Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

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.

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

[*] *Corresponding author.*
E-mail address: ghaemib@ferdowsi.um.ac.ir (A. Ghaemi-Bafghi).

http://dx.doi.org/10.1016/j.cose.2016.06.005
0167-4048/© 2016 Elsevier Ltd. All rights reserved.

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, midpoint, 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

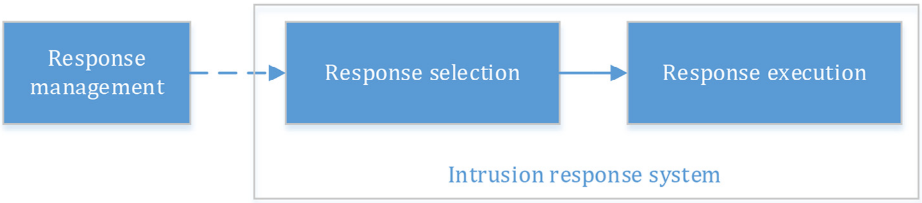


Fig. 1 – The role of intrusion response management in IRS.

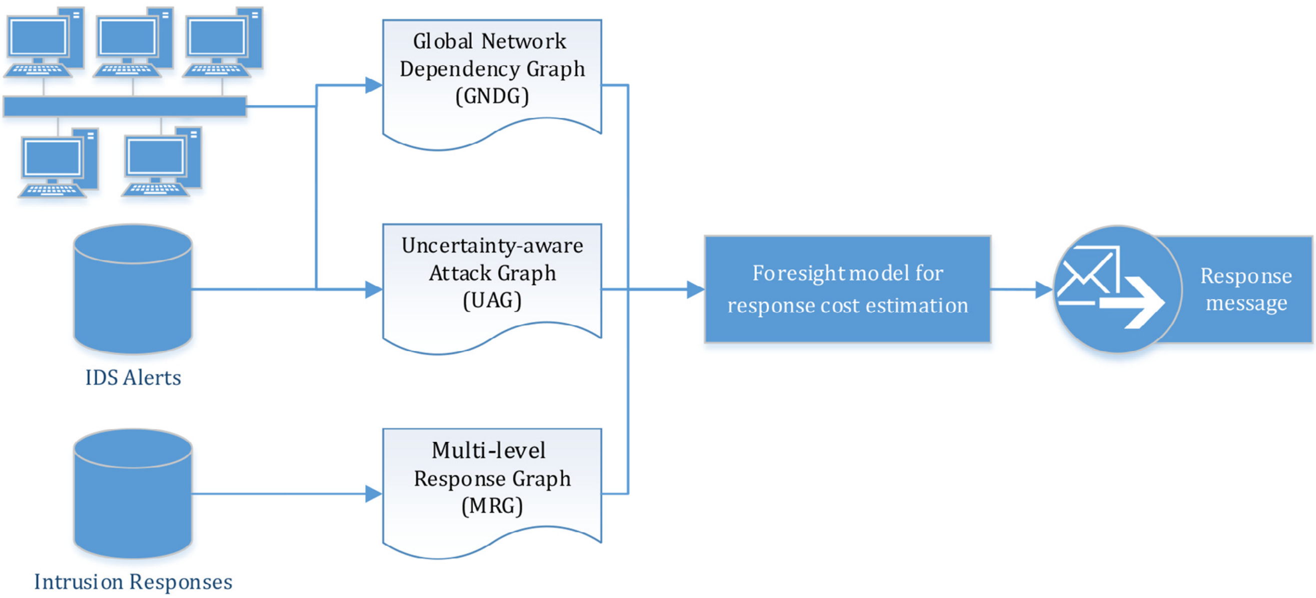

Global Network Dependency Graph (GNDG)
Uncertainty-aware Attack Graph (UAG)
Multi-level Response Graph (MRG)
IDS Alerts
Intrusion Responses
Foresight model for response cost estimation
Response message

Fig. 2 – The proposed foresight model for intrusion response management in IRS.

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 $GNDG = \langle H, E_H, L \rangle$ representing dependencies of services, processes, programs, files, users, and vulnerabilities toward each other.

- $H = \{\eta_1, \eta_2, \dots, \eta_m\}$ is the set of *GNDG* nodes including services, processes, programs, files, users, and vulnerabilities.
- $E_H = \{e_1, \dots, e_{|E_H|}\}$ is the set of directed edges that displays the all of dependencies between network elements.
- $L = \{l_1, \dots, l_{|E_H|}\}$ is a set of labels where $l_i \in \{\text{confidentiality}, \text{integrity}, \text{availability}\}$ indicates the label of edge e_i.

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

Definition 2. $DM^{(M)}$ 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,
- $DM^{(M)}(\eta_i, \eta_j) = 0$ if η_i and η_j are independent from each other,
- $DM^{(M)}(\eta_i, \eta_j) = 1$ if η_i is dependent on η_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 $UAG = \langle N, E_N, D, Pr, C, G \rangle$, where:

- $N = \{n_1, n_2, \dots, n_v\}$ is the set of *UAG* nodes.
- E_N is the set of uncertainty-aware attack graph edges that shows the relationships between vulnerabilities.
- Dis a set of pairs $\langle n_i, d_i \rangle, i = 1, \dots, v$ where $d_i \in \{LEAF, AND, OR\}$ denotes the type of node n_i.
- $Pr = \{\hat{P}(n_1), \hat{P}(n_2), \dots, \hat{P}(n_v)\}$ is the set of imprecise probabilities, where $\hat{P}(n_i) = \langle \underline{P}(n_i), \overline{P}(n_i) \rangle$. $\underline{P}(n_i) = \sup\{P(n_i): P \in \rho\}$ indicates the lower probability and $\overline{P}(n_i) = \inf\{P(n_i): P \in \rho\}$ shows the upper probability of each node in the graph and ρ is the set of probability distributions. In the classical case of probability theory, the lower bound is always equal to the upper bound.
- Cis 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 $\langle n_i, d_i \rangle \in D$, $d_i = \{LEAF\}$ then $\hat{P}(n_i) = \langle 1, 1 \rangle$.

  - If $\langle n_i, d_i \rangle \in D$, $d_i = \{AND\}$ then $\hat{P}(n_i) \le \prod \hat{P}(Predecessor(n_i))$
  - If $\langle n_i, d_i \rangle \in D$, $d_i = \{OR\}$ then $\hat{P}(n_i) \le 1 - \prod \left(1 - \hat{P}(Predecessor(n_i))\right)$
- $G \subseteq N$ is the set of the attacker's final goal.

We can also generate a dependency matrix by analyzing the uncertainty-aware attack graph.

Definition 4. *DMAg* is an $v \times v$ matrix which indicates dependencies between nodes in the uncertainty-aware attack graph where:

- $DMAg(n_i, n_j) = 0$ if n_i and n_j are independent from each other,
- $DMAg(n_i, n_j) = 1$ if n_i is dependent on n_j.

3.3. *Multi-level response graph (MRG)*

As Fig. 3illustrates, the intrusion responses can be categorized into different levels of impact as follows:

- The Notification-level responses are the lowest-level of responding that react to attacks by generating report or alarm.
- The Attacker-level responses affect the attacker's system directly (e.g. Block attacker IP in firewall).
- The Vulnerability-level responses are the lowest response level that contains the atomic countermeasures to eliminate the known vulnerabilities such as CVE countermeasures (Common vulnerability and exposures, 2015) (e.g. Patch or update the compromised software).
- File-level responses block a file or change its access permission.
- User-level responses block a user or reduce its privilege.
- Service-level responses block the compromised processes, services or ports to mitigate the attack damage, but it may impose an undesirable cost to authorized users.
- The Host-level responses consist of the most costly responding such as shutting down the victim machine.

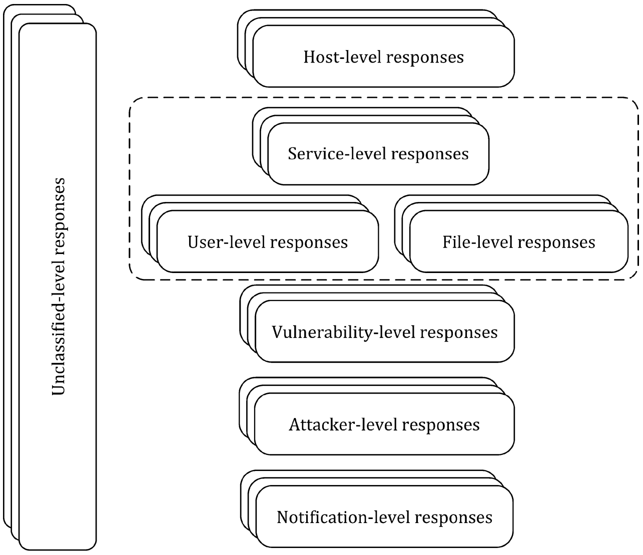


Fig. 3 – The impact level of intrusion responses.

- The other responses that cannot be categorized into any mentioned above levels are called Unclassified-level responses (e.g. Enable additional IDS).

This classification helps us to compare different responses according to the impact cost and find relationships between them. We can also model the intrusion responses as a multi-level graph to help the IRSs in choosing the appropriate responses.

Definition 5. We represent the set of intrusion responses as a multi-level response graph $MRG = (R, E_R, C, A)$where:

- $R = \{R_1, R_2, \ldots, R_n\}$ is the set of *MRG* nodes where each node is an intrusion response.
- E_R is the set of edges that shows the relationships between intrusion responses.
- Cis a set of pairs $\langle R_i, c_i \rangle, i = 1, \ldots, n$ where $c_i \in [0,1]$ denotes the response cost of R_i.
- A is a set of pairs $\langle R_i, a_i \rangle, i = 1, \ldots, n$ where $a_i \in \{Yes, No\}$ denotes the activation statement of a response.
- We use the mentioned 9-level of impacts for the proposed response graph.

3.4. *Foresight model for response cost estimation*

Response cost arises from negative impact of applying various responses on the network assets. Estimating a reliable response cost is an important step in selection of suitable response strategy in IRSs. Therefore, we propose a foresight model to measure the response cost with the aid of new metrics derived from IDS alerts, network dependencies, attack damage, response impact, and probability of potential attacks. We also consider the uncertainty of cost analysis due to incomplete or imprecise estimation of these metrics to find the optimistic and pessimistic view of response costs.

