Available online at www.sciencedirect.com

ScienceDirect

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

Computers & Security

Taxonomy of intrusion risk assessment and response system

Alireza Shameli-Sendi [a,*], Mohamed Cheriet [a], Abdelwahab Hamou-Lhadj [b]

[a] Department of Electrical and Computer Engineering, Ecole de Technologie Superieure (ETS), Montreal, Canada
[b] Department of Electrical and Computer Engineering, Concordia University, Montreal, Canada

ARTICLE INFO

Article history:
Received 14 December 2013
Received in revised form
30 March 2014
Accepted 27 April 2014
Available online 9 May 2014

Keywords:
Intrusion detection system
Intrusion response system
Intrusion risk assessment
Response time
Prediction
Response cost
Attack graph
Service dependency graph

ABSTRACT

In recent years, we have seen notable changes in the way attackers infiltrate computer systems compromising their functionality. Research in intrusion detection systems aims to reduce the impact of these attacks. In this paper, we present a taxonomy of Intrusion Response Systems (IRS) and Intrusion Risk Assessment (IRA), two important components of an intrusion detection solution. We achieve this by classifying a number of studies published during the last two decades. We discuss the key features of existing IRS and IRA. We show how characterizing security risks and choosing the right countermeasures are an important and challenging part of designing an IRS and an IRA. Poorly designed IRS and IRA may reduce network performance and wrongly disconnect users from a network. We propose techniques on how to address these challenges and highlight the need for a comprehensive defense mechanism approach. We believe that this taxonomy will open up interesting areas for future research in the growing field of intrusion risk assessment and response systems.

1. Introduction

Today's society relies increasingly on network services to manage its critical operations in a variety of domains including health, finances, public safety, telecommunication, and so on. It is therefore important to maintain high-availability and adequate response time of these services at all time. This is threatened by the presence of hostile attackers that look for ways to gain access to systems and infect computers (Zhou et al., 2010). To mitigate these threats, the deployment of an appropriate defense mechanism is needed. As Fig. 1 illustrates, the defense life-cycle includes four phases: *Prevention*, *Monitoring*, *Detection*, and *Mitigation*. The prevention phase ensures that appropriate safeguards are placed in different locations to secure services and data. In the monitoring phase, monitoring tools are deployed to gather useful host or network information to follow the execution of the system. The detection phase is where an Intrusion Detection System (IDS) analyzes the running systems, looking for deviations from a pre-established normal behavior.

IDSs vary depending on whether they monitor network traffic (Network-based IDS) or local hosts (Host-based IDS) (Scarfone and Mell, 2007; Stein et al., 2005; Anuar et al., 2008; Lazarevic et al., 2003; Xiao et al., 2010). IDSs are divided into two categories: *anomaly-based* and *signature-based*. Anomaly-

* *Corresponding author.*
E-mail addresses: alireza.shameli@synchromedia.ca, alireza.shameli-sendi@polymtl.ca (A. Shameli-Sendi), mohamed.cheriet@etsmtl.ca (M. Cheriet), wahab.hamou-lhadj@concordia.ca (A. Hamou-Lhadj).
http://dx.doi.org/10.1016/j.cose.2014.04.009
0167-4048/

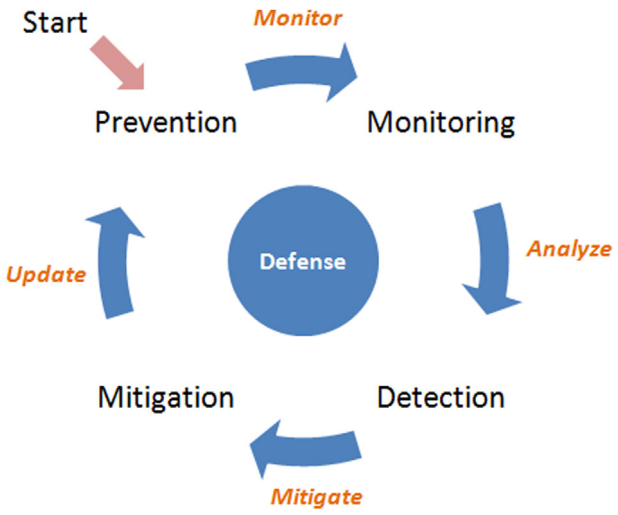

Fig. 1 – Defense life-cycle.

based techniques rely a two-step process. The first step, the training phase, a classifier is built using a machine learning algorithm, such as a decision trees, Bayesian Network, a Neural Network, etc. (Berkhin, 2001; Adetunmbi et al., 2008; Han and Kamber, 2006). The second step, the testing phase, tests the detection accuracy (by measuring true positive and false positive rates). The anomaly-based detection approach is able to detect unknown attack patterns and does not need predefined signatures. However, it suffers from the problem of characterizing the normal behavior. Signature-based techniques (also known as misuse detection) (The Snort Project, 2009), on the other hand, rely on known patterns (signatures) of attacks. Pattern matching makes this technique deterministic, which means that it can be customized for various systems, although it is difficult to find the right balance between accuracy and generality, which may lead to false negatives and false positives (Difference between Signature Based and Anomaly Based Detection in IDS; Yusof, 2009).

The last phase, mitigation, complements the defense life-cycle by evaluating the severity of attacks and selecting a correct response at the right time. In the mitigation phase, an Intrusion Response System (IRS) is responsible for selecting appropriate countermeasures to effectively handle malicious or unauthorized activities.

An IRS has to assess the value of the loss incurred by a compromised resource (Gehani and Kedem, 2004). It also has to have an accurate evaluation of the cost of the response (Strasburg et al., 2009; Stakhanova et al., 2007a). Otherwise, an automated IRS may reduce network performance, or wrongly disconnect valid users from the network. Moreover, a badly designed IRS may result in high costs associated with reestablishing the services. This incurred overhead often pushes the administrators to simply disable the IRS.

Designing an IRS poses several challenges. First, the chain of vulnerabilities exploited by an attacker can link services on either a single machine or those on different machines (Ammann et al., 2002; Jha et al., 2002). The complexity of the attack makes it a challenge to accurately calculate the risk impact. Then, there are the many decisions that an IRS needs to make, which can be summarized in the following questions:

- Is the attack harmful enough to warrant repelling?
- What is the value (importance) of the compromised target?
- Which set of responses is appropriate for repelling the attack?

Intrusion Risk Assessment (IRA) is the process of identifying and characterizing risks. The result of risk assessment helps minimize the cost of applying all available sets of responses. It may be enough in some situation to only apply a subset of available responses (Jahnke et al., 2007; Kanoun et al., 2008). That is said, risk assessment helps an IRS determine the probability that a detected anomaly is a valid attack that requires attention (in the form of a response) (Mu et al., 2008).

In this paper, we classify existing IRS and IRA design approaches. The goal is to identify the strengths and weaknesses of existing approaches. We also propose guidelines for improving IRS and IRA.

The rest of this paper is organized as follows: in Section 2, we propose our taxonomy of intrusion response and risk assessment and describe their main elements. A review of recent existing IRS and IRA is presented in Section 3. Section 4, we discuss the current state of the intrusion response and risk assessment, and suggestions for future research which can improve the current weaknesses of IRS. Finally, in Section 5, we present our conclusions.

2. A taxonomy of intrusion response systems and risk assessment

The criteria we propose for classifying IRS and IRA techniques are discussed in this section. The characteristics of the proposed taxonomy are depicted in Fig. 2. These criteria are based on extensive review of the literature:

- **Level of Automation**: An important feature of an IRS is whether it can be fully automated or requires administrator intervention after each incident.
- **Response Cost**: Knowing the power of responses to attune the response cost with attack cost plays a critical rule in IRS. The evaluation of the positive effects and negative impacts of responses are very important to identify response cost.
- **Response Time**: This criterion refers to whether the response can be applied with some delay or before the attack affects the target.
- **Adjustment Ability**: Usually, an IRS framework is run with a number of pre-estimated responses. It is very important to readjust the strength of the responses depending on the attacks.
- **Response Selection**: The task of an IRS is to choose the best possible response. Existing techniques vary in the way response selection is achieved.

