i), R\text{Impact}^{(M)}(R_i, \eta_j) \} \)

   b. \( \bar{C}_{im}^{(i)}(M, R_i, \eta_j) = \max \{ \bar{C}_{im}^{(i)}(M, R_i), R\text{Impact}^{(M)}(R_i, \eta_j) \} \)

   c. \( \bar{C}_{im}^{(i)}(M, R_i, \eta_j) = \bar{C}_{im}^{(i)}(M, R_i, \eta_j), \bar{C}_{im}^{(i)}(M, R_i, \eta_j) > \)

**Table 1 – The cost of each response category on the CIA parameters.**

<table>
<thead>
<tr>
<th>Category</th>
<th>Confidentiality</th>
<th>Integrity</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notification-level</td>
<td>Very high</td>
<td>Very high</td>
<td>Very low</td>
</tr>
<tr>
<td>Attacker-level</td>
<td>Low–high</td>
<td>Low–high</td>
<td>Low–high</td>
</tr>
<tr>
<td>Vulnerability-level</td>
<td>Low–very low</td>
<td>Low–very low</td>
<td>Low–very low</td>
</tr>
<tr>
<td>File-level</td>
<td>Low–very low</td>
<td>Low–very low</td>
<td>High–very high</td>
</tr>
<tr>
<td>User-level</td>
<td>Low–very low</td>
<td>Low–very low</td>
<td>High–very high</td>
</tr>
<tr>
<td>Service-level</td>
<td>Low–very low</td>
<td>Low–very low</td>
<td>High–very high</td>
</tr>
<tr>
<td>Host-level</td>
<td>Very low</td>
<td>Very low</td>
<td>Very high</td>
</tr>
<tr>
<td>Unclassified-level</td>
<td>Low–very high</td>
<td>Low–very high</td>
<td>Low–very high</td>
</tr>
</tbody>
</table>

On the other hand, the positive part of response \( R_i \) can be defined as follows:

\[
\text{Cost}^{(i)} = \sum_i \hat{D}^{(i)}(M, \text{Att}_{u_i}, \eta_j).G^{(i)}(M, R_i, \text{Att}_{u_i})
\]

(2)

where \( \hat{D}^{(i)}(M, \text{Att}_{u_i}, \eta_j) \) shows the total damage of attacks on each CIA parameter in node \( \eta_j \) and \( G^{(i)}(M, R_i, \text{Att}_{u_i}) \) is the total goodness of response \( R_i \) against attack \( \text{Att}_{u_i} \) regarding the CIA parameter \( M \). We can use Algorithms 4 and 5 to estimate the total damage of attacks. The direct damage of attacks can be estimated by using Algorithm 4 with comparing the feature of attacks and \( GNDG \) nodes. The outcome of this algorithm is \( \text{AttH}^{(M)} \) as an \( l \times m \) matrix which indicates the direct damage of attacks on each CIA parameter of nodes in the global network dependency graph.
We can also estimate the total damage of attacks by using Algorithm 5 with considering relationships between nodes in the global network dependency graph. In this algorithm, we apply the proposed matrix operation $\otimes$ to estimate $\text{AlImpact}^{(M)}(\text{Att}, H)$ as the amount of damage on each CIA parameter of nodes in the global network dependency graph by using $\text{Att}^{(M)}$ and $\text{DM}^{(M)}$ matrices. The total damage of attack $\text{Att}_k$ can be computed by multiplying $\tilde{\omega}(M, \eta_j)$ and $\text{AlImpact}^{(M)}(\text{Att}_k, \eta_j)$.

**Algorithm 4. Computing the direct attack damage on CIA parameters**

**Input:** $\text{GNDG}$, the number of network attacks ($I$), the number of nodes in $\text{GNDG}$ ($m$).

**Output:** $\text{Att}^{(M)}$ (The direct attack damage on each CIA parameter of nodes in the global network dependency graph)

1. Extract IP, servID, vulID, fileID, userID from $\eta_j$, $\text{Att}_k$
2. for each CIA parameter $M = \{C, I, A\}$
3.    for each $\text{Att}_k$
4.      for each $\eta_j$
5.       $\text{Att}^{(M)}(k, j) = 0$
6.       if $(\text{Att}_k, \text{IP} = \eta_j, \text{IP}) \& (\text{Att}_k, \{\text{servID, vulID, fileID, userID}\} = \eta_j, \{\text{servID, vulID, fileID, userID}\})$ then
7.           $\text{Att}^{(M)}(k, j) = 1$
8.       end if
9.       if $(\text{Att}_k, \text{IP} = \eta_j, \text{IP})$ then
10.      $\text{Att}^{(M)}(k, j) = \varepsilon$
11.     end if
12.   end for
13. end for
14. end for

**Algorithm 5 – Computing the total damage of attacks**

**Input:** $\tilde{\omega}(M, \eta_j)$, $\text{DM}^{(M)}$, $\text{Att}^{(M)}$, the number of network attacks ($I$), the number of nodes in $\text{GNDG}$ ($m$)

**Output:** $\tilde{D}^{(M)}(M, \text{Att}_k, \eta_j)$ (The total damage of attacks on each CIA parameter of nodes in the global network dependency graph)