In this paper, we describe the response cost as two separated parts: (1) the negative part and (2) the positive part. In the negative part, we deal with all of negative impacts of a response, while in the positive part, we focus on the positive aspects of a response. Therefore, we can define the negative part of response R_i in a specific time as the following equation:

$$Cost_{-}^{(t)} = \tilde{C}_{op}^{(t)}(R_i) + \sum_{M \in \{C,I,A\}} \sum_{j=1}^{m} \tilde{\omega}^{(t)}(M, \eta_j).\tilde{C}_{im}^{(t)}(M, R_i, \eta_j) \qquad (1)$$

where $\tilde{C}_{op}^{(t)}(R_i) = \langle \underline{C}_{op}^{(t)}(R_i), \overline{C}_{op}^{(t)}(R_i) \rangle$ shows the minimum and maximum operational costs of response R_i which can be defined by an expert, $\tilde{\omega}^{(t)}(M, \eta_j)$ indicates the weight of each CIA parameter in node η_j, and $\tilde{C}_{im}^{(t)}(M, R_i, \eta_j)$ is the total negative impact of response R_i on each CIA parameter in node η_j. We can use Algorithm 1 to estimate the weight of CIA parameters on each node in the global network dependency graph. In this algorithm, we analyze relationships between nodes in the global network dependency graph to extract some useful constraints. Then, we can find the minimum and maximum weights by solving two linear programming problems.

Algorithm 1 – Computing the weight of CIA parameters for all nodes in *GNDG*

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

Output: $\tilde{\omega}^{(t)}(M,\eta_j)$ (The weight of each CIA parameter in all *GNDG* nodes.

```
1: Extract the set of constraints ( $C$ ) from global network dependency graph or have them defined by an expert
    a. If $\eta_j$ is a leaf node in the global network dependency graph and $M=\{C,I,A\}$ is a critical parameter in $\eta_j$
       then $\tilde{\omega}^{(t)}(M,\eta_j)=<1,1>$
    b. According to the global network dependency graph $\tilde{\omega}^{(t)}(M,\eta_j)\le \max\left(\tilde{\omega}^{(t)}(M,\mathrm{Successor}(\eta_j))\right)$
2: Compute the weight of each CIA parameter in all $GNDG$ nodes
    a. $\underline{\omega}^{(t)}(M,\eta_j)=\arg\min_{\forall C}\sum_{\eta_j\in H}\tilde{\omega}^{(t)}(M,\eta_j)$
    b. $\overline{\omega}^{(t)}(M,\eta_j)=\arg\max_{\forall C}\sum_{\eta_j\in H}\tilde{\omega}^{(t)}(M,\eta_j)$
    c. $\tilde{\omega}^{(t)}(M,\eta_j)=<\underline{\omega}^{(t)}(M,\eta_j),\ \overline{\omega}^{(t)}(M,\eta_j)>$
```

We define Algorithms 2 and 3 to estimate $\tilde{C}^{(t)}_{im}(M, R_i, \eta_j)$ 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 $RH^{(M)}$ 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: $RH^{(M)}$ (The direct impact of applying responses on each CIA parameter of nodes in *GNDG*)

```
Extract IP, servID, vulID, fileID, userID from each $\eta_j$ and $R_i$
for each CIA parameter $M=\{C,I,A\}$
    for each $R_i$
        for each $\eta_j$
            $RH^{(M)}(i,j)=0$
            if ( $R_i$ is a host-level response) then
                if $(R_i.IP==\eta_j.IP)$ then
                    $RH^{(M)}(i,j)=1$
                end if
            end if
            if ( $R_i$ is a notification/attacker/service/vulnerability/file/user-level response) then
                if $(R_i.IP==\eta_j.IP)\&(R_i.\{ServID|\,vulID|\,fileID|\,userID\}==\eta_j.\{ServID|\,vulID|\,fileID|\,userID\})$ then
                    $RH^{(M)}(i,j)=1$
                end if
            end if
            if ( $R_i$ is an unclassified-level response) then
                $RH^{(M)}(i,j)=\varepsilon$
            end if
        end for
    end for
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 $\tilde{C}_{im}(M, R_i)$ 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 $\mathrm{RImpact}^{(M)}(R, H)$ 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 $\tilde{C}_{im}(M, R_i)$ and $\mathrm{RImpact}^{(M)}(R_i, \eta_j)$.

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: $\underline{C}^{(t)}_{im}(M, R_i, \eta_j)$ (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 $\tilde{C}_{im}(M, R_i)$ (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. $\otimes : RH^{(M)}{}_{(n\times m)} \times DM^{(M)}{}_{(m\times m)} \rightarrow \mathrm{RImpact}^{(M)}_{(n\times m)}$
       b. $\mathrm{RImpact}^{(M)}(R, H) = RH^{(M)}{}_{(n\times m)} \otimes DM^{(M)}{}_{(m\times m)}$
       c. $\mathrm{RImpact}^{(M)}(R_i, H) = \mathrm{RImpact}^{(M)}(R_i, H) + \left[\sum_q \mathrm{temp}^{(i)}(1,q), \sum_q \mathrm{temp}^{(i)}(2,q), \ldots, \sum_q \mathrm{temp}^{(i)}(m,q)\right]$ where:
           • $\mathrm{temp}^{(i)}(r,q) = RH^{(M)}(i,q).DMI^{(M)}(r,q),\ i = 1,\ldots,n,\ q = 1,\ldots,m,\ r = 1,\ldots,m$
           • $DMI^{(M)} = DM^{(M)} + I$
3: Compute the total negative impact of applying responses
       a. $\underline{C}^{(t)}_{im}(M, R_i, \eta_j) = \min\{\tilde{C}_{im}(M, R_i).\ \mathrm{RImpact}^{(M)}(R_i, \eta_j)\}$
       b. $\overline{C}^{(t)}_{im}(M, R_i, \eta_j) = \max\{\tilde{C}_{im}(M, R_i).\ \mathrm{RImpact}^{(M)}(R_i, \eta_j)\}$
       c. $\tilde{C}^{(t)}_{im}(M, R_i, \eta_j) = < \underline{C}^{(t)}_{im}(M, R_i, \eta_j), \overline{C}^{(t)}_{im}(M, R_i, \eta_j) >$
```

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

	Confidentiality	Integrity	Availability
Notification-level	Very high	Very high	Very low
Attacker-level	Low–high	Low–high	Low–high
Vulnerability-level	Low–very low	Low–very low	Low–very low
File-level	Low–very low	Low–very low	High–very high
User-level	Low–very low	Low–very low	High–very high
Service-level	Low–very low	Low–very low	High–very high
Host-level	Very low	Very low	Very high
Unclassified-level	Low–very high	Low–very high	Low–very high

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

$$\mathrm{Cost}^{(t)}_{+} = \sum_{k=1}^{l} \tilde{D}^{(t)}(M, Att_k, \eta_j).G^{(t)}(M, R_i, Att_k) \qquad (2)$$

where $\tilde{D}^{(t)}(M, Att_k, \eta_j)$ shows the total damage of attacks on each CIA parameter in node η_j and $G^{(t)}(M, R_i, Att_k)$ is the total goodness of response R_i against attack Att_k 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 $AttH^{(M)}$ 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.

Algorithm 4. Computing the direct attack damage on CIA parameters

```
Input: $GNDG$, the number of network attacks ($l$), the number of nodes in $GNDG$ ($m$),
Output: $Att\mathrm{H}^{(M)}$ (The direct attack damage on each CIA parameter of nodes in the global network dependency graph)
Extract IP, servID, vulID, fileID, userID from $\eta_j$, $Att_k$
for each CIA parameter $M=\{C,I,A\}$
    for each $Att_k$
        for each $\eta_j$
            $Att\mathrm{H}^{(M)}(k,j)=0$
            if $(Att_k.IP==\eta_j.IP)\&(Att_k.\{ServID|\ vulID|\ fileID|\ userID\}==\eta_j.\{ServID|\ vulID|\ fileID|\ userID\})$ then
                $Att\mathrm{H}^{(M)}(k,j)=1$
            end if
            if $(Att_k.IP==\eta_j.IP)$ then
                $Att\mathrm{H}^{(M)}(k,j)=\varepsilon$
            end if
        end for
    end for
end for
```

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 ⊗ to estimate $\mathrm{AImpact}^{(M)}(\mathrm{Att},\mathrm{H})$ as the amount of damage on each CIA parameter of nodes in the global network dependency graph by using $\mathrm{AttH}^{(M)}$ and $DM^{(M)}$ matrices. The total damage of attack Att_k can be computed by multiplying $\tilde{\omega}^{(t)}(\mathrm{M},\eta_j)$ and $\mathrm{AImpact}^{(M)}(\mathrm{Att}_k,\eta_j)$.

Algorithm 5 – Computing the total damage of attacks

```
Input: $\widetilde{\omega}^{(t)}(M,\eta_j)$, $DM^{(M)}$, $Att\mathrm{H}^{(M)}$, the number of network attacks ($l$), the number of nodes in $GNDG$ ($m$)
Output: $\widetilde{D}^{(t)}(M,Att_k,\eta_j)$ (The total damage of attacks on each CIA parameter of nodes in the global network dependency graph)
Compute the total damage of attacks on each CIA parameter of nodes in $GNDG$ ($\mathrm{AImpact}^{(M)}(Att_k,\eta_j)$) where:
    a. $\otimes: Att\mathrm{H}^{(M)}{}_{(l\times m)} \otimes DM^{(M)}{}_{(m\times m)} \rightarrow \mathrm{AImpact}^{(M)}{}_{(l\times m)}$
    b. $\mathrm{AImpact}^{(M)}(Att,\mathrm{H}) = Att\mathrm{H}^{(M)}{}_{(l\times m)} \otimes DM^{(M)}{}_{(m\times m)}$
    c. $\mathrm{AImpact}^{(M)}(Att_k,\mathrm{H}) = \mathrm{AImpact}^{(M)}(Att_k,\mathrm{H}) + \left[\sum_q \mathrm{temp}^{(k)}(1,q), \sum_q \mathrm{temp}^{(k)}(2,q), \ldots, \sum_q \mathrm{temp}^{(k)}(m,q)\right]$ where:
        - $\mathrm{temp}^{(\mathrm{k})}(r,q) = Att\mathrm{H}^{(M)}(k,q).DMI^{(M)}(r,q)$, $k=1,\ldots,l$, $q=1,\ldots,m$, $r=1,\ldots,m$
        - $DMI^{(M)} = DM^{(M)} + I$
Compute the total damage of attacks
    a. $\underline{D}^{(t)}(M,Att_k,\eta_j) = \min\{\widetilde{\omega}^{(t)}(M,\eta_j).\ \mathrm{AImpact}^{(M)}(Att_k,\eta_j)\}$
    b. $\overline{D}^{(t)}(M,Att_k,\eta_j) = \max\{\widetilde{\omega}^{(t)}(M,\eta_j).\ \mathrm{AImpact}^{(M)}(Att_k,\eta_j)\}$
    c. $\widetilde{D}^{(t)}(M,Att_k,\eta_j) = <\underline{D}^{(t)}(M,Att_k,\eta_j), \overline{D}^{(t)}(M,Att_k,\eta_j)>$
```