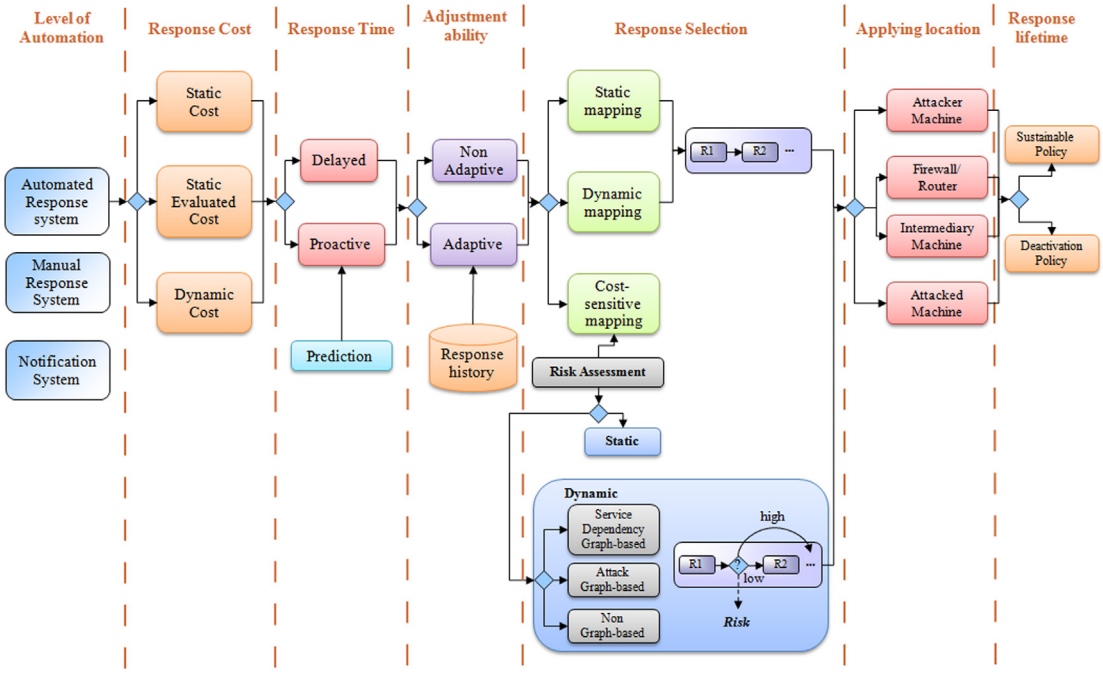

Fig. 2 – Taxonomy of intrusion response systems.

- **Applying Location**: There are different locations in the network to mitigate attacks. The location has different value in terms of online users and service dependencies.
- **Deactivation Ability**: Another distinguishing feature that separates IRSs is response deactivation (response lifetime), which can take into account users needs in terms of quality of service. Most countermeasures are temporary actions which have an intrinsic cost or induce side effects on the monitored system, or both (Kanoun et al., 2010).

2.1. *Level of automation*

Depending on their level of automation, an IRS can be categorized as *notification systems*, *manual response systems*, and *automated response systems*.

2.1.1. *Notification systems*

Notification systems mainly generate alerts when an attack is detected. An alert contains information about the attack including the attack description, time of attack, source IP, destination IP, and user account (Stakhanova et al., 2007b; Ragsdale et al., 2000). The alerts are then used by the administrator to select the applicable reactive measures, if any. This approach is not designed to prevent attacks or to bring back the systems to a safe mode. Its aim is to notify system administrator to select an appropriate response.

2.1.2. *Manual response systems*

In these systems, there are some preconfigured sets of responses based on the type of attacks. A preconfigured set of actions is applied by the administrator when a problem arises. This approach is more highly automated than the notification system approach (Toth and Kregel, 2002; Tanachaiwiwat et al., 2002). The challenge of this approach is the delay between the intrusion and the human response (Stakhanova et al., 2007b; Lee et al., 2002).

2.1.3. *Automated response systems*

Unlike the two previous methods which suffer from delay between intrusion detection and response, automated response systems are designed to be fully automated and no human intervention is required (Curtis and Carver, 2001; White et al., 1996). One of the problems with this approach is the possibility that an inappropriate response will be executed when a problem arises (Mu and Li, 2010). Another challenge with executing an automated response is to ensure that the response is adequate to neutralize the attack.

2.2. *Response cost*

First, we define the term response cost as follows:

Definition 1 (Response Cost). *Response cost is the impact of applying response in our network in terms of continuing network services and users' need. Although the strong response like disabling daemon has strong ability to mitigate attack and protect our network, has very high impact on continuing network service and online users.*

Response cost evaluation is an important part of an IRS. Although many automated IRS have been proposed, most of them use statically evaluated responses, avoiding the need for dynamic evaluation (Shameli-Sendi et al., 2012). However, the

static model has its own drawbacks, which can be overcome using dynamic evaluation models for the responses. Dynamic evaluation will also more effectively protect a system from attack, as threats will be more predictable. Verifying the effect of a response in both dynamic mode and static mode is a challenge. There is a need to specify accurate parameters to evaluate the quality of the response. For example, if we have an Apache process under the control of an attacker, this process is now a gateway for the attacker to access the network. The accepted countermeasure would be to kill this potentially dangerous process. When we apply this response, we will increase our data confidentiality and integrity (C and I of CIA) if the process was doing some damage on our system. The negative impact is that we lose the Apache availability (A of CIA), since the Web server is now dead which causes the user websites to be down. Let us imagine another scenario, where we have a process on a server consuming a considerable amount of CPU resources that is doing nothing but slowing down a machine (a kind of CPU DoS). This time, killing the process will improve service availability (system performance), but will not change anything in terms of data confidentiality and integrity. We now have two very different results for the same response. Also, of the effects of some responses may depend on the network infrastructure. For example, applying a response inside the external DMZ is probably very different from doing so inside the LAN or "secure zone" in terms of CIA. Responses cannot be evaluated without considering the attacks themselves, which are generally divided into the following four categories (Lee et al., 2002; Haslum et al., 2007):

1) **Denial of service (DoS)**: The attacker tries to make resources unavailable to their intended users, or consume resources such as bandwidth, disk space, or processor time. The attacker is not looking to obtain root access, and so there is not much permanent damage.
2) **User to root (U2R)**: An individual user tries to obtain root privileges illegally by exploiting system vulnerabilities. The attacker first gains local access on the target machine, and then exploits system vulnerabilities to perform the transition from user to root level. After acquiring root privileges, the attacker can install backdoor entries for future exploitation and change system files to collect information (Sabhnani and Serpen, 2003).
3) **Remote to local (R2L)**: The attacker tries to gain unauthorized access to a computer from a remote machine by exploiting system vulnerabilities.
4) **Probe**: The attacker scans a network to gather information and detect possible vulnerabilities. This type of attack is very useful, in that it can provide information for the first step of a multi-step attack. Examples are using automated tools such as ipsweep, nmap, portsweep, etc.

In the first category, where the attacker attempts to slow down the system, we are looking for a response that can increase service availability (or performance). In the second and third categories, because the system is under the control of an attacker, we are looking for a response that can increase data confidentiality and integrity. In the fourth category, attackers attempt to gather information about possible vulnerabilities from the network. Thus, responses that improve data confidentiality and service availability are called for. A dynamic response model offers the best response based on the current situation of the network, and so the positive effects and negative impacts of the responses must be evaluated online at the time of the attack. Evaluating the cost of the response in online mode can be based on resource interdependencies, the number of online users, the users privilege level, etc. There are three types of response cost model:

2.2.1. *Static cost model*

The static response cost (R^{s}_{cost}) is obtained by assigning a static value based on an expert's opinion. So, in this approach, a static value is considered for each response ($R^{s}_{cost} = CONSTANT$).