1. Compute the total damage of attacks on each CIA parameter of nodes in $\text{GNDG}$ ($\text{AlImpact}^{(M)}(\text{Att}_k, \eta_j)$) where:
   a. $\otimes: \text{Att}^{(M)}(\text{IP}, \text{servID, vulID, fileID, userID}) \otimes \text{DM}^{(M)}(\text{IP}, \text{servID, vulID, fileID, userID}) \rightarrow \text{AlImpact}^{(M)}(\text{IP}, \text{servID, vulID, fileID, userID})$
   b. $\text{AlImpact}^{(M)}(\text{Att}_k, H) = \text{Att}^{(M)}(\text{IP}, \text{servID, vulID, fileID, userID}) \otimes \text{DM}^{(M)}(\text{IP}, \text{servID, vulID, fileID, userID})$
   c. $\text{AlImpact}^{(M)}(\text{Att}_k, H) = \text{AlImpact}^{(M)}(\text{Att}_k, H) + \sum_q \text{temp}^{(k, q)}(1, q) \sum_q \text{temp}^{(k, q)}(2, q) \ldots \sum_q \text{temp}^{(k, q)}(m, q)$ where:
      - $\text{temp}^{(k, q)}(r, q) = \text{Att}^{(M)}(k, q) \cdot \text{DM}^{(M)}(q, r)$, $k = 1, \ldots, I$, $q = 1, \ldots, m$, $r = 1, \ldots, m$
      - $\text{DM}^{(M)} = \text{DM}^{(M)} + I$
2. Compute the total damage of attacks
   a. $\tilde{D}^{(M)}(M, \text{Att}_k, \eta_j) = \min \{\tilde{\omega}(M, \eta_j), \text{AlImpact}^{(M)}(\text{Att}_k, \eta_j)\}$
   b. $\tilde{D}^{(M)}(M, \text{Att}_k, \eta_j) = \max \{\tilde{\omega}(M, \eta_j), \text{AlImpact}^{(M)}(\text{Att}_k, \eta_j)\}$
   c. $\tilde{D}^{(M)}(M, \text{Att}_k, \eta_j) = (<\tilde{D}^{(M)}(M, \text{Att}_k, \eta_j), \tilde{D}^{(M)}(M, \text{Att}_k, \eta_j)>)$
We also define the total goodness of responses with the following equation:

\[
G^{(i)}(M, R, Attack) = \sum_{Path_{Attack}} Gp^{(i)}(M, R, Path_{Attack}) \cdot |Path_{Attack}|
\] (3)

where Path_{Attack} refers to the attack paths which can be extracted from the uncertainty-aware attack graph. We can also find the goodness of response R on the attack path Path_{Attack} by Eq. (4):

\[
Gp^{(i)}(M, R, Path_{Attack}) = \sum_{n_{Attack} \in Path_{Attack}} \left( \omega^{(i)}(M, n_{Attack}) \cdot C^{(i)}_{n_{Attack}}(M, R) \cdot Gn(R, n_{Attack}) \right)
\] (4)

where \( \omega^{(i)}(M, n_{Attack}) \) shows the weight of CIA parameter in the uncertainty-aware attack graph node n_{Attack}, and Gn(R, n_{Attack}) indicates the goodness of response R on the UAG nodes. We can use Eq. (5) to estimate \( \omega^{(i)}(M, n_{Attack}) \) as follows:

\[
\omega^{(i)}(M, n_{Attack}) = \omega^{(i)}(M, \eta_{i}, HN(\eta_{i}, n_{Attack}))
\] (5)

where \( \omega^{(i)}(M, \eta_{i}) \) indicates the weight of each CIA parameter in all GNDG nodes, and HN(\eta_{i}, n_{Attack}) shows the total relationships between GNDG nodes and UAG nodes. We define Algorithms 6 and 7 to compute HN(\eta_{i}, n_{Attack}). Algorithm 6 estimates the direct relationships between GNDG nodes and UAG nodes and returns the HAg matrix.

Algorithm 6. Computing the direct relationships between GNDG nodes and UAG nodes

**Input:** UAG, GNDG, the number nodes in GNDG (\( m \)), the number of nodes in UAG (\( v \))

**Output:** HAg (The direct relationships between GNDG nodes and UAG nodes)

1: Extract IP, servID, vulID, fileID, userID from \( \eta_{j} \) and \( n_{i} \)
2: \textbf{for each} \( \eta_{j} \)
3: \textbf{for each} \( n_{i} \)
4: \hspace{1em} HAg(j, i) = 0
5: \textbf{if} \( \{\eta_{j}, IP = \eta_{j}, IP\} \& \{n_{i}, \{\text{ServID, vulID, fileID, userID}\} = = \{\text{ServID, vulID, fileID, userID}\} \) \textbf{then}
6: \hspace{1em} HAg(j, i) = 1
7: \textbf{end if}
8: \textbf{end for}
9: \textbf{end for}

Also, Algorithm 7 uses HAg and DMAg matrices to estimate HN(\eta_{j}, n_{Attack}). As mentioned before, DMAg is a dependency matrix that indicates dependencies between nodes in the uncertainty-aware attack graph.