We also define the total goodness of responses with the following equation:

$$G^{(t)}(M, R_i, Att_k) = \sum_{q \in Path_{Att_k}} Gp^{(t)}\left(M, R_i, Path^{q}_{Att_k}\right) \Big/ \left|Path_{Att_k}\right| \qquad (3)$$

where $Path_{Att_k}$ refers to the attack paths which can be extracted from the uncertainty-aware attack graph. We can also find the goodness of response R_i on the attack path $Path^{q}_{Att_k}$ by Eq. (4):

$$Gp^{(t)}\left(M, R_i, Path^{q}_{Att_k}\right) = \sum_{n_{Att_k} \in Path^{q}_{Att_k}} \left\{\tilde{\omega}^{(t)}(M, n_{Att_k}).\tilde{C}^{(t)}_{im}(M, R_i).Gn(R_i, n_{Att_k})\right\} \qquad (4)$$

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

$$\tilde{\omega}^{(t)}(M, n_{Att_k}) = \tilde{\omega}^{(t)}(M, \eta_j).HN(\eta_j, n_{Att_k}) \qquad (5)$$

where $\tilde{\omega}^{(t)}(M, \eta_j)$ indicates the weight of each CIA parameter in all *GNDG* nodes, and $HN(\eta_j, n_{Att_k})$ shows the total relationships between *GNDG* nodes and *UAG* nodes. We define Algorithms 6 and 7 to compute $HN(\eta_j, n_{Att_k})$. Algorithm 6 estimates the direct relationships between *GNDG* nodes and *UAG* nodes and returns the H*Ag* 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)
Extract IP, servID, vulID, fileID, userID from $\eta_j$ and $n_i$
for each $\eta_j$
    for each $n_i$
        $HAg(j,i) = 0$
        if $(n_i.IP == \eta_j.IP) \& (n_i.\{ServID| vulID| fileID| userID\} == \eta_j.\{ServID| vulID| fileID| userID\})$ then
            $HAg(j,i) = 1$
        end if
    end for
end for
```

Also, Algorithm 7 uses H*Ag* and *DMAg* matrices to estimate $HN(\eta_j, n_{Att_k})$. 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: $HN(\eta_j, n_{Att_k})$ (The relationships between $GNDG$ nodes and $UAG$ nodes)
Compute the total relationships between $GNDG$ nodes and $UAG$ nodes where:
    a. $\otimes : HAg_{(m\times v)} \otimes DMAg_{(v\times v)} \rightarrow HN_{(m\times v)}$
    b. $HN(H, n_{Att_k}) = HAg_{(m\times v)} \otimes DMAg_{(v\times v)}$
    c. $HN(\eta_j, n_i) = HN(\eta_j, n_i) + \left[\sum_q temp^{(j)}(1,q), \sum_q temp^{(j)}(2,q), \ldots, \sum_q temp^{(j)}(v,q)\right]$ where:
        • $temp^{(j)}(r,q) = HAg(j,q).DMAgI^{(M)}(r,q)$, $j = 1,\ldots,m$, $q = 1,\ldots,v$, $r = 1,\ldots,v$
        • $DMAgI^{(M)} = DMAg^{(M)} + I$
```

We define Algorithms 8 and 9 to estimate $Gn(R_i, n_{Att_k})$ 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)
Extract IP, servID, vulID, fileID, userID from each $n_j$ and $R_i$
for each $R_i$
    for each $n_j$
        $RAg(i,j)=0$
        if ( $R_i$ is a host-level response) then
            if $(R_i.IP==n_j.IP)$ then
                $RAg(i,j)=1$
            end if
        end if
        if ( $R_i$ is a notification/attacker/service/vulnerability/file/user-level response) then
            if $(R_i.IP==n_j.IP)\&(R_i.\{ServID|vulID|fileID|userID\}==n_j.\{ServID|vulID|fileID|userID\})$ then
                $RAg(i,j)=1$
            end if
        end if
        if ( $R_i$ is an unclassified-level response) then
            $RAg(i,j)=\varepsilon$
        end if
    end for
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.

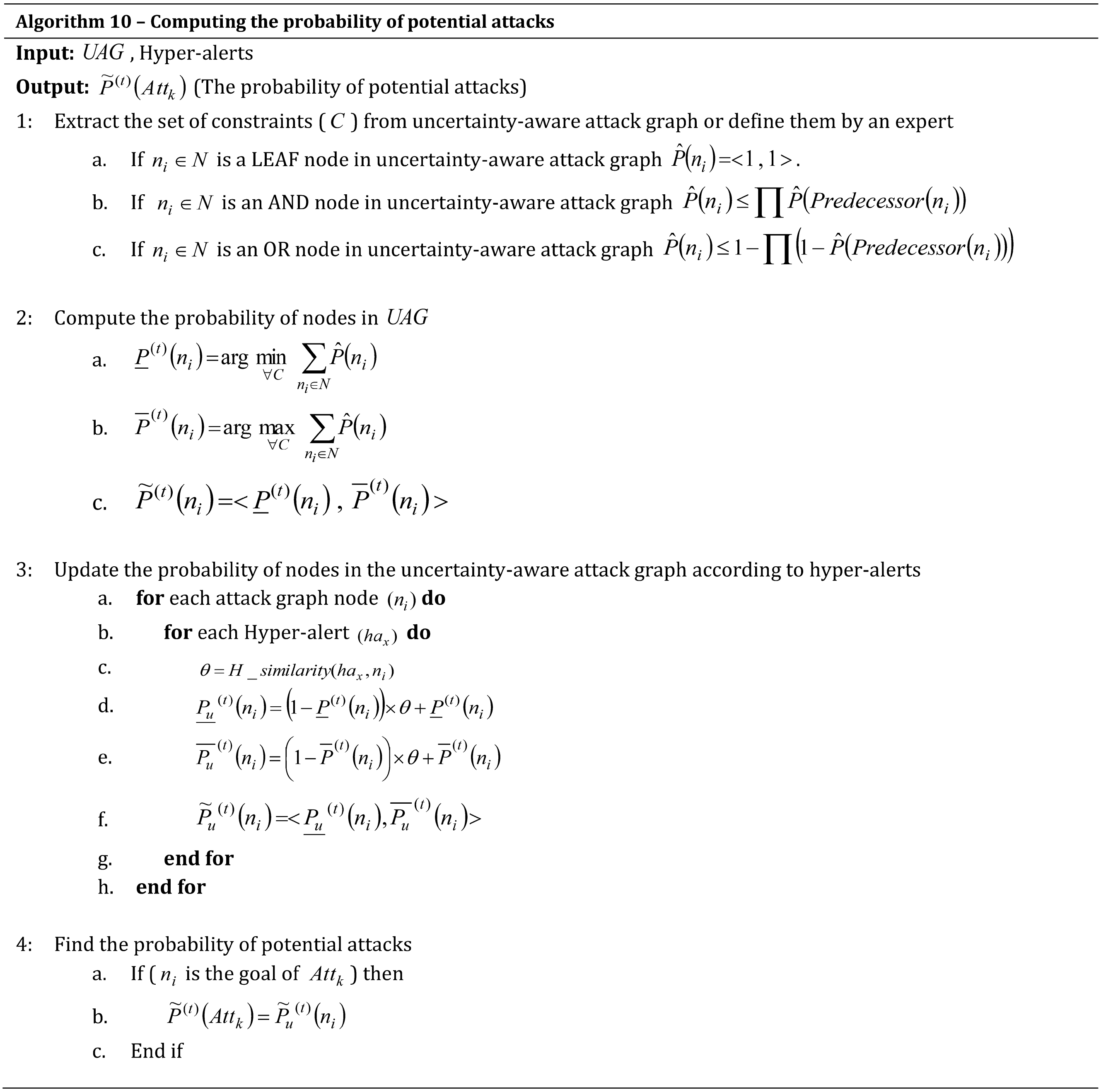
Algorithm 9 – Computing the total goodness of responses on the UAG nodes

```
Input: $RAg$, $DMAg$, the number of responses ($n$), the number of nodes in $UAG$ ($v$)
Output: $\mathrm{G}n(R_i, n_{Att_k})$ (The total goodness of responses on the $UAG$ nodes)
Compute the total goodness of responses on the $UAG$ nodes ($\mathrm{G}n(R_i, n_{Att_k})$) where:
    a. $\otimes : RAg_{(n\times v)} \otimes DMAg_{(v\times v)} \rightarrow \mathrm{G}n_{(n\times v)}$
    b. $\mathrm{G}n(R, n_{Att}) = RAg_{(n\times v)} \otimes DMAg_{(v\times v)}$
    c. $Gn(R_i, n_{Att}) = \mathrm{G}n(R_i, n_{Att}) + \left[\sum_q \mathrm{temp}^{(i)}(1,q), \sum_q \mathrm{temp}^{(i)}(2,q), \ldots, \sum_q \mathrm{temp}^{(i)}(v,q)\right]$ where:
        • $\mathrm{temp}^{(i)}(r,q) = RAg(i,q).DMAgI^{(M)}(r,q)$, $i=1,\ldots,n$, $q=1,\ldots,v$, $r=1,\ldots,v$
        • $DMAgI^{(M)} = DMAg^{(M)} + I$
```

Now, we can estimate the range of response cost by combining $Cost_{-}^{(t)}$ and $Cost_{+}^{(t)}$. In the optimistic situation, we define the response cost in a specific time as follows:

$$\min\left\{\left(1-\tilde{P}^{(t)}(Att_k)\right)\left(Cost_{-}^{(t)}\right)/\tilde{P}^{(t)}(Att_k)\left(Cost_{+}^{(t)}\right)\right\} \tag{6}$$

where $\tilde{P}^{(t)}(Att_k)$ 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 $H\_similarity(ha_x, n_i)$ that returns the similarity of uncertainty-aware attack graph node n_i with hyper-alerts ha_j. 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 (GhasemiGol and Ghaemi-Bafghi, 2015).