2.2.2. *Static evaluated cost model*

In this approach, a statically evaluated cost, obtained by an evaluation mechanism, is associated with each response ($R^{se}_{cost} = f(x)$). The response cost in the majority of existing models is statically evaluated. A common solution is to evaluate the positive effects (P) of the responses based on their consequences on confidentiality (C), integrity (I), availability (A), and performance (P). To evaluate the negative impacts (N), we can consider the consequences for the other resources in terms of availability ($\neg A$) and performance ($\neg P$) (Strasburg et al., 2009, 2008). For example, after running a response that blocks a specific subnet, a Web server under attack is no longer at risk, but the availability of the service has decreased. After evaluating the positive effect and negative impact of each response, we then calculate the response cost. One solution is as Eq. (1) illustrates (Mu and Li, 2010), obviously the higher RC, the better the response in ordering list:

$$R^{se}_{cost} = \frac{P}{N} = \frac{C + I + A + P}{\neg A + \neg P} \tag{1}$$

2.2.3. *Dynamic evaluated cost model*

The dynamic evaluated cost (R^{de}_{cost}) is based on the network situation. We can evaluate the response cost online based on the dependencies between resources (Jahnke et al., 2007; Kheir et al., 2010) and online users. For example, the effect of terminating a dangerous process depends on the number of other entities (other processes, online users, etc.) that use this process. If the cost of terminating the process is high then perhaps another response should be selected. Evaluating the response cost should take into account the resource dependencies, the number of online users, and the user privilege levels. In other words, we need an accurate cost-sensitive response system.

2.3. *Response time*

In point of response time, IRSs can be classified into type categories: *Delayed* and *Proactive* (Stakhanova et al., 2007b; Anuar et al., 2010). In the delayed mode, the responses are formulate only after an intrusion is detected. Most existing IRS use this approach (e.g., Strasburg et al., 2009; Papadaki and Furnell, 2006) although it is known to be ineffective for maximum security. This is because an attacks can cause

serious harm (stealing confidential information) before an IDS can detect it. This approach has been criticized because of the fact that an attack. Take, for example, the case where an attacker gains access to an unauthorized database. An IDS may detect this intrusion only after the attacker had illegally gained possession of critical information. In such as case, a delayed response would not be useful. Another important limitation of the delayed approach is that it is often difficult (if not impossible) to return the system to a healthy state because of the damages that an attack may cause before it is detected (Anuar et al., 2008). In contrast, the proactive approach aims to control and prevent a malicious activity before it happens. This approach is considered critical for defending hosts and networks against attacks. The proactive IRS needs an intrusion prediction mechanism that usually relies on probability measures (Feng et al., 2009) and it is often hard to guarantee that the prediction result is 100 accurate (Stakhanova et al., 2007b).

2.4. *Adjustment ability*

There are two types of adjustment models: *Non-adaptive* and *Adaptive* (Stakhanova et al., 2007b; Foo et al., 2005). In the non-adaptive model, the order of the responses remains the same during the life of the IRS software. In fact, there is no mechanism for tracing the behaviors of the deployed responses. In the adaptive model, the system has the ability to automatically and appropriately adjust the order of the responses based on response history (Stakhanova et al., 2007b). The response goodness concept was introduced by Stakhanova et al. (2007a), Foo et al. (2005) can be used to convert a non-adaptive model to an adaptive one.

Definition 2 (Response Goodness (RG)). *Response goodness represents the history of success (R_s) and failure (R_f) of each response to mitigate attack over time.*

The Algorithm 1 can be used to convert a non-adaptive model to an adaptive one:

Require: $\triangle$: list of responses
Require: τ: attack cost
1: $\triangle = \{R_1, R_2, R_3, ..., R_n\}$
2: **for each** $R \in \triangle$ **do**
3: $R_G(t) = \frac{\sum_{i=1}^{n} R_{s_i} - \sum_{j=1}^{m} R_{f_i}}{n+m}$
4: $R_E(t) = R_{cost}^{s|se|de} \times R_G(t)$
5: **end for**
6: $reorder(\triangle)$

Algorithm 1 – Adaptive intrusion response system.

For each response, first we calculate goodness factor (line 3). As a simple way, the response goodness is calculated by sum of success rates ($\sum_{i=1}^{n} R_{s_i}$) minus sum of failure rates ($\sum_{j=1}^{m} R_{f_i}$) divided by the total number of response deployment. Then, the response effectiveness (R_E) can be calculated by multiplying the response cost times response goodness (line 4). The response cost can be one of the cost functions explained in Section B: R_{cost}^{s}, R_{cost}^{se}, or R_{cost}^{de}. Finally, the adaptive model presents a new ordering list of responses (line 6).

2.5. *Response selection*

There are three response selection models: *static mapping*, *dynamic mapping*, and *cost-sensitive mapping*.

2.5.1. *Static mapping*

An alert is mapped to a predefined response. This model is easy to build, but its weakness is that the response measures are predictable by attackers (Toth and Kregel, 2002).

2.5.2. *Dynamic mapping*

The responses of this model are based on multiple factors, such as the system state, attack metrics (frequency, severity, confidence, etc.), and the network policy (Curtis and Carver, 2001). In other words, responses to an attack may differ, depending on the targeted host, for instance. One drawback of this model is that it does not learn anything from attack to attack, so the intelligence level remains the same until the next upgrade (White et al., 1996; Porras and Neumann, 1997).

2.5.3. *Cost-sensitive mapping*

This is an interesting technique that attempts to attune intrusion damage and response cost (Zhang et al., 2011; Mu and Li, 2010; Toth and Kregel, 2002; Zhang et al., 2009).

Definition 3 (Intrusion Damage Cost). *Intrusion damage cost represents the "amount of damage to an attack target when the IDS and other protective measures are either unavailable or ineffective (Wei et al., 2001)".*

The results of a risk assessment are very important, in terms of minimizing the performance cost of applying strong responses, as a weak response is enough to mitigate a weak attack. Some cost-sensitive approaches have been proposed (e.g., Stakhanova et al., 2007a; Foo et al., 2005; Papadaki and Furnell, 2006) that use an offline risk assessment component, which is calculated by evaluating all the resources in advance. The value of each resource is static. In contrast, online risk assessment components can help accurately measure intrusion damage. The challenge with online risk assessment is the accuracy of calculating intrusion damage. In case of inaccurate calculation, the IRS may select an unduly high impact response for the network or apply a weaker response.

Intrusion risk assessment is very important in cost-sensitive mapping. Many real-time risk assessment models have been proposed during the last decade. As illustrated in Fig. 2, the proposed approaches can be grouped into three main categories:

(i) ***Attack Graph-based***: The attack graphs not only help to identify attacks, but also to quantitatively analyze their impact on the critical services in the network, based on the attackers behavior and vulnerabilities that can be exploited (Jahnke et al., 2007; Kanoun et al., 2008; Dantu et al., 2004). The attack graph is a useful model that can show the attack paths in a network based on service vulnerabilities (Jha et al., June 2002; Wang et al., 2008). It not only correlates the intrusion detection system (Noel

and Jajodia, 2005; Wang et al., 2006) outputs, but also helps intrusion response systems to apply responses in a timely fashion, at the right place, and with the appropriate intensity (Jahnke et al., 2007; Kanoun et al., 2008). One challenge in this approach is attack modeling. The correlation methods proposed in the last decade to connect attack steps can be classified into three categories (Kanoun et al., 2007; Totel et al., 2004): *implicit*, *explicit*, and *semi-explicit* correlations.

The implicit correlation attempts to find similarities between alerts in order to correlate them. In the explicit correlation, attack scenarios have to be defined statically. The attack signatures form the attack graph (Goubault-Larrec, 2001). The semi-explicit correlation type generalizes the explicit method by introducing preconditions and postconditions for each step in the attack graph (Cuppens and Ortalo, 2000).