Algorithm 7. Computing the total relationships between GNDG nodes and UAG nodes

**Input:** HAg, DMAg, the number nodes in GNDG (\( m \)), the number of nodes in UAG (\( v \))

**Output:** HN(\eta_{j}, n_{Attack}) (The relationships between GNDG nodes and UAG nodes)

1: Compute the total relationships between GNDG nodes and UAG nodes where:
   a. \( \otimes: HAg_{(mov)} \otimes DMAg_{(ovv)} \rightarrow HN_{(mov)} \)
   b. \( HN(H, n_{Attack}) = HAg_{(mov)} \otimes DMAg_{(ovv)} \)
   c. \( HN(\eta_{j}, n_{i}) = HN(\eta_{j}, n_{i}) + \left[ \sum_{q} \text{temp}^{(j)}(1, q) \cdot \sum_{ q} \text{temp}^{(j)}(2, q) \cdots \sum_{ q} \text{temp}^{(j)}(v, q) \right] \) where:
      - \( \text{temp}^{(j)}(r, q) = HAg(j, q) \cdot DMAg^{(M)}(r, q), j = 1, \ldots m, q = 1, \ldots v, \ r = 1, \ldots v \)
      - \( DMAg^{(M)} = DMAg^{(M)} + I \)
We define Algorithms 8 and 9 to estimate $G_n(R, n_{\text{Att}})$ as the total goodness of responses on the UAG nodes. Algorithm 8 computes $RAg$ as the direct goodness of applying responses on the UAG nodes.

**Algorithm 8. Computing the direct goodness of applying responses on UAG nodes**

**Input:** $MRG$, $UAG$, the number of responses $(n)$, the number of nodes in $UAG$ $(v)$

**Output:** $RAg$ (The direct impact of applying responses on the UAG nodes)

1. Extract IP, servID, vulID, fileID, userID from each $n_j$ and $R_i$
2. for each $R_i$
   3. for each $n_j$
   4. $RAg(i, j) = 0$
   5. if ($R_i$ is a host-level response) then
      6. if ($R_i, IP == n_j, IP$) then
         7. $RAg(i, j) = 1$
      8. end if
   9. end if
10. if ($R_i$ is a notification/attacker/service/vulnerability/file/user-level response) then
    11. if ($R_i, IP == n_j, IP$) & ($R_i, \{\text{servID, vulID, fileID, userID}\} == n_j, \{\text{servID, vulID, fileID, userID}\}$) then
       12. $RAg(i, j) = 1$
    13. end if
    14. end if
15. if ($R_i$ is an unclassified-level response) then
    16. $RAg(i, j) = \varepsilon$
    17. end if
18. end for
19. end for

In Algorithm 9, we apply the proposed matrix operation $\otimes$ to estimate the total goodness of responses on the UAG nodes by using $RAg$ and $DMAg$ matrices.

**Algorithm 9 – Computing the total goodness of responses on the UAG nodes**

**Input:** $RAg$, $DMAg$, the number of responses $(n)$, the number of nodes in $UAG$ $(v)$

**Output:** $Gn(R, n_{\text{Att}})$ (The total goodness of responses on the UAG nodes)

2. Compute the total goodness of responses on the UAG nodes ($Gn(R, n_{\text{Att}})$) where:
   a. $\otimes : RAg_{(\text{nov})} \otimes DMAg_{(\text{nov})} \rightarrow Gn_{(\text{nov})}$
   b. $Gn(R, n_{\text{Att}}) = RAg_{(\text{nov})} \otimes DMAg_{(\text{nov})}$
   c. $Gn(R, n_{\text{Att}}) = Gn(R, n_{\text{Att}}) + \left[ \sum_q \text{temp}^{(i)}(1, q) \sum_q \text{temp}^{(i)}(2, q) \ldots \sum_q \text{temp}^{(i)}(\nu, q) \right]$ where:
      - $\text{temp}^{(i)}(r, q) = RAg(i, q) \cdot DMAg^{(M)}(r, q)$, $i = 1, \ldots, n$, $q = 1, \ldots, \nu$, $r = 1, \ldots, \nu$
      - $DMAg^{(M)} = DMAg^{(M)} + I$
Now, we can estimate the range of response cost by combining $\text{Cost}_{t}^{\text{lo}}$ and $\text{Cost}_{t}^{\text{hi}}$. In the optimistic situation, we define the response cost in a specific time as follows:

$$\min \left\{ 1 - \tilde{P}_{1}^{(t)}(\text{Att}_{t}) / \text{Cost}_{t}^{\text{lo}}(\text{Att}_{t}), \text{Cost}_{t}^{\text{hi}}(\text{Att}_{t}) \right\}$$

where $\tilde{P}_{1}^{(t)}(\text{Att}_{t})$ indicates the lower bound and upper bound of probability of potential attacks. We can analyze the uncertainty-aware attack graph to predict future attacks and find some useful information about the attack paths and the probability of vulnerability exploitation. Algorithm 10 shows the procedure of computing the probability of potential attacks with the help of uncertainty-aware attack graph. We can also update the attack probabilities according to the issued IDS alerts. For this reason, we define a similarity function $H_{\text{similarity}}(h_{a}, n_{i})$ that returns the similarity of uncertainty-aware attack graph node $n_{i}$ with hyper-alerts $h_{a}$. The algorithm of this function is explained in the appendix. Hyper-alerts can be generated by analyzing the IDS alerts with the aid of E-correlator as an alert correlation system (Ghasemi Gol and Ghaemi-Bafghi, 2015).