Algorithm 10 – Computing the probability of potential attacks

```
Input: $UAG$ , Hyper-alerts
Output: $\tilde{P}^{(t)}(Att_k)$ (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 $\hat{P}(n_i) = <1, 1>$.
    b. If $n_i \in N$ is an AND node in uncertainty-aware attack graph $\hat{P}(n_i) \le \prod \hat{P}(Predecessor(n_i))$
    c. If $n_i \in N$ is an OR node in uncertainty-aware attack graph $\hat{P}(n_i) \le 1 - \prod(1 - \hat{P}(Predecessor(n_i)))$

2: Compute the probability of nodes in $UAG$
    a. $\underline{P}^{(t)}(n_i) = \arg\min_{\forall C} \sum_{n_i \in N} \hat{P}(n_i)$
    b. $\overline{P}^{(t)}(n_i) = \arg\max_{\forall C} \sum_{n_i \in N} \hat{P}(n_i)$
    c. $\tilde{P}^{(t)}(n_i) = <\underline{P}^{(t)}(n_i), \overline{P}^{(t)}(n_i)>$

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 $(ha_x)$ do
    c.         $\theta = H\_similarity(ha_x, n_i)$
    d.         $\underline{P_u}^{(t)}(n_i) = (1 - \underline{P}^{(t)}(n_i)) \times \theta + \underline{P}^{(t)}(n_i)$
    e.         $\overline{P_u}^{(t)}(n_i) = (1 - \overline{P}^{(t)}(n_i)) \times \theta + \overline{P}^{(t)}(n_i)$
    f.         $\tilde{P}_u^{(t)}(n_i) = <\underline{P_u}^{(t)}(n_i), \overline{P_u}^{(t)}(n_i)>$
    g.     end for
    h. end for

4: Find the probability of potential attacks
    a. If ( $n_i$ is the goal of $Att_k$ ) then
    b.     $\tilde{P}^{(t)}(Att_k) = \tilde{P}_u^{(t)}(n_i)$
    c. End if
```

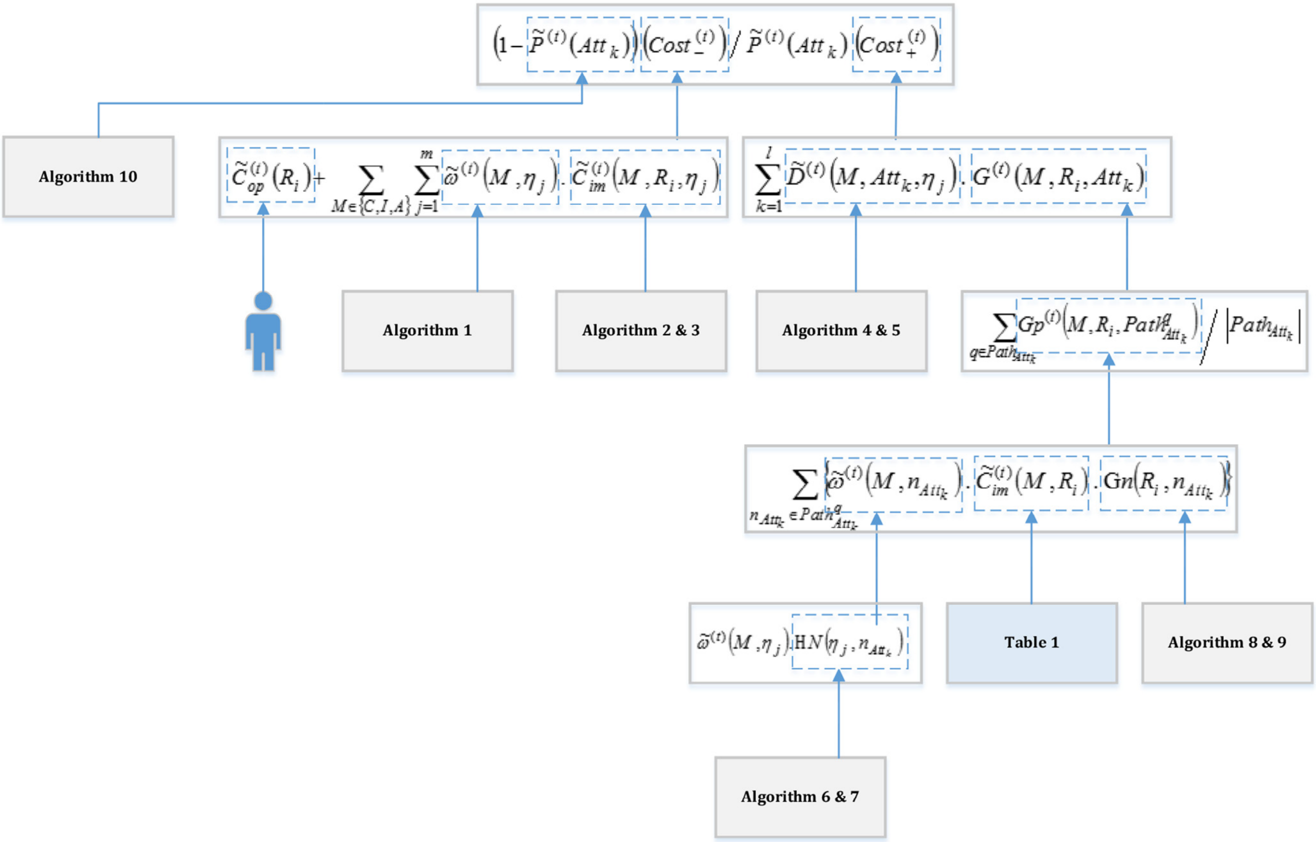

$\left(1-\tilde{P}^{(t)}(Att_k)\right)\left(Cost_-^{(t)}\right)/\tilde{P}^{(t)}(Att_k)\left(Cost_+^{(t)}\right)$
Algorithm 10
$\tilde{C}_{op}^{(t)}(R_i)+\sum_{M\in\{C,I,A\}}\sum_{j=1}^{m}\tilde{\varpi}^{(t)}(M,\eta_j).\tilde{C}_{im}^{(t)}(M,R_i,\eta_j)$
$\sum_{k=1}^{l}\tilde{D}^{(t)}(M,Att_k,\eta_j).G^{(t)}(M,R_i,Att_k)$
Algorithm 1
Algorithm 2 & 3
Algorithm 4 & 5
$\sum_{q\in Path_{Att_k}}Gp^{(t)}(M,R_i,Path_{Att_k}^{q})/\left|Path_{Att_k}\right|$
$\sum_{n_{Att_k}\in Path_{Att_k}^{q}}\{\tilde{\varpi}^{(t)}(M,n_{Att_k}).\tilde{C}_{im}^{(t)}(M,R_i).Gn(R_i,n_{Att_k})\}$
$\tilde{\varpi}^{(t)}(M,\eta_j).HN(\eta_j,n_{Att_k})$
Table 1
Algorithm 8 & 9
Algorithm 6 & 7

Fig. 4 – The process of presented response cost estimating model.

Similarly, in the pessimistic situation we can use the following equation:

$$\max\left\{\left(1-\tilde{P}^{(t)}(Att_k)\right)\left(Cost_-^{(t)}\right)/\tilde{P}^{(t)}(Att_k)\left(Cost_+^{(t)}\right)\right\} \tag{7}$$

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, $m$is 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, l is the number of network attacks, v is 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 $O(nm)+O(m^2)+O(nm^3)$. Similarly, Algorithms 4 and 5 have the time complexity of $O(lm)+O(m^2)+O(lm^3)$ 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 $\{O(mv)+O(v^2)+O(mv^3)\}+\{O(nv)+O(v^2)+O(nv^3)\}$. 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 $O(v_C)$.

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

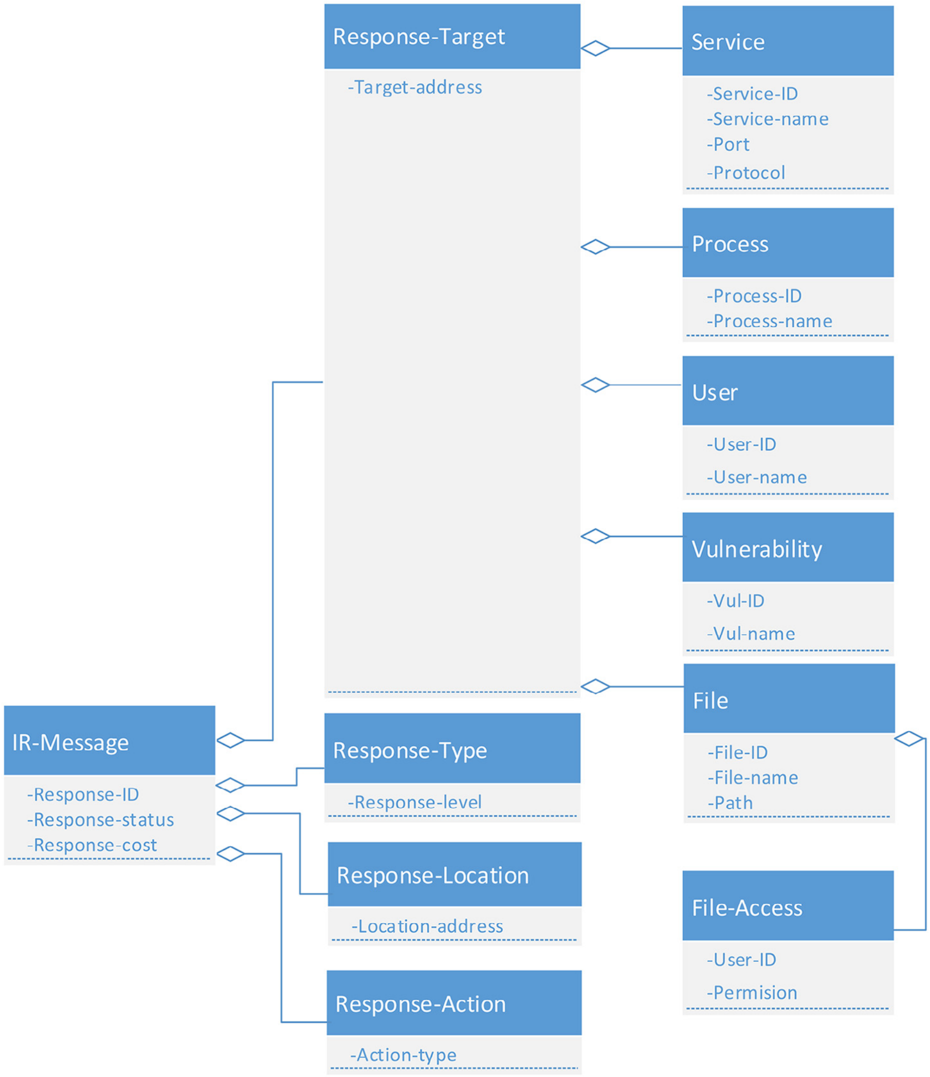


Fig. 5 – Intrusion response message format.