(ii) ***Service Dependency Graph-based***: Three properties are defined for each service: $C(S)$, $I(S)$, and $A(S)$, which denote the confidentiality, integrity, and availability of service (S) respectively. The impact of the attack on a service is propagated to other services based on the type of dependency. In this type of approach, the attack graph is not used to evaluate attack cost (Kheir et al., 2010).

(iii) ***Non Graph-based***: Risk assessment is carried out independently of the attack detected by the IDS. This means that the IDS detects an attack and sends an alert to the risk assessment component, which performs a risk analysis based on alert statistics and other information provided in the alert(s) (Gehani and Kedem, 2004; Mu et al., 2008; Arnes et al., 2005; Haslum et al., 2008b).

2.6. *Applying location*

Most IRSs apply responses either on the attacked machine or the intruders machine if it is accessible. By extracting the "attack path", we can identify appropriate locations, those with the lowest penalty cost, for applying them. Moreover, responses can be assigned to calculate the dynamic cost associated with the location type, as discussed in the "Response cost model" section. The numerous locations and the variety of responses at each location will constitute a more effective framework for defending a system from attack, as its behavior will be less predictable. An attack path consists of four points: 1) the start point, which is the intruder machine; 2) the firewall point, which includes firewalls and routers; 3) the midpoint, which includes all the intermediary machines that the intruder exploits (through vulnerabilities) to compromise the target host; and 4) the end point (the intruders target machine). Despite the research advances in the detection of attack paths (Chen et al., 2007; Zhang et al., 2008; Savage et al., 2000), this method has rarely been implemented in actual IDSs or IRSs.

2.7. *Deactivation ability*

The need to deactivate a response action is not recognized in the majority of existing automated IRS. The importance of this need was first suggested in Kanoun et al. (2010). The authors argue that most responses are temporary actions which have an intrinsic cost or can even induce side effects on the monitored system. The question is how and when to deactivate the response. The deactivation of policy-based responses is not a trivial task.

3. Classification of existing models

3.1. *Response cost*

Lee et al. (2002) proposed an intrusion response system based on cost factors. Attack damage and response costs have been statically defined based on four categories (ROOT, R2L, DoS, and PROBE). Maximum damage cost is 100 considered for ROOT category meanwhile minimum damage cost is 2 allocated for PROBE category. Maximum response cost is 60 considered for ROOT category when attack is trying from a remote host. In contrast, minimum response cost is 5 considered for PROBE category when probing is being done in a short period of time. In this work there is not any list and evaluation of responses. The important feature of this work from response cost view is that response cost has tight relationship with attack category.

Papadaki and Furnell (2006) proposed a static evaluated cost response system. To evaluate the characteristics of each response action, they have proposed the following parameters: *counter-effects*, *stopping power*, *transparency*, *efficiency*, and *confidence level*. Also, the proposed model assesses the static and dynamic contexts of the attack. A database for analyzing the static context is needed to manage important characteristics of an attack, such as targets, applications, vulnerabilities, and so on. In terms of evaluating the dynamic context of an attack, there are some interesting ideas embodied in the proposed model. The two main features of this model are: 1) the ability to easily propose different orders of responses for different attack scenarios; and 2) the ability to adapt decisions in response to changes in the environment.

Strasburg et al. (2009) proposed a structured methodology for evaluating the cost of a response based on three parameters: operational cost (OC), impact of the response on the system (RSI), and response goodness (RG). The response cost model is: $RC = OC + RSI - RG$. OC refers to the cost of setting up and developing responses. The RSI quantifies the negative effect of the response on the system resources. RG is defined based on two concepts: 1) the number of possible intrusions that the response can potentially address; 2) the amount of resources that can be protected by applying the response.

Dynamic evaluated response cost approach is firstly proposed in Toth and Kregel (2002). Toth and Kregel (2002) presented a network model that takes into account relationships between users and resources, since users perform their activities by utilizing the available resources. The goal of a response model is to keep the system in as high a state of usability as possible. Each response alternative (which node to isolate) is inserted temporarily into the network model and a calculation is performed to determine which response has the lowest negative impact on services. In this model, every service has a static cost, and there is only the "block IP" response to evaluate as a way to repel an attack. When the IDS detects

an incoming attack, an algorithm attempts to find the firewall/gateway that can effectively minimize the penalty cost of the response action.

3.2. *Response time*

Tanachaiwiwat et al. (2002) proposed a non-adaptive response system. Although they claim that their method is adaptive, they have, in fact, implemented a non-adaptive mechanism. They point out that verifying the effectiveness of a response is quite expensive. They check, IDS efficiency, alarm frequency (per week), and damage cost, in order to select the best strategy. The alarm frequency reveals the number of alarms triggered per attack, and damage cost assesses the amount of damage that could be caused by the attacker. An appropriate list of response is available in the proposed model.

In (2007a), Stakhanova et al. proposed a proactive IRS. This model focuses on detecting anomalous behavior in software systems. It monitors system behaviors in terms of system calls, and has two levels of classification mechanism to detect intrusion. In the first detection step, when both normal and abnormal patterns are available, the model attempts to determine what kind of pattern is triggered when sequences of system calls are monitored. If the sequences do not match the normal or abnormal patterns, the system relies on machine learning techniques to establish whether the system is normal or anomalous. These authors have presented a response system that is automated, cost-sensitive, preemptive, and adaptive. The response is triggered before the attack completes.

Haslum et al. (2007) proposed a real time intrusion prevention model. They designed a prediction model based on the Hidden Markov Model (HMM) to model the interaction between the intruder and the network (Haslum et al., 2008a). The proposed HMM is based on four states: *Normal*, *Intrusion Attempt*, *Intrusion in progress*, and *Successful attack*. When the attacker gets appropriate results in attack, system moves from *Normal* state to the *Intrusion attempt* state and so on. When the probability of *Normal* state is down, it means the probability of other states are up. That model can detect the U2R, R2L, and PROBE categories of attacks, but not the DoS category.

3.3. *Adjustment ability*

Foo et al. (2005) presented a graph-based approach, called ADEPTS. The responses for the affected nodes are based on parameters such as confidence level of attack, previous measurements of responses in similar cases, etc. The model is adaptive and ADEPTS uses a feedback mechanism to estimate the success or failure of an applied response.

Stakhanova et al. (2007a) proposed an adaptive IRS. There is a mapping between system resources, response actions, and intrusion patterns which has to be defined in advance. Whenever a sequence of system calls matches a prefix in an abnormal graph, the response algorithm decides whether to repel the attack or not, based on a confidence level threshold. Multiple candidate responses may be available, and the one with the least negative effect is selected based on utility theory. The effectiveness of each applied response is measured for future response selection. If the selected response succeeds in neutralizing the attack, its success factor is increased by one, otherwise it is decreased by one.

3.4. *Response selection*

Chen and Yang (2004) proposed a static-mapping intrusion detection and prevention system based on firewalls. The idea is an attack response matrix which maps attack types to some responses. They do not consider trading off security enforcement levels and system performance.

Curtis and Carver (2001), Carver and Pooch (2000), Carver et al. (2000) propose a complex dynamic mapping based on an agent architecture (AAIRS). In AAIRS, multiple IDS monitor a host and generate alarms. The alarms are first processed by the Master Analysis agent. This agent indicates the confidence level of the attack and passes it on to an Analysis agent, which then generates a response plan based on *degree of suspicion*, *attack time*, *attacker type*, *attack type*, *attack implications*, *response goal*, and *policy constraints*.

Lee et al. (2002) proposed a cost-sensitive model based on three factors: 1) operational cost, which refers to the cost of processing the stream of events by an IDS; 2) damage cost, the amount of damage to a resource caused by an attacker when the IDS is ineffective; and 3) response cost, which is the cost of applying a response when an attack is detected.