Algorithm 10 – Computing the probability of potential attacks

**Input:** $UAG$, Hyper-alerts

**Output:** $\tilde{P}_{i}^{(t)}(\text{Att}_{t})$ (The probability of potential attacks)

1. Extract the set of constraints $\{C\}$ from uncertainty-aware attack graph or define them by an expert
   a. If $n_{i} \in N$ is a LEAF node in uncertainty-aware attack graph $\tilde{P}(n_{i}) = <1, 1>$.
   b. If $n_{i} \in N$ is an AND node in uncertainty-aware attack graph $\tilde{P}(n_{i}) = \prod_{i \in \text{Predecessor}(n_{i})} \tilde{P}(n_{i})$.
   c. If $n_{i} \in N$ is an OR node in uncertainty-aware attack graph $\tilde{P}(n_{i}) = 1 - \prod_{i \in \text{Predecessor}(n_{i})} (1 - \tilde{P}(n_{i}))$.

2. Compute the probability of nodes in $UAG$
   a. $P_{i}^{(t)}(n_{i}) = \arg\min_{C} \sum_{n_{i} \in N} \tilde{P}(n_{i})$.
   b. $\tilde{P}_{i}^{(t)}(n_{i}) = \arg\max_{C} \sum_{n_{i} \in N} \tilde{P}(n_{i})$.
   c. $\tilde{P}_{i}^{(t)}(n_{i}) = < P_{i}^{(t)}(n_{i}), \tilde{P}_{i}^{(t)}(n_{i}) >$.

3. Update the probability of nodes in the uncertainty-aware attack graph according to hyper-alerts
   a. for each attack graph node $(n_{i})$ do
   b. for each Hyper-alert $(h_{a})$ do
   c. $\theta = H_{\text{similarity}}(h_{a}, n_{i})$.
   d. $P_{i}^{(t)}(n_{i}) = (1 - P_{i}^{(t)}(n_{i})) \times \theta + P_{i}^{(t)}(n_{i})$.
   e. $\tilde{P}_{i}^{(t)}(n_{i}) = (1 - \tilde{P}_{i}^{(t)}(n_{i})) \times \theta + \tilde{P}_{i}^{(t)}(n_{i})$.
   f. $\tilde{P}_{i}^{(t)}(n_{i}) = < P_{i}^{(t)}(n_{i}), \tilde{P}_{i}^{(t)}(n_{i}) >$.
   g. end for
   h. end for

4. Find the probability of potential attacks
   a. If $(n_{i})$ is the goal of $\text{Att}_{t}$ then
   b. $\tilde{P}_{i}^{(t)}(\text{Att}_{t}) = \tilde{P}_{i}^{(t)}(n_{i})$
   c. End if.
Similarly, in the pessimistic situation we can use the following equation:

$$\max \left\{ \left( 1 - \overline{P}(\text{Att}_k) \right) \left( \text{Cost}(t) \right) / \overline{P}(\text{Att}_k) \left( \text{Cost}(t) \right) \right\}$$ (7)

Obviously, using some information about future attacks can result in better estimation for response cost. For example, we know that \(R_i\) is a costly response in present time; however its cost might be acceptable considering the probability of future attacks. Therefore, the given information about future attacks can be useful in estimating the response cost. The overall process of proposed response cost estimating is depicted in Fig. 4.

### 3.4.1. Complexity analysis

In this section, we analyze the complexity of the proposed foresight model for response cost estimating. Suppose that \(n\) is the number of responses, \(m\) the number of all services, processes, programs, files, users, and vulnerabilities in the global network dependency graph, \(m_c\) is the number of constraints on GNDG nodes, \(m\) is the number of network attacks, \(v\) the number of UAG nodes, and \(v_c\) is the number of constraints on UAG nodes. As mentioned in Algorithm 1, we need to solve two linear programming problems to compute the weight of CIA parameters for all services, processes, programs, files, users, or vulnerabilities in global network dependency graph. The linear programming problem with \(d\) variables and \(m\) constraints can be solved in \(O(m)\) time when \(d\) is fixed (Megiddo, 1984). Therefore, the total complexity of computing the weight of CIA parameters for all nodes in global network dependency graph is \(O(m_c)\). According to Algorithms 2 and 3, the time complexity for computing the total negative impact of applying responses on CIA parameters is \(O(nm) + O(m^2) + O(nm^2)\). Similarly, Algorithms 4 and 5 have the time complexity of \(O(m) + O(m^2) + O(m^3)\) for estimating the amount of attack damage. We can also apply Algorithms 6, 7, 8, and 9 to estimate the response goodness with total complexity of \(O(nu) + O(\nu^2) + O(nu^2)\). Finally, in Algorithm 10, we need to solve two linear programming problems again to compute the probability of potential attacks with a time complexity of \(O(\nu_c)\).

### 3.5. Data model for intrusion responses representation

In this section, we define a data model in the XML to represent the intrusion response messages with a standard data format. A common data exchange format can help network administrators to manage the intrusion responses in either manual or automated IRSs. XML is a simplified version of the Standard Generalized Markup Language (SGML) that is gaining widespread attention as a language for representing and exchanging documents and data on the Internet (Bray et al., 1998; De Campos et al., 2010). The individual components of the response data model are explained with Unified Modeling Language (UML) diagrams. Similar to IDMEF, the top-level class for all responses is Intrusion Response Message (IR-Message). There are several subclasses for the IR-Message to provide the...
detailed information carried in the message. The relationship between the principal components of the presented data model is shown in Fig. 5. In the following, we briefly describe the main classes.