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

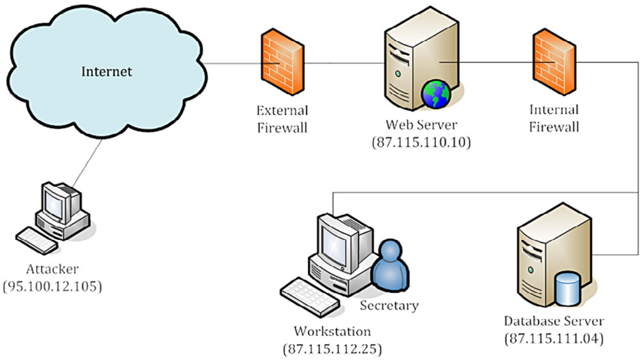


Fig. 6 – Network topology for the mentioned example.

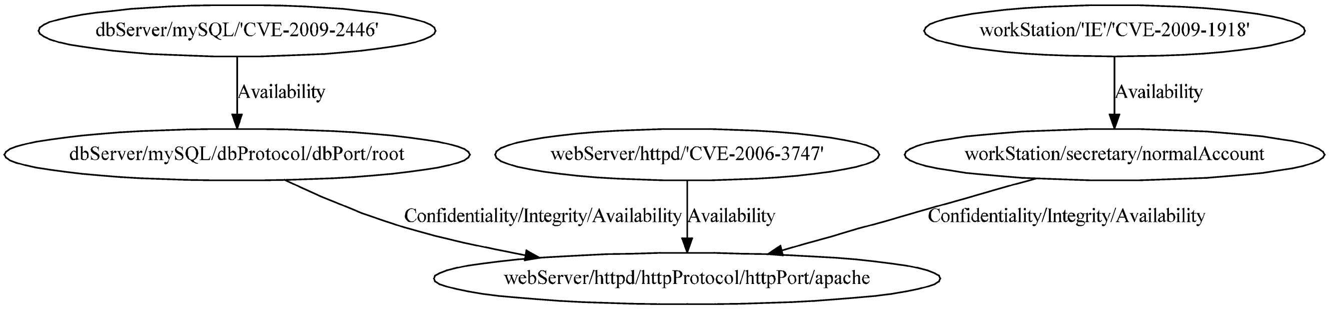

dbServer/mySQL/'CVE-2009-2446'
Availability
workStation/'IE'/'CVE-2009-1918'
Availability
dbServer/mySQL/dbProtocol/dbPort/root
webServer/httpd/'CVE-2006-3747'
workStation/secretary/normalAccount
Confidentiality/Integrity/Availability
Availability
Confidentiality/Integrity/Availability
webServer/httpd/httpProtocol/httpPort/apache

Fig. 7 – Global network dependency graph for the mentioned example.

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

Table 2 – The weight of nodes in the global network dependency graph shown in Fig. 7.

GNDG nodes	Confidentiality weight		Integrity weight		Availability weight	
	Min	Max	Min	Max	Min	Max
dbServer/mySQL/‘CVE-2009-2446”	0	0.2000	0	0.2000	0	0.2000
dbServer/mySQL/dbProtocol/dbPort/root	0.5000	0.9000	0.5000	0.9000	0.5000	0.9000
webServer/httpd/’CVE-2006-3747’	0	0.2000	0	0.2000	0	0.2000
webServer/httpd/httpProtocol/httpPort/apache	0.6000	1.0000	0.6000	1.0000	0.6000	1.0000
workStation/secretary/normalAccount	0.5000	0.9000	0.5000	0.9000	0	0.2000
workStation/‘IE’/’CVE-2009-1918’	0	0.2000	0	0.2000	0	0.2000

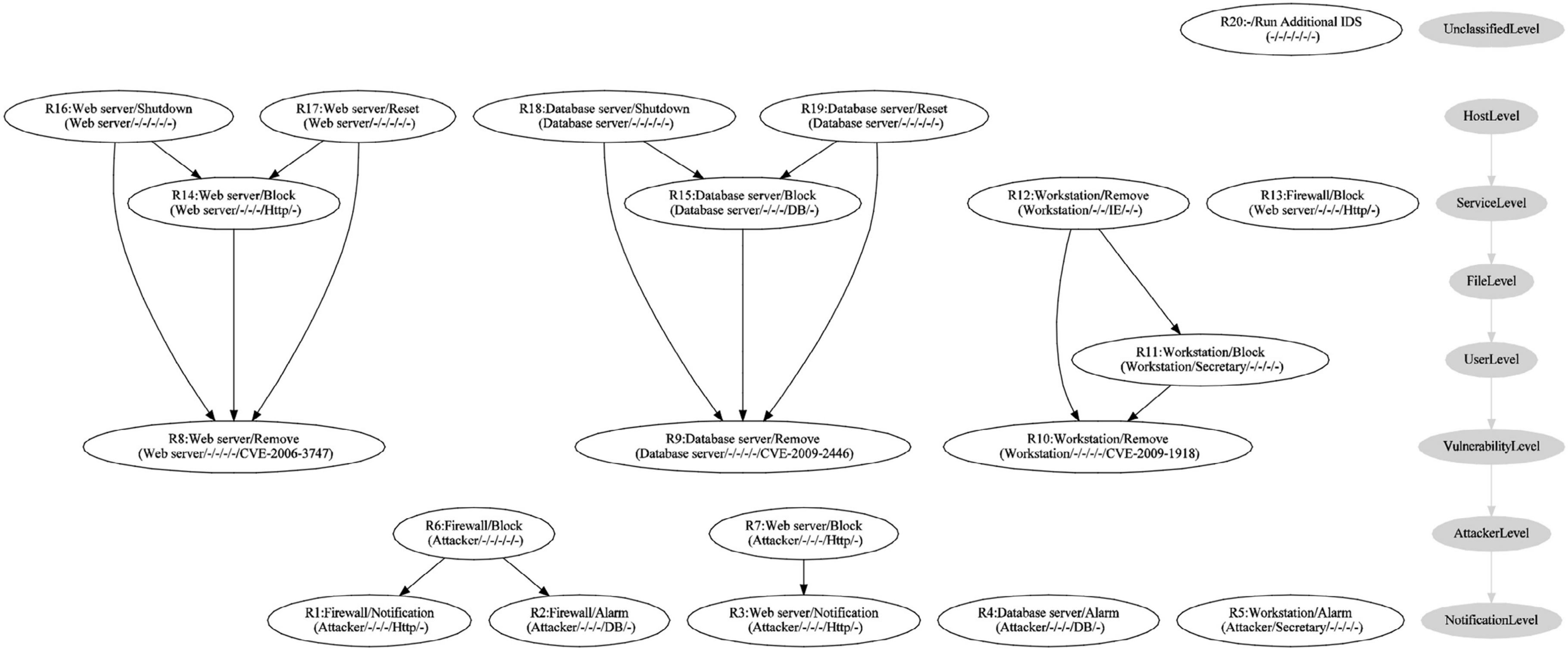


Fig. 8 – Response graph for the mentioned network.

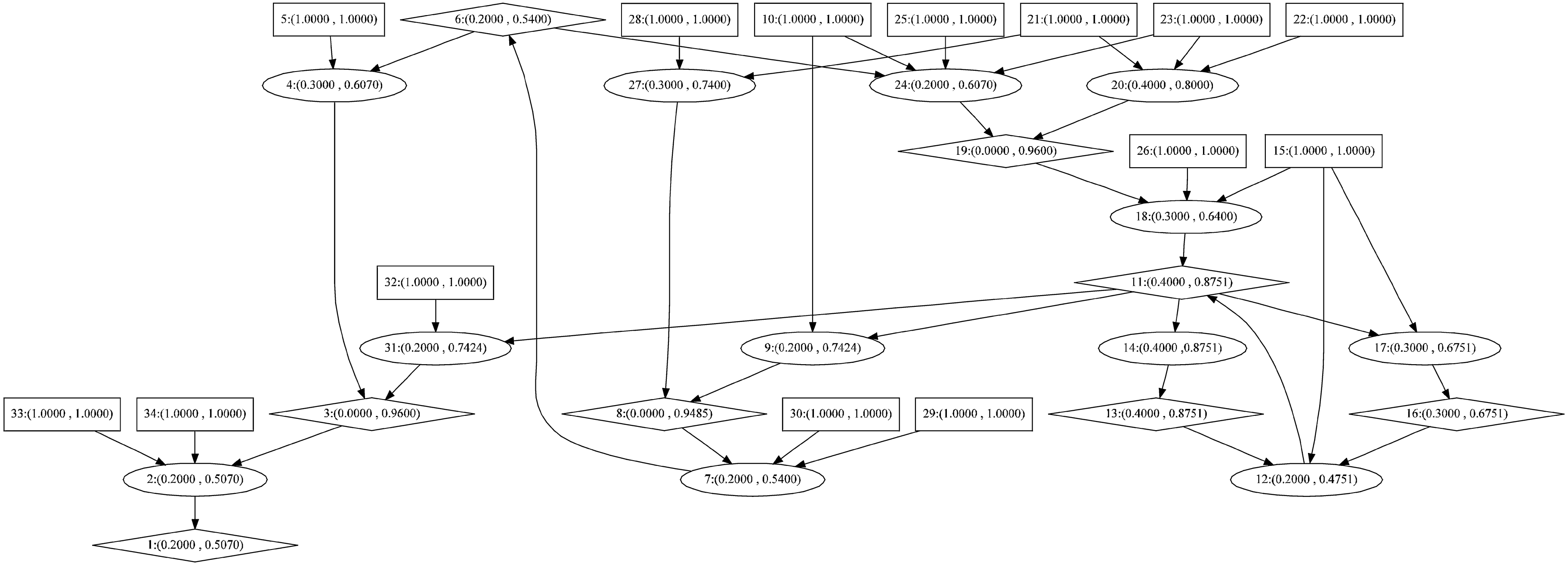


Fig. 9 – The obtained uncertainty-aware attack graph for the mentioned network.