Balepin et al. (2003) presented a dynamic cost-sensitive model and a response cost model. They proposed a local resource dependency model to evaluate responses. Their approach considers the current state of the system so as to calculate the response cost. Each resource has common response measures associated with the current state. The authors argue that designing a model to assess the value of each resource is a difficult task, so they rank the resources by their importance to produce a cost configuration. Then, static costs are assigned to high priority resources. Costs are injected into the resource dependency model when associated resources are involved in an incident. A particular response for a node is selected based on three criteria: 1) response benefit: sum of costs of resources that response action restores to a working state, 2) response cost: sum of costs of resources which is negatively affected by the response action, and 3) attack cost: sum of costs of resources that are negatively affected by the intruder. This approach suffers from multiple limitations. First, it is not clear how the response benefit is calculated in terms of confidentiality and integrity. Secondly, restoring the state of resources alone cannot be used to evaluate the response positive effect (Kheir et al., 2010). Finally, the proposed model is applicable for host-based intrusion response systems. Its application to network-based intrusion response requires significant modifications in the cost model (Kheir et al., 2010).

Mu and Li (2010) presented a hierarchical task network planning model to repel intrusions. In their approach, every response has an associated static risk threshold that can be calculated by its ratio of positive to negative effects. The permission to run each response is based on the current risk index of the network. When the risk index is greater than the response static threshold, the next response is allowed to run. The authors proposed a response selection window, where the most effective responses are selected to repel intrusions.

There is no evaluation of responses in this work. Also, it is unclear how the positive and negative effects of responses have been calculated. In that framework, the communication component is responsible for receiving alerts from multiple IDSs. The authors proposed to use an intrusion response planning to find a sequence of actions that achieve a response goal. These goals are the same as those in Curtis and Carver (2001): *analyze the attack*, *capture the attack*, *mask the attack*, *maximize confidentiality*, *maximize integrity*, *recovery gracefully*, and *sustain service*. Each goal has its own sequence of responses. For example, if the goal is to analyze an attack, the earlier responses in the sequence have to be weak, but later responses have to be strong.

Zhang et al. (2009) presented an approach for measuring attack impact with the objective being to suggest rational response using cost-benefit analysis. The proposed architecture is composed of three components: events processor, system state estimator, and response actuator. Observable Markov Decision Process has been used in this model and the system states were classified into four coarse-grained categories, namely, normal, probing, under exploitation, and compromised states. The rational response is chosen by estimating system states, and taking a rational response. Two cost functions were defined for rational response policy: *maintenance cost* due to intrusion response and *failure cost* due to attack. The proposed automated response mechanism can tune the tradeoff between system maintenance cost and failure cost for achieving rational defense.

Kheir et al. (2010) proposed a cost-sensitive IRS based on a service dependency graph to evaluate the confidentiality and integrity impacts, as well as the availability impact. The authors argue that it is really difficult to identify the impact on data confidentiality and integrity of other resources when we apply a response on a resource. To address this problem, the authors use a specific type of responses (e.g., "*allow unsecure connections*") (Kheir et al., 2009) in case of an openSSL attack. They targeted specific response that has negative effect on data confidentiality and integrity.

3.4.1. *Cost-sensitive mapping with dynamic risk assessment*

Kanoun et al. (2008) presented a risk assessment model based on attack graphs to evaluate the severity of the total risk of the monitored system. The LAMBDA (Cuppens and Ortalo, 2000) language is used to model attack graphs when an attack is detected. When an attack graph is obtained, the risk gravity model begins to compute the risk, which is a combination of two major factors: (i) *Potentiality*, which measures the probability of a given scenario taking place and successfully achieving its objective. Evaluating this factor is based on calculating its minor factors: *natural exposition*, and *dissuasive measures*. The first of these minor factors measures the natural exposure of the target system facing the detected attack. To reduce the probability of an attack progressing, the second minor factor, dissuasive measures, can be enforced. (ii) *Impact*, which is defined as a vector with three cells that correspond to the three fundamental security principles: Availability, Confidentiality, and Integrity. The interesting point with this model is that the impact parameters are calculated dynamically. That impact depends on the importance of the target assets, as well as the impact of the level of reduction measures deployed on the system to reduce and limit the impact, when the attack is successful.

Wang et al. (2013) presented a middleware approach to bridge the gap between system-level vulnerabilities and organization-level security metrics. The model is fundamentally different from previous methods because it uses dependency attack graphs rather than state-based attack graphs to represent network observations. The proposed approach systematically integrates attack graphs and Hidden Markov Models together for exploring the probabilistic relation between system observations and states. It then applies a cost-driven heuristic algorithm to search for the optimal security hardening from a list of countermeasure candidates. A set of security metrics and defence cost factors was specified in this work for calculating attack cost and defense cost. Attack impact was measured by confidentiality loss, denial of service, public embarrassment, privilege escalation, and integrity loss, while defence cost factors was calculated by system downtime, installation cost, operation cost, training cost, and incompatibility cost.

Jahnke et al. (2007) presented a graph-based approach for modeling the effects of attacks against resources and the effects of the response measures taken in reaction to those attacks. The proposed approach extends the idea put forward by Toth and Kregel (2002) by using general, directed graphs showing dependencies between resources and by deriving quantitative differences between system states from these graphs. If we assume that $G1$ and $G2$ are the graphs we obtain before and after the reaction respectively, then calculation of the responses positive effect is the difference between the availability plotted in the two graphs: $A(G2) - A(G1)$. Like Toth and Kregel (2002), Balepin et al. (2003), these authors focus on the availability impacts.

Kheir et al. (2010) proposed a dependency graph to evaluate the confidentiality and integrity impacts, as well as the availability impact. The confidentiality and integrity criteria are not considered in Jahnke et al. (2007). In Kheir et al. (2010), the impact propagation process proposed by Jahnke et al. is extended to include these impacts. Now, each service in the dependency graph is described with a 3D CIA vector, the values of which are subsequently updated, either by actively monitoring estimation or by extrapolation using the dependency graph. In the proposed model, dependencies are classified as structural or functional dependencies.

In (2005), Årnes et al. presented a non graph-based real-time risk assessment method for information systems and networks based on observations from network sensors. The proposed model is a multi-agent system where each agent observes objects in a network using sensors. An object is any kind of asset in the network that is valuable in terms of security. To perform dynamic risk assessment with this approach, discrete-time Markov chains are used. For each object, a Hidden Markov Model (HMM) is considered and the HMM states illustrate the security state, which changes over time. The proposed states are: *Good*, *Attacked*, and *Compromised*. The compromised state indicates that the host has been compromised. Thus, each object in the network can be in a different state at any time. In their model, it is assumed that there is no relationship between objects and that the HHMs work independently. A static cost, C_i, is allocated to each state,

Table 1 – Classification of existing IRSs based on proposed taxonomy.