- The IR-Message class is composed of three attributes and four aggregated classes. The Response-ID shows the response identification number. The Response-status indicates the response condition in terms of being active or inactive. The Response-cost contains the negative impact of response on the network assets.
- The Response-Target class indicates the address of response goal. It may contain additional information from user, service, process, file, or vulnerability classes.
- The Response-Type class shows the kind of responses in terms of impact level.
- The Response-Location class indicates the place of applying intrusion responses (e.g. Firewall, Client, Server, etc.).
- The Response-Action class shows the kind of actions can be applied by responses (such as shutdown/run, reset, block/unblock, disable/enable, deny/access, notification, alarm, etc.).

4. Experiments

In this section, we present an example to show applicability of the proposed response models in IRSs. Suppose that we have a small network as shown in Fig. 6. There is a firewall to protect...
the network from the internet access. External users are allowed to access the web server through the HTTP protocol and port. The web server has a vulnerability with CVE ID CVE-2006-3747. This vulnerability allows remote attackers to cause a denial of service attack and possibly execute arbitrary code via crafted URLs that are not properly handled using certain rewrite rules. In addition, there is a database server and a workstation user in the internal subnet. The database server can be only accessed by the web server, and it has a remote vulnerability in the MySQL DB service with CVE ID CVE-2009-2446. The workstation machine runs Internet Explorer (IE) in a Windows operating system. IE has vulnerability CVE-2009-1918 that would enable execution of arbitrary code on the victim’s machine. This vulnerability can be exploited while a user visits a maliciously crafted web page. Secretary is an authorized user that is located on workstation machine.

According to the abovementioned topology, we can define the following global network dependency graph (see Fig. 7). In this graph, each node indicates a process, service, program, file, user, or vulnerability; and edges show relationships between nodes by considering CIA parameters.

The weight of each node in the global network dependency graph can be calculated by using Algorithm 1. Table 2 shows the obtained weights for the mentioned example.

In Table 3, we define some of the candidate responses that can be applied in this network. We can also generate the response graph according to the mentioned responses. As shown in Fig. 8, there are some relationships between responses in different levels of impact that help us in estimating the response cost and selecting the appropriate responses in IRSs.

The proposed foresight model can be used to estimate the response cost by considering future situations. In this study, we apply MulVAL network security analyzer (Ou et al., 2005) to generate uncertainty-aware attack graph and predict potential attacks; however we modify it to handle the uncertainty of attack probabilities. The obtained uncertainty-aware attack graph for this example is shown in Fig. 9 (node details are explained in Table 4 of appendix).

In the proposed uncertainty-aware attack graph, we attach two numbers to each node which indicate the lower and upper probabilities of node exploitation. Table 5 contains the lower and upper probabilities of potential attacks and the number of attack paths which can be extracted by analyzing the uncertainty-aware attack graph. Furthermore, the detail of attack paths is shown in Table 6.

Now, we can use Algorithms 2 and 3 to calculate the total negative impact of responses on CIA parameters for GNDG nodes (see Table 7).

On the other hand, the total damage of attacks can be estimated by applying Algorithms 4 and 5. Table 8 shows the total damage of three potential attacks on GNDG nodes for the mentioned example.

The total goodness of responses can be estimated by using Eq. (3). Table 9 shows the total goodness of each response for three potential attacks in the mentioned network. The level of goodness is related to many parameters such as the weight of CIA parameters in UAG nodes, the impact of

![Fig. 7 – Global network dependency graph for the mentioned example.](image-url)
Fig. 8 – Response graph for the mentioned network.
Fig. 9 – The obtained uncertainty-aware attack graph for the mentioned network.
responses on CIA parameters, and the goodness of responses on attack paths.

We can estimate the total response cost on CIA parameters by using Eq. (6) and Eq. (7). According to the results shown in Table 10, host-level responses have the lowest impact on confidentiality and integrity; however from the availability point of view, they generate a significant cost. In addition, notification-level and attacker-level responses are the best responses from the availability point of view but result in high impact on confidentiality and integrity. An automated intrusion response system can apply this information to select a proper subset of responses. In Fig. 10 we compare the average cost of responses on CIA parameters.

5. Conclusion

In this paper we investigated the response management in intrusion response systems. One of the big challenges in IRSs that
Table 7 – Total impact of responses on GNDG nodes.

<table>
<thead>
<tr>
<th></th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
<th>R6</th>
<th>R7</th>
<th>R8</th>
<th>R9</th>
<th>R10</th>
<th>R11</th>
<th>R12</th>
<th>R13</th>
<th>R14</th>
<th>R15</th>
<th>R16</th>
<th>R17</th>
<th>R18</th>
<th>R19</th>
<th>R20</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Min</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.0300</td>
<td>0.0300</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.0700</td>
<td>0.0700</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I</td>
<td>Min</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.0300</td>
<td>0.0300</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.1000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>0.1000</td>
<td>0.1000</td>
<td>0.0700</td>
<td>0.0700</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
<td>0.3000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>Min</td>
<td>0.0100</td>
<td>0.0100</td>
<td>0.0100</td>
<td>0.0100</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0700</td>
<td>0.0700</td>
<td>0.0700</td>
<td>0.0700</td>
<td>0.0700</td>
<td>0.0700</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>0.0100</td>
<td>0.0100</td>
<td>0.0100</td>
<td>0.0100</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.1167</td>
<td>0.1167</td>
<td>0.1167</td>
<td>0.1167</td>
<td>0.1167</td>
<td>0.1167</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 10 – The average of response cost on each CIA parameter for the mentioned example.
is not considered in literature is the lack of standardization in intrusion responses. In this paper, our main goal was to propose a basic model for intrusion response management that can lead to a standard in future. We defined a multilevel model to classify the intrusion responses. The presented multilevel model provides a high-level view of intrusion responses that helps us in estimating the response cost and selecting appropriate responses against the attacks. We also proposed a foresight model to estimate the response cost by considering IDS alerts, network dependencies, attack damage, response impact, and probability of potential attacks. Moreover, we defined a data model to represent and exchange the intrusion response messages with a standard format. As part of our future work, we are planning to present an adaptive response cost estimating and optimum response selection approach based on the proposed intrusion response management subsystem.