Table 3 – Defined responses for the example network.

Response-level	Response number	Response-location	Response-action	Response-target	User	File	Process	Service	Vulnerability	Operational cost
Notification-level	R1	Firewall	Notification	Attacker	–	–	–	Http	–	0.1
	R2	Firewall	Alarm	Attacker	–	–	–	DB	–	0.1
	R3	Web server	Notification	Attacker	–	–	–	Http	–	0.1
	R4	Database server	Alarm	Attacker	–	–	–	DB	–	0.1
	R5	Workstation	Alarm	Attacker	Secretary	–	–	–	–	0.1
Attacker-level	R6	Firewall	Block	Attacker	–	–	–	–	–	0.2
	R7	Web server	Block	Attacker	–	–	–	Http	–	0.2
Vulnerability-level	R8	Web server	Remove	Web server	–	–	–	–	CVE-2006-3747	0.4
	R9	Database server	Remove	Database server	–	–	–	–	CVE-2009-2446	0.4
	R10	Workstation	Remove	Workstation	–	–	–	–	CVE-2009-1918	0.3
User-level	R11	Workstation	Block	Workstation	Secretary	–	–	–	–	0.4
Service-level	R12	Workstation	Remove	Workstation	–	–	IE	–	–	0.2
	R13	Firewall	Block	Web server	–	–	–	Http	–	0.2
	R14	Web server	Block	Web server	–	–	–	Http	–	0.2
	R15	Database server	Block	Database server	–	–	–	DB	–	0.2
Host-level	R16	Web server	Shutdown	Web server	–	–	–	–	–	0.1
	R17	Web server	Reset	Web server	–	–	–	–	–	0.1
	R18	Database server	Shutdown	Database server	–	–	–	–	–	0.1
	R19	Database server	Reset	Database server	–	–	–	–	–	0.1
Unclassified-level	R20	–	Run additional IDS	–	–	–	–	–	–	0.5

Table 5 – The lower and upper probabilities of potential attacks for the mentioned example.

Attack number	Attack name	Lower probability	Upper probability	Number of attack path
Att1	execCode(dbServer,root)	0.2000	0.5070	8
Att2	execCode(webServer,apache)	0.2000	0.5400	4
Att3	execCode(workStation,normalAccount)	0.4000	0.8751	4

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 6 – Extracted attack paths from uncertainty-aware attack graph shown in Fig. 9.

	Nodes in the path
Path1	1 2 3 33 34 4 5 6 7 8 29 30 9 10 11 12 13 15 16 14 17
Path2	1 2 3 33 34 4 5 6 7 8 29 30 9 10 11 18 19 15 26 20 21 22 23
Path3	1 2 3 33 34 4 5 6 7 8 29 30 9 10 11 18 19 15 26 24
Path4	1 2 3 33 34 4 5 6 7 8 29 30 27 28 21
Path5	1 2 3 33 34 31 32 11 12 13 15 16 14 17
Path6	1 2 3 33 34 31 32 11 18 19 15 26 20 21 22 23
Path7	1 2 3 33 34 31 32 11 18 19 15 26 24 6 10 25 23 7 8 29 30 9
Path8	1 2 3 33 34 31 32 11 18 19 15 26 24 6 10 25 23 7 8 29 30 27 28 21
Path9	11 12 13 15 16 14 17
Path10	11 18 19 15 26 20 21 22 23
Path11	11 18 19 15 26 24 6 10 25 23 7 8 29 30 9
Path12	11 18 19 15 26 24 6 10 25 23 7 8 29 30 27 28 21
Path13	6 7 8 29 30 9 10 11 12 13 15 16 14 17
Path14	6 7 8 29 30 9 10 11 18 19 15 26 20 21 22 23
Path15	6 7 8 29 30 9 10 11 18 19 15 26 24
Path16	6 7 8 29 30 27 28 21

Table 7 – Total impact of responses on *GNDG* nodes.

			R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20
S1	C	Min	0	0.1000	0	0.1000	0	0.0300	0.0300	0	0	0	0	0	0	0	0.1000	0	0	0.1000	0.1000	0.0300
		Max	0	0.1000	0	0.1000	0	0.0700	0.0700	0	0	0	0	0	0	0	0.3000	0	0	0.1000	0.1000	0.1000
	I	Min	0	0.1000	0	0.1000	0	0.0300	0.0300	0	0	0	0	0	0	0	0.1000	0	0	0.1000	0.1000	0.0300
		Max	0	0.1000	0	0.1000	0	0.0700	0.0700	0	0	0	0	0	0	0	0.3000	0	0	0.1000	0.1000	0.1000
	A	Min	0	0.0100	0	0.0100	0	0.0100	0.0100	0	0	0	0	0	0	0	0.7000	0	0	1	1	0.0300
		Max	0	0.0100	0	0.0100	0	0.0300	0.0300	0	0	0	0	0	0	0	1	0	0	1	1	0.1000
S2	C	Min	0.0167	0.1000	0.0167	0.1000	0	0.0350	0.0350	0	0	0	0	0	0.0167	0.0167	0.1000	0.0167	0.0167	0.1000	0.1000	0.0350
		Max	0.0167	0.1000	0.0167	0.1000	0	0.0817	0.0817	0	0	0	0	0	0.0500	0.0500	0.3000	0.0167	0.0167	0.1000	0.1000	0.1167
	I	Min	0.0167	0.1000	0.0167	0.1000	0	0.0350	0.0350	0	0	0	0	0	0.0167	0.0167	0.1000	0.0167	0.0167	0.1000	0.1000	0.0350
		Max	0.0167	0.1000	0.0167	0.1000	0	0.0817	0.0817	0	0	0	0	0	0.0500	0.0500	0.3000	0.0167	0.0167	0.1000	0.1000	0.1167
	A	Min	0.0017	0.0100	0.0017	0.0100	0	0.0117	0.0117	0	0	0	0	0	0.1167	0.1167	0.7000	0.1667	0.1667	1	1	0.0350
		Max	0.0017	0.0100	0.0017	0.0100	0	0.0350	0.0350	0	0	0	0	0	0.1667	0.1667	1	0.1667	0.1667	1	1	0.1167
S3	C	Min	0.1000	0	0.1000	0	0	0.0300	0.0300	0	0	0	0	0	0.1000	0.1000	0	0.1000	0.1000	0	0	0.0300
		Max	0.1000	0	0.1000	0	0	0.0700	0.0700	0	0	0	0	0	0.3000	0.3000	0	0.1000	0.1000	0	0	0.1000
	I	Min	0.1000	0	0.1000	0	0	0.0300	0.0300	0	0	0	0	0	0.1000	0.1000	0	0.1000	0.1000	0	0	0.0300
		Max	0.1000	0	0.1000	0	0	0.0700	0.0700	0	0	0	0	0	0.3000	0.3000	0	0.1000	0.1000	0	0	0.1000
	A	Min	0.0100	0	0.0100	0	0	0.0100	0.0100	0	0	0	0	0	0.7000	0.7000	0	1	1	0	0	0.0300
		Max	0.0100	0	0.0100	0	0	0.0300	0.0300	0	0	0	0	0	1	1	0	1	1	0	0	0.1000
S4	C	Min	0.1000	0	0.1000	0	0	0.0300	0.0300	0	0	0	0	0	0.1000	0.1000	0	0.1000	0.1000	0	0	0.0300
		Max	0.1000	0	0.1000	0	0	0.0700	0.0700	0	0	0	0	0	0.3000	0.3000	0	0.1000	0.1000	0	0	0.1000
	I	Min	0.1000	0	0.1000	0	0	0.0300	0.0300	0	0	0	0	0	0.1000	0.1000	0	0.1000	0.1000	0	0	0.0300
		Max	0.1000	0	0.1000	0	0	0.0700	0.0700	0	0	0	0	0	0.3000	0.3000	0	0.1000	0.1000	0	0	0.1000
	A	Min	0.0100	0	0.0100	0	0	0.0100	0.0100	0	0	0	0	0	0.7000	0.7000	0	1	1	0	0	0.0300
		Max	0.0100	0	0.0100	0	0	0.0300	0.0300	0	0	0	0	0	1	1	0	1	1	0	0	0.1000
S5	C	Min	0.0167	0	0.0167	0	0.1000	0.0350	0.0350	0	0	0	0.1000	0	0.0167	0.0167	0	0.0167	0.0167	0	0	0.0350
		Max	0.0167	0	0.0167	0	0.1000	0.0817	0.0817	0	0	0	0.3000	0	0.0500	0.0500	0	0.0167	0.0167	0	0	0.1167
	I	Min	0.0167	0	0.0167	0	0.1000	0.0350	0.0350	0	0	0	0.1000	0	0.0167	0.0167	0	0.0167	0.0167	0	0	0.0350
		Max	0.0167	0	0.0167	0	0.1000	0.0817	0.0817	0	0	0	0.3000	0	0.0500	0.0500	0	0.0167	0.0167	0	0	0.1167
	A	Min	0.0017	0	0.0017	0	0.0100	0.0117	0.0117	0	0	0	0.7000	0	0.1167	0.1167	0	0.1667	0.1667	0	0	0.0350
		Max	0.0017	0	0.0017	0	0.0100	0.0350	0.0350	0	0	0	1	0	0.1667	0.1667	0	0.1667	0.1667	0	0	0.1167
S6	C	Min	0	0	0	0	0	0.0300	0.0300	0	0	0	0	0.1000	0	0	0	0	0	0	0	0.0300
		Max	0	0	0	0	0	0.0700	0.0700	0	0	0	0	0.3000	0	0	0	0	0	0	0	0.1000
	I	Min	0	0	0	0	0	0.0300	0.0300	0	0	0	0	0.1000	0	0	0	0	0	0	0	0.0300
		Max	0	0	0	0	0	0.0700	0.0700	0	0	0	0	0.3000	0	0	0	0	0	0	0	0.1000
	A	Min	0	0	0	0	0.0017	0.0117	0.0117	0	0	0	0.1167	0.7000	0	0	0	0	0	0	0	0.0350
		Max	0	0	0	0	0.0017	0.0350	0.0350	0	0	0	0.1667	1	0	0	0	0	0	0	0	0.1167

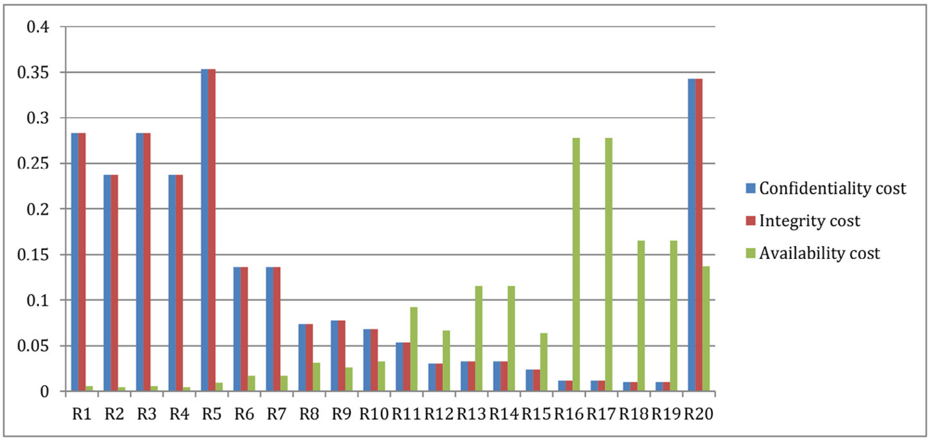


Fig. 10 – The average of response cost on each CIA parameter for the mentioned example.