IRS	Response selection	Risk assessment	Manage false positive	Response time	Adjustment ability	Response cost	Response lifetime
DC&A (Fisch, 1996)	Dynamic		No	Delayed	Non-adaptive	Static	Sustainable
CSM (White et al., 1996)	Dynamic		No	Delayed	Non-adaptive	Static	Sustainable
EMERALD (Porras and Neumann, 1997)	Dynamic		No	Delayed	Non-adaptive	Static	Sustainable
BMSL-based response (Bowen et al., 2000)	Static		No	Delayed	Non-adaptive	Static	Sustainable
SoSMART (Musman and Flesher, 2000)	Static		No	Delayed	Non-adaptive	Static	Sustainable
PH (Somayaji and Forrest, 2000)	Static		No	Delayed	Non-adaptive	Static	Sustainable
Lee's IRS (Lee et al., 2002)	Cost-sensitive	Static	No	Delayed	Non-adaptive	Static	Sustainable
AAIRS (Curtis and Carver, 2001; Carver and Pooch, 2000; Carver et al., 2000; Ragsdale et al., 2000)	Dynamic		No	Delayed	Adaptive	Static Evaluated	Sustainable
SARA (Lewandowski et al., 2001)	Dynamic		No	Delayed	Non-adaptive	Static	Sustainable
CITRA (Schnackenberg et al., 2001)	Dynamic		No	Delayed	Non-adaptive	Static	Sustainable
TBAIR (Wang et al., 2001)	Dynamic		No	Delayed	Non-adaptive	Static	Sustainable
Network IRS (Toth and Kregel, 2002)	Cost-sensitive	Static	No	Delayed	Non-adaptive	Dynamic	Sustainable
Tanachaiwiwat's IRS (Tanachaiwiwat et al., 2002)	Cost-sensitive	Static	No	Delayed	Non-adaptive	Static	Sustainable
Specification-based IRS (Balepin et al., 2003)	Cost-sensitive	Dynamic (SDG)[a]	No	Delayed	Non-adaptive	Dynamic	Sustainable
ADEPTS (Foo et al., 2005)	Cost-sensitive	Static	No	Proactive	Adaptive	Static	Sustainable
FAIR (Papadaki and Furnell, 2006)	Cost-sensitive	Static	No	Delayed	Non-adaptive	Static Evaluated	Sustainable
Stakhanova's IRS (Stakhanova et al., 2007a)	Cost-sensitive	Static	No	Proactive	Adaptive	Static Evaluated	Sustainable
DIPS (Haslum et al., 2007)	Cost-sensitive	Dynamic (NG)[b]	No	Proactive	Non-adaptive	Static	Sustainable
Jahnke (Jahnke et al., 2007)	Cost-sensitive	Dynamic (AG)[c]	No	Delayed	Non-adaptive	Dynamic	Sustainable
Strasburg's IRS (Strasburg et al., 2009)	Cost-sensitive	Static	No	Delayed	Adaptive	Static Evaluated	Sustainable
Zhang's IRS (Zhang et al., 2009)	Cost-sensitive	Dynamic (NG)	*Yes*	Delayed	Non-adaptive	Static	Sustainable
IRDM-HTN (Mu and Li, 2010; Mu et al., 2008)	Cost-sensitive	Dynamic (NG)	*Yes*	Delayed	Non-adaptive	Static Evaluated	Sustainable
OrBAC (Kanoun et al., 2008; Kanoun et al., 2010)	Cost-sensitive	Dynamic (AG)	No	Proactive	Adaptive	Static Evaluated	Deactiveable
Kheir's IRS (Kheir et al., 2010; Kheir et al., 2009)	Cost-sensitive	Dynamic (SDG)	No	Proactive	Non-adaptive	Dynamic	Sustainable
Wang's IRS (Wang et al., 2013)	Cost-sensitive	Dynamic (AG)	No	Delayed	Non-adaptive	Dynamic	Sustainable

[a] SDG: Service Dependency Graph-based.
[b] NG: Non Graph-based.
[c] AG: Attack Graph-based.

S_i. The total risk for each object at time t can be calculated as: $R_t = \sum_{i=1}^{n} \gamma_t(i)C(i)$. The $\gamma_t(i)$ value gives the probability that the object is in state S_i at time t.

Gehani and Kedem (2004) presented a non graph-based real-time risk management model, called *RheoStat*. This model dynamically alters the exposure of a host to contain an intrusion when it occurs. A host's exposure consists of the exposure of all its services. To analyze a system's risk, a combination of three factors is considered: 1) the likelihood of occurrence of an attack; 2) the impact on assets, i.e., the loss of confidentiality, integrity, and availability; and 3) the vulnerability's exposure, which is managed by safeguards.

Haslum et al. (2008b) proposed a fuzzy model for online risk assessment in networks. Human experts rely on their experience and judgment to estimate risk based on a number of dependent variables. Fuzzy logic is applied to capture and automate this process. The knowledge of security and risk experts is embedded in rules for a fuzzy automatic inference

Table 2 – Comparison of existing online intrusion risk assessment approaches.

IRA	Technique used	Quantitative	Qualitative	Hybrid
Balepin et al. (2003)	Decision Theory Convention	√		
Gehani and Kedem (2004)	Addition, Multiplication, and Division Operations	√		
Arnes et al. (2005)	Hidden Markov Model	√		
Papadaki and Furnell (2006)	Decision-making		√	
Jahnke et al. (2007)	Attack Graph	√		
Mu et al. (2008)	Dempster-Shafer Theory			√
Kanoun et al. (2008)	Attack Graph	√		
Haslum et al. (2008b)	Fuzzy Logic	√		
Kheir et al. (2009)	Service Dependency Graph	√		
Zhang et al. (2009)	Decision Theoretic	√		
Wang et al. (2013)	Dependency attack graph	√		

system. The main contribution is the use of fuzzy logic controllers. These were developed to quantify the various risks based on a number of variables derived from the inputs of various components. The fuzzy model is used to model *threat level*, *vulnerability effect*, and *asset value*. Threat level *(FLC-T)* is modeled using three linguistic variables: *Intrusion frequency*, *Probability of threat success*, and *Severity*. The HMM module used for predicting attacks provides an estimate of intrusion frequency. The asset value *(FLC-A)* is derived from three other linguistic variables: *Cost*, *Criticality*, *Sensitivity*, and *Recovery*. In addition, the vulnerability effect *(FLC-V)* has been modeled as a derived variable from *Threat Resistance* and *Threat Capability*. Eventually, the risk is estimated based on the output of the three fuzzy logic controllers *FLC-T, FLC-A, and FLC-V*.

Mu et al. (2008) proposed a non graph-based real-time risk assessment model based on *D–S evidence theory*. D–S evidence theory is a method for solving a complex problem where the evidence is uncertain or incomplete. The proposed model consists of two steps, which identify: *Risk Index* and *Risk Distribution*. In the first step, the risk index has to be calculated. The risk index is the probability that a malicious activity is a true attack and can achieve its mission successfully. In D–S evidence theory, five factors are used to calculate the risk index: *Number of alerts*, *Alert Confidence*, *Alert Type*, *Alert Severity*, and *Alert Relevance Score*. Risk distribution is the real evaluation of risk with respect to the value of the target host, and can be *low*, *medium*, or *high*. The risk distribution has two inputs: the risk index, and the value of the target host. The latter depends on all the services it provides.

3.5. *Deactivation ability*

The importance of deactivation ability in IRS was first suggested in Kanoun et al. (2010). Kanoun et al. specified two associated event-based contexts for each response: *Start (response context)*, and *End (response context)*. The risk assessment component can also help decide when a countermeasure has to be deactivated. In Kanoun et al. (2010), countermeasures are classified into one of two categories, in terms of their lifetime: 1) One-shot countermeasures, which have an effective lifetime that is negligible. When a response in this category is launched, it is automatically deactivated; and 2) Sustainable countermeasures, which remain active to deal with future threats after a response in this category is applied.

4. Discussion

A list of research studies on intrusion response systems and intrusion risk assessment systems in the last two decades is given in Table 1. As we can see, the cost-sensitive approaches have been the common paradigm for designing IRSs. Intrusion risk assessment mechanism is very important in cost-sensitive mapping. As seen in Table 1, recent proposed approaches use either attack graph-based (Jahnke et al., 2007; Kanoun et al., 2010) or service dependency graph-based (Kheir et al., 2010) methods to calculate multi-step attack costs online. We propose to use a combination of both to compute the damage cost and accurately react to attacks. In fact, when we use the attack graph approach for calculating risk, we do not have any knowledge about the true value of the compromised service, nor do we know the real impact of an attacker gaining full access to a compromised service based on predefined service permissions. In contrast, when we use the second method to calculate the risk separately, we do not have any information about the intruder's knowledge level. Therefore, an accurate attack cost is obtained based on information provided by service dependency and attack graphs. Eventually, the response selection module applies a response in which the attack and response costs are in proportion. Risk assessment methodologies are classified into three main categories: *Quantitative*, *Qualitative*, and *Hybrid*. Table 2 compares the online intrusion risk assessment approaches discussed in this paper (listed in Table 1).