Table 8 – Total damage of attacks on GNDG nodes.

<table>
<thead>
<tr>
<th></th>
<th>Att1</th>
<th>Att2</th>
<th>Att3</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Min</td>
<td>0.2000</td>
<td>0.0833</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>0.9000</td>
<td>0.1500</td>
</tr>
<tr>
<td>S2</td>
<td>Min</td>
<td>0.5000</td>
<td>0.0833</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>0.9000</td>
<td>0.1500</td>
</tr>
<tr>
<td>S3</td>
<td>Min</td>
<td>0.5000</td>
<td>0.0833</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>0.9000</td>
<td>0.1500</td>
</tr>
<tr>
<td>S4</td>
<td>Min</td>
<td>0.5000</td>
<td>0.0833</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>0.9000</td>
<td>0.1500</td>
</tr>
<tr>
<td>S5</td>
<td>Min</td>
<td>0.5000</td>
<td>0.0833</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>0.9000</td>
<td>0.1500</td>
</tr>
<tr>
<td>S6</td>
<td>Min</td>
<td>0.5000</td>
<td>0.0833</td>
</tr>
<tr>
<td></td>
<td>Max</td>
<td>0.9000</td>
<td>0.1500</td>
</tr>
</tbody>
</table>

Table 9 – The response goodness for three potential attacks in the mentioned example.

<table>
<thead>
<tr>
<th></th>
<th>Confidentiality</th>
<th>Integrity</th>
<th>Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>R1</td>
<td>0.0101</td>
<td>0.0198</td>
<td>0.0099</td>
</tr>
<tr>
<td>R2</td>
<td>0.0045</td>
<td>0.1998</td>
<td>0.0099</td>
</tr>
<tr>
<td>R3</td>
<td>0.0010</td>
<td>0.1998</td>
<td>0.0099</td>
</tr>
<tr>
<td>R4</td>
<td>0.0045</td>
<td>0.1998</td>
<td>0.0099</td>
</tr>
<tr>
<td>R5</td>
<td>0.0164</td>
<td>0.0198</td>
<td>0.0099</td>
</tr>
<tr>
<td>R6</td>
<td>0.0349</td>
<td>0.1998</td>
<td>0.0099</td>
</tr>
<tr>
<td>R7</td>
<td>0.0349</td>
<td>0.1998</td>
<td>0.0099</td>
</tr>
<tr>
<td>R8</td>
<td>0.0557</td>
<td>0.0198</td>
<td>0.0099</td>
</tr>
<tr>
<td>R9</td>
<td>0.0222</td>
<td>0.1998</td>
<td>0.0099</td>
</tr>
<tr>
<td>R10</td>
<td>0.0562</td>
<td>0.0198</td>
<td>0.0099</td>
</tr>
<tr>
<td>R11</td>
<td>0.0901</td>
<td>0.0198</td>
<td>0.0099</td>
</tr>
<tr>
<td>R12</td>
<td>0.0858</td>
<td>0.0198</td>
<td>0.0099</td>
</tr>
<tr>
<td>R13</td>
<td>0.1719</td>
<td>0.0198</td>
<td>0.0099</td>
</tr>
<tr>
<td>R14</td>
<td>0.1719</td>
<td>0.0198</td>
<td>0.0099</td>
</tr>
<tr>
<td>R15</td>
<td>0.0870</td>
<td>0.0198</td>
<td>0.0099</td>
</tr>
<tr>
<td>R16</td>
<td>0.2518</td>
<td>0.0198</td>
<td>0.0099</td>
</tr>
<tr>
<td>R17</td>
<td>0.2518</td>
<td>0.0198</td>
<td>0.0099</td>
</tr>
<tr>
<td>R18</td>
<td>0.1045</td>
<td>0.0198</td>
<td>0.0099</td>
</tr>
<tr>
<td>R19</td>
<td>0.1045</td>
<td>0.0198</td>
<td>0.0099</td>
</tr>
<tr>
<td>R20</td>
<td>0.0244</td>
<td>0.0198</td>
<td>0.0099</td>
</tr>
</tbody>
</table>
### Table 10 – Response cost on CIA parameters for the mentioned example.