Table 8 – Total damage of attacks on *GNDG* nodes.

			Att1	Att2	Att3
S1	C	Min	0	0	0
		Max	0.2000	0	0
	I	Min	0	0	0
		Max	0.2000	0	0
	A	Min	0	0	0
		Max	0.2000	0	0
S2	C	Min	0.5000	0.0833	0
		Max	0.9000	0.1500	0
	I	Min	0.5000	0.0833	0
		Max	0.9000	0.1500	0
	A	Min	0.5000	0.0833	0
		Max	0.9000	0.1500	0
S3	C	Min	0	0	0
		Max	0	0.2000	0
	I	Min	0	0	0
		Max	0	0.2000	0
	A	Min	0	0	0
		Max	0	0.2000	0
S4	C	Min	0	0.6000	0
		Max	0	1.0000	0
	I	Min	0	0.6000	0
		Max	0	1.0000	0
	A	Min	0	0.6000	0
		Max	0	1.0000	0
S5	C	Min	0	0.0833	0.5000
		Max	0	0.1500	0.9000
	I	Min	0	0.0833	0.5000
		Max	0	0.1500	0.9000
	A	Min	0	0	0
		Max	0	0.0333	0.2000
S6	C	Min	0	0	0
		Max	0	0	0.2000
	I	Min	0	0	0
		Max	0	0	0.2000
	A	Min	0	0	0
		Max	0	0	0.2000

is not considered in literature is the lack of standardization of 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 9 – The response goodness for three potential attacks in the mentioned example.

	Confidentiality						Integrity						Availability					
	Min			Max			Min			Max			Min			Max		
	Att1	Att2	Att3	Att1	Att2	Att3	Att1	Att2	Att3	Att1	Att2	Att3	Att1	Att2	Att3	Att1	Att2	Att3
R1	0.0101	0.0102	0.0099	0.0196	0.0198	0.0191	0.0101	0.0102	0.0099	0.0196	0.0198	0.0191	0.1931	0.2245	0.2622	0.4147	0.4940	0.5585
R2	0.0045	0	0	0.0089	0	0	0.0045	0	0	0.0089	0	0	0.0988	0	0	0.2072	0	0
R3	0.0101	0.0102	0.0099	0.0196	0.0198	0.0191	0.0101	0.0102	0.0099	0.0196	0.0198	0.0191	0.1931	0.2245	0.2622	0.4147	0.4940	0.5585
R4	0.0045	0	0	0.0089	0	0	0.0045	0	0	0.0089	0	0	0.0988	0	0	0.2072	0	0
R5	0.0064	0.0070	0.0055	0.0125	0.0137	0.0107	0.0064	0.0070	0.0055	0.0125	0.0137	0.0107	0.1092	0.1259	0.1202	0.2458	0.3031	0.2777
R6	0.0349	0.0304	0.0265	0.1592	0.1388	0.1200	0.0349	0.0304	0.0265	0.1592	0.1388	0.1200	0.1531	0.1465	0.1509	1	1	1
R7	0.0349	0.0304	0.0265	0.1592	0.1388	0.1200	0.0349	0.0304	0.0265	0.1592	0.1388	0.1200	0.1531	0.1465	0.1509	1	1	1
R8	0.0557	0.0545	0.0555	0.3278	0.3214	0.3272	0.0557	0.0545	0.0555	0.3278	0.3214	0.3272	0.1073	0.1219	0.1481	0.6977	0.8131	0.9647
R9	0.0222	0	0	0.1373	0	0	0.0222	0	0	0.1373	0	0	0.0493	0	0	0.3221	0	0
R10	0.0562	0.0649	0.0492	0.3340	0.3900	0.2922	0.0562	0.0649	0.0492	0.3340	0.3900	0.2922	0.0964	0.1191	0.1088	0.6617	0.8825	0.7683
R11	0.0901	0.1098	0.0802	0.5290	0.6456	0.4696	0.0901	0.1098	0.0802	0.5290	0.6456	0.4696	0.0442	0.0561	0.0500	0.1449	0.1979	0.1690
R12	0.0858	0.1009	0.0750	0.5082	0.6035	0.4440	0.0858	0.1009	0.0750	0.5082	0.6035	0.4440	0.0442	0.0561	0.0500	0.1439	0.1957	0.1671
R13	0.1719	0.1724	0.1734	1	1	1	0.1719	0.1724	0.1734	1	1	1	0.1005	0.1182	0.1413	0.3054	0.3648	0.4240
R14	0.1719	0.1724	0.1734	1	1	1	0.1719	0.1724	0.1734	1	1	1	0.1005	0.1182	0.1413	0.3054	0.3648	0.4240
R15	0.0870	0	0	0.5185	0	0	0.0870	0	0	0.5185	0	0	0.0575	0	0	0.1724	0	0
R16	0.2518	0.2391	0.2298	0.4890	0.4632	0.4423	0.2518	0.2391	0.2298	0.4890	0.4632	0.4423	0.0494	0.0548	0.0618	0.1051	0.1186	0.1304
R17	0.2518	0.2391	0.2298	0.4890	0.4632	0.4423	0.2518	0.2391	0.2298	0.4890	0.4632	0.4423	0.0494	0.0548	0.0618	0.1051	0.1186	0.1304
R18	0.1045	0	0	0.2074	0	0	0.1045	0	0	0.2074	0	0	0.0230	0	0	0.0483	0	0
R19	0.1045	0	0	0.2074	0	0	0.1045	0	0	0.2074	0	0	0.0230	0	0	0.0483	0	0
R20	0.0244	0.0213	0.0185	0.1592	0.1388	0.1200	0.0244	0.0213	0.0185	0.1592	0.1388	0.1200	0.0459	0.0440	0.0453	0.3333	0.3333	0.3333

Table 10 – Response cost on CIA parameters for the mentioned example.

	Confidentiality cost		Integrity cost		Availability cost	
	Min	Max	Min	Max	Min	Max
R1	0.0587	0.7360	0.0587	0.7360	5.1672e–04	0.0166
R2	0.0293	0.6391	0.0293	0.6391	1.8714e–04	0.0137
R3	0.0587	0.7360	0.0587	0.7360	5.1672e–04	0.0166
R4	0.0293	0.6391	0.0293	0.6391	1.8714e–04	0.0137
R5	0.0639	1	0.0639	1	0	0.0324
R6	0.0177	0.4043	0.0177	0.4043	0.0047	0.0476
R7	0.0177	0.4043	0.0177	0.4043	0.0047	0.0476
R8	0.0136	0.1916	0.0136	0.1916	0.0133	0.0712
R9	0.0108	0.1982	0.0108	0.1982	0.0069	0.0629
R10	0.0136	0.1890	0.0136	0.1890	0.0160	0.0845
R11	0.0069	0.1494	0.0069	0.1494	0.0483	0.2292
R12	0.0030	0.0887	0.0030	0.0887	0.0245	0.1809
R13	0.0026	0.0904	0.0026	0.0904	0.0574	0.2457
R14	0.0026	0.0904	0.0026	0.0904	0.0574	0.2457
R15	0.0015	0.0664	0.0015	0.0664	0.0203	0.1558
R16	0.0024	0.0299	0.0024	0.0299	0.2117	0.4872
R17	0.0024	0.0299	0.0024	0.0299	0.2117	0.4872
R18	0.0013	0.0272	0.0013	0.0272	0.0804	0.3618
R19	0.0013	0.0272	0.0013	0.0272	0.0804	0.3618
R20	0.0131	0.2506	0.0131	0.2506	0.0138	0.0912

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(webServer)”,”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)

28,”hacl(internet,webServer,httpProtocol,httpPort)”,”LEAF”,(1.0000 - 1.0000)

29,”networkServiceInfo(webServer,httpd,httpProtocol,httpPort,apache)”,”LEAF”,(1.0000 - 1.0000)

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. $H\_similarity(ha_x, n_i)$

```
Input: $UAG$ , $MRG$
Output: $H\_similarity(ha_x, n_i)$ ( Similarity of $UAG$ node $n_i$ with hyper-alerts $ha_x$ )
Initialization: $n_{AND} = \{n_i \in N \mid < n_i, d_i > \in D,\ d_i = AND\}$, $n_{OR} = \{n_i \in N \mid < n_i, d_i > \in D,\ d_i = OR\}$, $n_{LEAF} = \{n_i \in N \mid < n_i, d_i > \in D,\ d_i = LEAF\}$,
$\xi \approx 0$

Extract SIP, DIP, Host, Client, Server, Protocol, Port, vulID from $n_i$
if $(n_i \in n_{LEAF})$ then
    if $(n_i.Host \in ha_x.DIP) \& (n_i.vulID \in ha_x.vulID)$ then
        $H\_similarity(ha_x, n_i) = 1$
    end if
    if $(n_i.SIP \in ha_x.SIP) \& (n_i.DIP \in ha_x.DIP) \& (n_i.\mathrm{Protocol} \in ha_x.Pro) \& (n_i.\mathrm{Port} \in ha_x.Dp)$ then
        $H\_similarity(ha_x, n_i) = 1$
    end if
    if$(n_i.\{Host \mid SIP \mid DIP \mid Client \mid Server\} \in ha_x.\{SIP \mid DIP\})$ then
        $H\_similarity(ha_x, n_i) = H\_similarity(ha_x, n_i) + \xi$
    end if
    if$(n_i.\mathrm{Protocol} \in ha_x.Pro)$ then
        $H\_similarity(ha_x, n_i) = H\_similarity(ha_x, n_i) + \xi$
    end if
    if$(n_i.\mathrm{Port} \in ha_x.\{Sp \mid Dp\})$ then
        $H\_similarity(ha_x, n_i) = H\_similarity(ha_x, n_i) + \xi$
    end if
    if $(n_i.vulID \in ha_x.vulID)$ then
        $H\_similarity(ha_x, n_i) = H\_similarity(ha_x, n_i) + \xi$
    end if
end if
if $(n_i \in n_{OR})$ then
        $Hsimilarity(ha_x, n_i) = \max_k (Hsimilarity(ha_x, Parent_k(n_i)))$
end if
if $(n_i \in n_{AND})$ then
        $Hsimilarity(ha_x, n_i) = \prod_k (Hsimilarity(ha_x, Parent_k(n_i)))$
end if
```

REFERENCES

Balepin I, Maltsev S, Rowe J, Levitt K. Using specification-based intrusion detection for automated response, in The 6th international symposium on recent advances in intrusion detection, Pittsburgh, PA, USA, 2003.