As we can see from the table, the most common approaches used for online mode is the quantitative approach. Quantitative approaches rely on hard numbers, complex calculations, probability theory and statistics to determine the risk exposure and they may be difficult for non-technical people to interpret (Hulitt and Vaughn, 2010; Lo and Chen, 2012).

Qualitative approaches use classes and relative values to show the impact and probability of a particular scenario (Ekelhart et al., 2007). They also assess information security risks by employing methods and principles with non-numerical levels. The result of qualitative approaches is vastly dependent on the security experts who conduct the risk analysis (Hulitt and Vaughn, 2010; Karabacak and Sogukpinar, 2005). To address the weaknesses of both methodologies, some models have been proposed (e.g. Mu et al., 2008) to

combine the best of both worlds into a unique hybrid approach.

Many IRS models choose responses according to raw IDS alerts. This may lead to false positive responses because of the high IDS false positive alert rate (Lee et al., 2006; Spathoulas and Katsikas, 2010; Lin et al., 2013). Unfortunately, there have not been many studies that address the tolerance of IRSs to false positive IDS alerts. Mu and Li (2010), Zhang et al. (2009) proposed a model to control false positives in IRS. Mu and Li (2010) defined a risk threshold for each countermeasure where an online risk assessment module measures the alert risk. Since the risk value for a false positive is not high, it cannot reach the countermeasure risk threshold. Zhang et al. (2009) proposed an alert correlation mechanism to combine the low-level alerts of a group of IDS sensors into one observation vector and treating them as a whole.

In addition, as discussed in the paper, the adjustment ability of IRS is an important factor that influences the strength of the responses against attack over time. As we see in Table 1, only five IRS out of twenty five supports adjustment ability (Stakhanova et al., 2007a; Kanoun et al., 2010; Mu and Li, 2010; Curtis and Carver, 2001; Foo et al., 2005). The response goodness (R_G) concept plays a critical role in the adaptive approach that was introduced by Stakhanova et al., (2007a) and Foo et al. (2005). This parameter shows the history of each response (success or failure) in the past to mitigate an attack. One way to measure the success or failure of a response is to use the result of the online risk assessment component. R_G can be calculated as proposed by Stakhanova et al. in (2007a): if the selected response succeeds in neutralizing the attack, its success factor (R_s) is increased by one, otherwise, its failure factor (R_f) is increased. Unfortunately, the current solutions to calculate response goodness do not consider the time factor. A point of interest is that the most recent results must be considered more valuable than the earlier ones. For example, assume the results of R_s and R_f for a response are 10 and 3 respectively, the most recent result being: {F, F, F}. If we calculate the response goodness based on Algorithm 1 (line 3), R_G is equal to 0.54. Unfortunately, although $R_G = 0.54$ indicates that this response is a good one, and it was appropriate for mitigating the attack, over time and with the occurrence of new attacks, this response becomes insufficient to stage a counter attack.

Response cost evaluation is an important part of an IRS. As we can see in Table 1, the majority of the proposed IRSs use *Static Cost* or *Static Evaluated Cost* models (Strasburg et al., 2009; Stakhanova et al., 2007a; Kanoun et al., 2010; Mu and Li, 2010; Lee et al., 2002; Haslum et al., 2007; Curtis and Carver, 2001; White et al., 1996; Porras and Neumann, 1997; Fisch, 1996; Bowen et al., 2000; Musman and Flesher, 2000; Somayaji and Forrest, 2000; Carver and Pooch, 2000; Carver et al., 2000; Ragsdale et al., 2000; Lewandowski et al., 2001; Schnackenberg et al., 2001; Wang et al., 2001; Tanachaiwiwat et al., 2002; Foo et al., 2005; Papadaki and Furnell, 2006). Only five dynamic evaluated cost models have been used (Jahnke et al., 2007; Toth and Kregel, 2002; Kheir et al., 2010; Balepin et al., 2003; Wang et al., 2013). Considering service dependencies model to calculate response cost in IRS, firstly proposed by Toth and Kregel (2002). This approach suffers multiple limitations. First, they did not consider positive effect of responses. Evaluation of responses without considering positive effect leads to inaccurate evaluation. For example, if negative impact of response A is greater than response B, this does not mean that response B has to be applied first. Maybe, the positive effect of response A is better that B and if we calculate the response effectiveness, overall, response A is better. Second, from the security perspective (CIA), there exist no evaluation in terms of data confidentiality and integrity. Eventually, in the proposed model only the "block IP" response has been considered. In other words, it tries to decrease the availability of the target resource completely.

Similar to Toth and Kregel (2002), Balepin et al. (2003) presented a local resource dependency model to evaluate response in case of attack. Unlike (Toth and Kregel, 2002), this model considers the positive effects of responses. However, the authors' approach suffers from multiple limitations. First, it is not clear how response benefit is calculated in terms of confidentiality and integrity. Second, restoring the state of a resource cannot be only measured to evaluate the response positive effect as suggested by the authors. Note to mention that the proposed model is applicable to host-based intrusion response system only. To apply it to network-based intrusion response, it requires significant modifications to the cost model (Kheir, 2010).

Jahnke et al. (2007) proposed a graph-based approach for modeling the effects of attacks against resources and the effects of the response measures taken in reaction to those attacks. Kheir et al. (2010) proposed a dependency graph to evaluate the confidentiality and integrity impacts, as well as the availability impact. The confidentiality and integrity criteria are not considered in Jahnke et al. (2007). In Kheir et al. (2010), the impact propagation process proposed by Jahnke et al. is extended to include these impacts. To address this issue, the authors use a specific type of responses (e.g., *"allow unsecure connections"*) (Kheir et al., 2009) in case of an openSSL attack. They targeted a specific response that has negative effect on data confidentiality and integrity.

The response selection mechanism in *Dynamic* or *Cost-sensitive* mapping approaches can be done with a dynamic or static response list. In a dynamic response list, there are two approaches for the response ordering mechanism. The first approach is to order responses based on response cost (R_{cost}). As explained in this paper, response cost can be: *Static cost* (R^{s}_{cost}), *Static evaluated cost* (R^{se}_{cost}), or *Dynamic evaluated cost* (R^{de}_{cost}). If we chose the *Dynamic evaluated cost* model, our response list will be dynamic automatically. If the response cost be static (*Static cost* or *Static evaluated cost*), the sorted list of responses will remain static throughout an attack, and so it may be predictable by an intruder. One idea to convert a static response list to dynamic one is using *Goodness* factor, as illustrated in Eq. (2). We update the response effectiveness (R_E) by multiplying response cost by *Goodness* factor.

$$R_E(t) = R^{s|se}_{cost} \times R_G(t) \tag{2}$$

Even though a strong response may not be at the top of the ordered list when we initialize the response system, R_G being a dynamic factor causes it to move to that position over time. The higher the *Goodness* factor, the higher the response places in the ordered list over time. One drawback to using R_G is that

Table 3 – Attack and response costs parameters in Cost-sensitive approaches.