<table>
<thead>
<tr>
<th></th>
<th>Confidentiality cost</th>
<th></th>
<th>Integrity cost</th>
<th></th>
<th>Availability cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
<td>Max</td>
<td>Min</td>
</tr>
<tr>
<td>R1</td>
<td>0.0587</td>
<td>0.7360</td>
<td>0.0587</td>
<td>0.7360</td>
<td>5.1672e−04</td>
</tr>
<tr>
<td>R2</td>
<td>0.0293</td>
<td>0.6391</td>
<td>0.0293</td>
<td>0.6391</td>
<td>1.8714e−04</td>
</tr>
<tr>
<td>R3</td>
<td>0.0587</td>
<td>0.7360</td>
<td>0.0587</td>
<td>0.7360</td>
<td>5.1672e−04</td>
</tr>
<tr>
<td>R4</td>
<td>0.0293</td>
<td>0.6391</td>
<td>0.0293</td>
<td>0.6391</td>
<td>1.8714e−04</td>
</tr>
<tr>
<td>R5</td>
<td>0.0639</td>
<td>1</td>
<td>0.0639</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>R6</td>
<td>0.0177</td>
<td>0.4043</td>
<td>0.0177</td>
<td>0.4043</td>
<td>0.0047</td>
</tr>
<tr>
<td>R7</td>
<td>0.0177</td>
<td>0.4043</td>
<td>0.0177</td>
<td>0.4043</td>
<td>0.0047</td>
</tr>
<tr>
<td>R8</td>
<td>0.0136</td>
<td>0.1916</td>
<td>0.0136</td>
<td>0.1916</td>
<td>0.0133</td>
</tr>
<tr>
<td>R9</td>
<td>0.0108</td>
<td>0.1982</td>
<td>0.0108</td>
<td>0.1982</td>
<td>0.0160</td>
</tr>
<tr>
<td>R10</td>
<td>0.0136</td>
<td>0.1890</td>
<td>0.0136</td>
<td>0.1890</td>
<td>0.0483</td>
</tr>
<tr>
<td>R11</td>
<td>0.0069</td>
<td>0.1494</td>
<td>0.0069</td>
<td>0.1494</td>
<td>0.0245</td>
</tr>
<tr>
<td>R12</td>
<td>0.0030</td>
<td>0.0887</td>
<td>0.0030</td>
<td>0.0887</td>
<td>0.0574</td>
</tr>
<tr>
<td>R13</td>
<td>0.0026</td>
<td>0.0904</td>
<td>0.0026</td>
<td>0.0904</td>
<td>0.0574</td>
</tr>
<tr>
<td>R14</td>
<td>0.0026</td>
<td>0.0904</td>
<td>0.0026</td>
<td>0.0904</td>
<td>0.0211</td>
</tr>
<tr>
<td>R15</td>
<td>0.0015</td>
<td>0.0664</td>
<td>0.0015</td>
<td>0.0664</td>
<td>0.0804</td>
</tr>
<tr>
<td>R16</td>
<td>0.0024</td>
<td>0.0299</td>
<td>0.0024</td>
<td>0.0299</td>
<td>0.0804</td>
</tr>
<tr>
<td>R17</td>
<td>0.0024</td>
<td>0.0299</td>
<td>0.0024</td>
<td>0.0299</td>
<td>0.0804</td>
</tr>
<tr>
<td>R18</td>
<td>0.0013</td>
<td>0.0272</td>
<td>0.0013</td>
<td>0.0272</td>
<td>0.0804</td>
</tr>
<tr>
<td>R19</td>
<td>0.0013</td>
<td>0.0272</td>
<td>0.0013</td>
<td>0.0272</td>
<td>0.0804</td>
</tr>
<tr>
<td>R20</td>
<td>0.0131</td>
<td>0.2506</td>
<td>0.0131</td>
<td>0.2506</td>
<td>0.0138</td>
</tr>
</tbody>
</table>

### Appendix

### Table 4 – Node details for the uncertainty-aware attack graph that is shown in Fig. 9.

1."execCode(dbServer,root)","OR",(0.2000 – 0.5070)
2."RULE 2 (remote exploit of a server program)","AND",(0.2000 – 0.5070)
3."netAccess(dbServer,dbProtocol,dbPort)","OR",(0.0000 – 0.9600)
4."RULE 5 (multi-hop access)","AND",(0.3000 – 0.6070)
5."hacl(webServer,dbServer,dbProtocol,dbPort)","LEAF",(1.0000 – 1.0000)
6."execCode(webServer,apache)","OR",(0.2000 – 0.5400)
7."RULE 2 (remote exploit of a server program)","AND",(0.2000 – 0.5400)
8."netAccess(webServer,httpProtocol,httpPort)","OR",(0.0000 – 0.9485)
9."RULE 5 (multi-hop access)","AND",(0.2000 – 0.7424)
10."hacl(workStation,webServer,httpProtocol,httpPort)","LEAF",(1.0000 – 1.0000)
11."execCode(workStation,normalAccount)","OR",(0.4000 – 0.8751)
12."RULE 0 (When a principal is compromised any machine he has an account on will also be compromised)","AND",(0.2000 – 0.4751)
13."canAccessHost(workStation)","OR",(0.4000 – 0.8751)
14."RULE 8 (Access a host through executing code on the machine)","AND",(0.4000 – 0.8751)
15."hasAccount(secretary,workStation,normalAccount)","LEAF",(1.0000 – 1.0000)
16."principalCompromised(secretary)","OR",(0.3000 – 0.6751)
17."RULE 12 (password sniffing)","AND",(0.3000 – 0.6751)
18."RULE 3 (remote exploit for a client program)","AND",(0.3000 – 0.6400)
19."accessMaliciousInput(workStation,secretary,IE)"","OR",(0.0000 – 0.9600)
20."RULE 22 (Browsing a malicious website)","AND",(0.4000 – 0.8000)
21."attackerLocated(internet)","LEAF",(1.0000 – 1.0000)
22."hacl(workStation,internet,httpProtocol,httpPort)","LEAF",(1.0000 – 1.0000)
23."inCompetent(secretary)","LEAF",(1.0000 – 1.0000)
24."RULE 24 (Browsing a compromised website)","AND",(0.2000 – 0.6070)
25."isWebServer(workServer)","LEAF",(1.0000 – 1.0000)
26."vulExists(workStation,'CVE-2009-1918',IE,remoteClient,privEscalation)","LEAF",(1.0000 – 1.0000)
27."RULE 6 (direct network access)","AND",(0.3000 – 0.7400)
30."vulExists(webServer,'CVE-2006-3747',httpd,remoteExploit,privEscalation)","LEAF",(1.0000 – 1.0000)
31."RULE 5 (multi-hop access)","AND",(0.2000 – 0.7424)
32."hacl(workStation,dbServer,dbProtocol,dbPort)","LEAF",(1.0000 – 1.0000)
33."networkServiceInfo(dbServer,mysql,dbProtocol,dbPort,root)","LEAF",(1.0000 – 1.0000)
34."vulExists(dbServer,'CVE-2009-2446',mysql,remoteExploit,privEscalation)","LEAF",(1.0000 – 1.0000)
Algorithm 11. $H\_\text{similarity}(ha_x,n_i)$