Bowen T, Chee D, Segal M, Sekar R, Shanbhag T, Uppuluri P. Building survivable systems: An integrated approach based on intrusion detection and damage containment, in DARPA information survivability conference and exposition (DISCEX I), Hilton Head, SC, 2000, pp. 84–99.

Bray T, Paoli J, Sperberg-McQueen CM. Extensible markup language (XML), World Wide Web consortium recommendation REC-xml-19980210, vol. 16, <http://www.w3.org/TR/1998/REC-xml-19980210>; 1998 [accessed 2015].

Carver CA, Hill JMD, Surdu JR, Pooch UW. A methodology for using intelligent agents to provide automated intrusion response, in The 2000 IEEE workshop on information assurance and security, United States Military Academy, West Point, NY, 2000, pp. 110–16.

Curtis A, Carver J. Adaptive agent-based intrusion response, Texas A&M University, 2001.

CVE, Common vulnerability and exposures, <http://www.cve.mitre.org/about>; 2015 [accessed 2015].

De Campos LM, Fernández-Luna JM, Huete JF, Martín-Dancausa C. Managing structured queries in probabilistic XML retrieval systems. Inf Proc Manage 2010;46(5):514–32.

Debar H, Curry D, Feinstein B. The intrusion detection message exchange format (IDMEF), 2007.

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

Foo B, Wu Y-S, Mao Y-C, Bagchi S, Spafford E. ADEPTS: Adaptive intrusion response using attack graphs in an e-commerce environment, in The 2005 international conference on dependable systems and networks, Yokohama, Japan, 2005, pp. 508–17.

GhasemiGol M, Ghaemi-Bafghi A. E-correlator: an entropy-based alert correlation system. Secur Commun Netw 2015;8(5):822–36.

GhasemiGol M, Ghaemi-Bafghi A, Takabi H. A comprehensive approach for network attack forecasting. Comput Secur 2016;58:83–105.

Haslum K, Abraham A, Knapskog S. DIPS : A framework for distributed intrusion prediction and prevention using hidden markov models and online fuzzy risk assessment, in the 3rd international symposium on information assurance and security, Manchester, United Kingdom, 2007, pp. 183–8.

Jahnke M, Thul C, Martini P. Graph-based Metrics for Intrusion Response Measures in Computer Networks, in The 3rd LCN workshop on network Security. Held in conjunction with the 32nd IEEE conference on local computer networks (LCN), Dublin, Ireland, 2007, pp. 1035–42.

Kanoun W, Cuppens-Boulahia N, Cuppens F, Dubus S. Risk-Aware Framework for Activating and Deactivating Policy-Based Response, in The fourth international conference on network and system security, Melbourne, VIC, 2010, pp. 207–15.

Kanoun W, Samarji L, Cuppens-Boulahia N, Dubus S, Cuppens F. Towards a Temporal Response Taxonomy, in 7th international workshop, DPM 2012, and 5th international workshop, SETOP 2012, Pisa, Italy, 2013.

Kheir N, Cuppens-Boulahia N, Cuppens F, Debar H. A service dependency model for cost sensitive intrusion response, in The 15th European conference on research in computer security, Athens, Greece, 2010, pp. 626–42.

Lee W, Fan W, Miller M, Stolfo SJ, Zadok E. Toward cost-sensitive modeling for intrusion detection and response. J Comput Secur 2002;10:5–22.

Lewandowski SM, Hook DJV, O'Leary GC, Haines JW, Rossey LM. SARA: Survivable autonomic response architecture, in DARPA information survivability conference and exposition II, Anaheim, CA, 2001, pp. 77–88.

Locasto ME, Wang K, Keromytis AD, Stolfo SJ. FLIPS: Hybrid adaptive intrution prevention, in Recent advances in intrusion detection (RAID), Seattle, WA, USA, 2005, pp. 82–101.

Mateos V, Villagrá VA, Romero F, Berrocal J. Definition of response metrics for an ontology-based automated intrusion response systems. Comput Elect Eng 2012;38:1102–14.

Megiddo N. Linear programming in linear time when the dimension is fixed. J ACM 1984;31(1):114–27.

Mu C, Li Y. An intrusion response decision-making model based on hierarchical task network planning. Expert Syst Appl 2010;37(3):2465–72.

Mu C, Shuai B, Liu H. Analysis of Response Factors in Intrusion Response Decision-Making, in Third international joint conference on computational science and optimization (CSO), Huangshan, Anhui, China, 2010, pp. 395–9.

Musman S, Flesher P. System or security managers adaptive response tool, in DARPA information survivability conference and exposition II, Hilton Head, SC, 2000, pp. 56–68.

Ou X, Govindavajhala S, Appel AW. MulVAL: A Logic-based Network Security Analyzer, in USENIX security, 2005.

Papadaki M, Furnell SM. Achieving automated intrusion response: a prototype implementation. Inf Manage Comput Secur 2006;14(3):235–51.

Porras P, Neumann P. EMERALD: Event monitoring enabling response to anomalous live disturbances, in The 20th national information systems security conference, Baltimore, MD, 1997, pp. 353–65.

Ragsdale DJ, Carver CA, Humphries JW, Pooch UW. Adaptation techniques for intrusion detection and intrusion response systems, in 2000 IEEE international conference on systems, man, and cybernetics, Nashville, TN, 2000, pp. 2344–9.

Schnackenberg D, Holliday H, Smith R, Djahandari K, Sterne D. Cooperative intrusion traceback and response architecture (CITRA), in DARPA information survivability conference and exposition II, Anaheim, CA, 2001, pp. 56–68.

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

Shameli-Sendi A, Dagenais M. ORCEF: online response cost evaluation framework for intrusion response system. J Netw Comput Appl 2015;55:89–107.

Shameli-Sendi A, Cheriet M, Hamou-Lhadj A. Taxonomy of intrusion risk assessment and response system. Comput Secur 2014;45:1–16.

Somayaji A, Forrest S. Automated response using system-call delays, in The 9th USENIX security symposium, Denver, Colorado, 2000.

Stakhanova N, Basu S, Wong J. A taxonomy of intrusion response systems. Int J Inf Comput Secur 2007a;1(1/2):169–84.

Stakhanova N, Basu S, Wong J. A cost-sensitive model for preemptive intrusion response systems, in 21st international conference on advanced networking and applications, Niagara Falls, ON, Canada, 2007b, pp. 428–35.

Stakhanova N, Strasburg C, Basu S, Wong JS. Towards cost-sensitive assessment of intrusion response selection. J Comput Secur 2012;20.

Strasburg C, Stakhanova N, Basu S, Wong JS. Intrusion response cost assessment methodology, in Proceedings of the 4th International symposium on information, computer, and communications security, 2009a, pp. 388–91.

Strasburg C, Stakhanova N, Basu S, Wong JS. A Framework for Cost Sensitive Assessment of Intrusion Response Selection, in 33rd annual IEEE international computer software and applications conference, Seattle, WA, 2009b, pp. 355–60.

Tanachaiwiwat S, Hwang K, Chen Y. Adaptive intrusion response to minimize risk over multiple network attacks. ACM Trans Inform Syst Secur 2002;19:1–30.

Toth T, Kruegel C. Evaluating the impact of automated intrusion response mechanisms, in The 18th annual computer security applications conference, Las Vegas, Nevada, 2002, pp. 301–10.

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.

Wang HQ, Wang GF, Lan Y, Wang K, Liu D. A new automatic intrusion response taxonomy and its application, in The 8th Asia-Pacific web conference and workshops (APWeb 2006), Harbin, People R China, 2006, pp. 999–1003.

Wang X, Reeves DS, Wu SF. Tracing based active intrusion response. J Inf Warefare 2001;1(1):50–61.

White G, Fisch E, Pooch U. Cooperating security managers: a peer-based intrusion detection system. IEEE Netw 1996;10:20–3.

Zhang Z, Ho P-H, He L. Measuring IDS-estimated attack impacts for rational incident response: a decision theoretic approach. Comput Secur 2009;28:605–14.

Mohammad GhasemiGol is a PhD candidate in computer engineering at Ferdowsi University of Mashhad (FUM). He will join the Department of Computer Engineering at the University of Birjand in fall 2016. He received the B.S. degree in Computer Engineering from Payame Noor University (PNU), Birjand, Iran, in 2006. He also received the M.S. degree in Computer Engineering at FUM, Iran, in 2009. November 2014 to July 2015, he was with the Department of Computer Science and Engineering, University of North Texas, Denton, TX, USA as a visiting research scholar. His research interests include network security, intrusion detection and response systems, alert management, data mining, and optimization problems.

Hassan Takabi is Assistant Professor of Computer Science and Engineering at the University of North Texas, Denton, TX, USA. He is director and founder of the Information Security and Privacy: Interdisciplinary Research and Education (INSPIRE) Lab and a member of the Center for Information and Computer Security (CICS). His research is focused on various aspects of cybersecurity and privacy 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 1997and 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.