Input: $UAG, MRG$

Output: $H\_\text{similarity}(ha_x,n_i)$ (Similarity of $UAG$ node $n_i$ with hyper-alerts $ha_x$)

Initialization: $n_{\text{AND}} = \{n_i \in N | <n_i,d_i,>_D, d_i = \text{AND} \}$, $n_{\text{OR}} = \{n_i \in N | <n_i,d_i,>_D, d_i = \text{OR} \}$, $n_{\text{LEAF}} = \{n_i \in N | <n_i,d_i,>_D, d_i = \text{LEAF} \}$.

$\xi \equiv 0$

Extract SIP, DIP, Host, Client, Server, Protocol, Port, vuldID from $n_i$

if $(n_i \in n_{\text{LEAF}})$ then
  if $(n_i,\text{Host} \in ha_x, \text{DIP}) \& (n_i,\text{vuldID} \in ha_x, \text{vuldID})$ then
    $H\_\text{similarity}(ha_x,n_i) = 1$
  end if
  if $(n_i,\text{SIP} \in ha_x, \text{SIP}) \& (n_i,\text{DIP} \in ha_x, \text{DIP}) \& (n_i,\text{Protocol} \in ha_x, \text{Protocol}) \& (n_i,\text{Port} \in ha_x, \text{Port})$ then
    $H\_\text{similarity}(ha_x,n_i) = 1$
  end if
  if $(n_i,\text{Host} | \text{SIP} | \text{DIP} | \text{Client} | \text{Server} \in ha_x, \{\text{SIP} | \text{DIP}\})$ then
    $H\_\text{similarity}(ha_x,n_i) = H\_\text{similarity}(ha_x,n_i) + \xi$
  end if
  if $(n_i,\text{Protocol} \in ha_x, \text{Protocol})$ then
    $H\_\text{similarity}(ha_x,n_i) = H\_\text{similarity}(ha_x,n_i) + \xi$
  end if
  if $(n_i,\text{Port} \in ha_x, \{\text{Sp} | \text{Dp}\})$ then
    $H\_\text{similarity}(ha_x,n_i) = H\_\text{similarity}(ha_x,n_i) + \xi$
  end if
  if $(n_i,\text{vuldID} \in ha_x, \text{vuldID})$ then
    $H\_\text{similarity}(ha_x,n_i) = H\_\text{similarity}(ha_x,n_i) + \xi$
  end if
end if

if $(n_i \in n_{\text{OR}})$ then
  $H\text{similarity}(ha_x,n_i) = \max_k (H\text{similarity}(ha_x,\text{Parent}_k(n_i)))$
end if

if $(n_i \in n_{\text{AND}})$ then
  $H\text{similarity}(ha_x,n_i) = \prod_k (H\text{similarity}(ha_x,\text{Parent}_k(n_i)))$
end if

References


Fisch E. A Taxonomy and implementation of automated responses to intrusive behavior, Texas A&M University, 1996.


Shameli-Sendi A. System health monitoring and proactive response activation, Université de Montréal, Canada, 2013.


Uppuluri P, Sekar R. Experiences with specification-based intrusion detection, in 4th international symposium on recent advances in intrusion detection, Davis, CA, USA, 2001, pp. 172–89.


including advanced access control models, insider threats, cloud computing security, mobile privacy, privacy and security of online social networks, and usable security and privacy. He is a member of ACM and IEEE.

Abbas Ghaemi Bafghi was born on April 1973 in Bojnord, Iran. He received his BS degree in Applied Mathematics in Computer from Ferdowsi University of Mashhad, Iran in 1995. He received his MS and PhD degrees in Computer engineering from Amirkabir (Tehran Polytechnique) University of Technology, Iran in 1997 and 2004 respectively. He is member of Computer Society of Iran (CSI) and Iranian Society of Cryptology (ISC). He is an associated professor in Department of Computer Engineering, Ferdowsi University of Mashhad, Iran. His research interests are in cryptology and security and he has published more than 80 conference and journal papers.