IRS	Attack cost parameter	Response cost parameter
Lee's IRS (Lee et al., 2002)	Attack type	Operational cost
Network IRS (Toth and Kregel, 2002)	N/A	Availability loss
Tanachaiwiwat's IRS (Tanachaiwiwat et al., 2002)	Monetary loss for each attack type	Monetary loss for each response type
Specification-based IRS (Balepin et al., 2003)	Availability loss	Availability loss, Availability gain
ADEPTS (Foo et al., 2005)	Alert confidence	Response goodness
FAIR (Papadaki and Furnell, 2006)	Alarm confidence, Perpetrator threat, Overall threat, Urgency, Number of systems at risk, Memory usage at target, CPU usage at target, Target idle?, Critical applications running?, Critical files open?, Other applications running?, Auditing software running?	Counter-effects, Stopping power, Transparency, Efficiency, Confidence
Stakhanova's IRS (Stakhanova et al., 2007a; Stakhanova, 2007)	N/A	Availability loss, Integrity loss, Confidentiality loss, System performance, Man-hours, Additional storage
DIPS (Haslum et al., 2007)	Intrusion frequency, Probability for threat success, Threat severity, Threat resistance, Threat capability, Asset cost, Asset criticality, Asset sensitivity, Asset recovery	N/A
Jahnke (Jahnke et al., 2007)	N/A	Availability loss
Strasburg's IRS (Strasburg et al., 2009)	Confidentiality loss, Integrity loss, Availability loss	Operational cost, Response goodness, Availability loss
Zhang's IRS (Zhang et al., 2009)	Confidentiality loss, Integrity loss, Availability loss	Maintenance cost
IRDM-HTN (Mu et al., 2008; Mu and Li, 2010)	Alert amount, Alert confidence, Alert type number, Alert severity, Alert relevance score	N/A
OrBAC (Kanoun et al., 2008; Kanoun et al., 2010)	Success Likelihood, History, Logging, Warning, Jurisdiction, Backup_Exist, Third_Party, Confidentiality loss, Integrity loss, Availability loss	Backup_Exist, Third_Party, Confidentiality loss, Integrity loss, Availability loss
Kheir's IRS (Kheir et al., 2010; Kheir et al., 2009)	Confidentiality loss, Integrity loss, Availability loss	Confidentiality loss, Integrity loss, Availability loss
Wang's IRS (Wang et al., 2013)	Confidentiality loss, Denial of service, Public embarrassment, Privilege escalation, Integrity loss	System downtime, Installation cost, Operation cost, Training cost, Incompatibility cost

it blocks the response selection mechanism after a while. Since a strong response is better to repel an attack, its *Goodness* attribute increases all the time. If we sort the responses based on R_G, we will be selecting the strong response all the time after a while, which is not desirable. Another drawback is that Quality of Service (QoS) in the network is not considered. Because many services are available and accessed by large numbers of users, it is important to maintain the users' QoS, the response time of applications, and the critical services that are in high demand. Since, when we use R_G, the strongest response is selected in case of attack, we are restricting network functionality until the response is deactivated.

The second approach that we propose for future research is not to consider R_G in the response ordering mechanism, and instead, to start with a poor response. It does not matter if a poor response is applied, because in this case the current network risk level slips under the risk threshold, based on the response *Goodness*, and brings us very close to the risk threshold again. If the attack is in progress, the network risk passes the risk threshold again very quickly and response system applies the next response. This approach has two main benefits. The first one is that all non-optimal responses will be reconsidered, and one or more of them may be able to prevent the attack this time. So, even if one of the responses applied previously was inefficient, it may work for a new attack. The second is that user's needs are considered in terms of QoS. So, in this approach, we start with a poor response, and, when the attack is likely to prove dangerous for our network, stronger responses are applied and network functionality is reduced slowly.

In the cost-sensitive approach, attack and response costs are attuned. Table 3 illustrates the parameters that are defined in the surveyed cost-sensitive approaches (discussed in Table 1) to measure these two costs. The common way to calculate attack cost is by assessing the attack's impact on CIA (Confidentiality, Integrity, and Availability) (Strasburg et al., 2009; Kheir et al., 2010; Kheir et al., 2009; Zhang et al., 2009). Response cost, on the other hand, is only assessed by measuring the impact of the response on resource availability (Toth and Kregel, 2002; Balepin et al., 2003; Stakhanova, 2007). This makes it difficult to compare attack cost to response cost since they impact different security attributes. In other words, they do not use the same measurement unit. There have been a few studies that aim to reconcile the measurement unit used to measure response cost with the one used for attack cost. For example, Kheir et al. (2010), Kheir et al. (2009) proposed a

model that system under attack with strong protocol like https moves to another state that allows users to use unsecure protocol like http. The activation of unsecure protocol increases an attacker's ability to beat the system. Thus, this type of response affects not only on our critical resource availability but also on data confidentiality and integrity.

Balepin et al. (2003) attempted to put these two costs in the same measurement unit by simplifying the problem. For the attack cost, they considered the sum of costs of resources that are negatively affected by the intruder (availability loss). For the response cost, they considered not only negative effect of the response on services but also the sum of costs of resources to restore the system to a working state (availability gain).

In some models (Stakhanova et al., 2007a; Jahnke et al., 2007; Toth and Kregel, 2002), only the response cost is considered. For example, Toth and Kregel (2002) and Jahnke et al. (2007) applied different responses to a model to understand which one has the lowest negative effect on services. In contrast, in some other models (Mu et al., 2008; Mu and Li, 2010; Haslum et al., 2007), only the attack cost is calculated and the response cost has a static value. Haslum et al. (2007) classified attack cost parameters into three categories: *asset value*, *vulnerability effect*, and *threat impact*. Asset value was modeled as *cost*, *criticality*, *sensitivity*, and *recovery*. Vulnerability effect was measured by two criteria: *threat resistance* and *threat capability*. Threat impact was modeled as the frequency of attacks, the probability that an intruder succeeds to subvert security controls, and the severity of attacks. Usually asset value and vulnerability effect are calculated statically. Threat effect can be measured dynamically based on IDS results. Mu et al. (2008), Mu and Li (2010), calculated the attack cost by *alert amount, confidence, type, severity*, and *relevance score*. The attack cost results is in range of [0, 1]. A list of responses are distributed in a range of [0, 1] based on their static power cost. The online risk assessment component calculates the current risk, which is the sum of the previous and new risk costs. When the current risk reaches the first response cost threshold (the weakest response), the first response it is then applied to mitigate the attack. When this response could not stop the attacker, the current risk cost will reach to the next response cost and this strategy guaranties the balance between response and attack costs. The major weakness of this model is that the response effectiveness remains same during the attack period and does not use the response history to order responses.

There exist other models that evaluate attack and response costs but without putting them in the same measurement unit (e.g. Strasburg et al., 2009; Stakhanova et al., 2007a; Zhang et al., 2009; Wang et al., 2013; Stakhanova, 2007). These defined different parameters to calculate costs separately and then defined some techniques to compare them. For example, in (2007), Stakhanova et al. proposed more parameters to evaluate response in addition to CIA such as the Man-hours of labor required to deploy or manage the response, additional resources used to support a response, such as disk-space for additional logging.

Another observation that emerged from this study is almost all security studies are validated by applying them to old datasets (MIT Lincoln Laboratory, 2000; Uni of California, 1999). Their accuracy and ability to reflect real-world conditions is a major concern was argued by Davis et al., in (2011). Also, many datasets are internal and cannot be shared due to privacy issues, others are heavily anonymized, or they lack certain statistical characteristics. These shortcomings are important reasons why a perfect dataset has yet to exist (Shiravi et al., 2012). In order to better test and optimize the selection of these parameters, and compare with other IRS systems, it is necessary to start working towards building a large dataset of recent attacks. This dataset of attacks would need to be executable and should include the attacks, monitoring information, and system configuration (software packages, data, configuration, etc.), a major undertaking for any single research group. The main suggestion for future research on the development of IRS is to prepare a strong, real dataset of single and multi-step attacks. Such a dataset is needed by all security researchers and will be useful for testing the efficiency and scalability aspects of the intrusion response systems in real-time in large environments. Shiravi et al. (2012) proposed a set of guidelines to how to create valid datasets, which can be followed to create the new datasets.

5. Conclusion

The paper surveys existing techniques and tools for Intrusion Risk Assessment and Intrusion Response Systems. The main findings of this paper are, that despite two decades of research in the area, existing approaches suffer from serious limitations. First the online risk assessment component is not tightly integrated and attuned with the response system. As we discussed earlier, perfect coordination between the risk assessment mechanism and the response system leads to an efficient framework that is able to manage false positive and select appropriate response in which to be attuned to attack cost.

We also found that most adaptive IRSs do not support effective algorithms for updating response history over time. Many studies claim to achieve this but the review of the literature shows that they only support very basic mechanisms. For example, they do not consider time in their calculation of response goodness. Not considering time causes these technique to overlook the most recent results while they must be considered more valuable than earlier ones. Moreover, it is not clear how most studies measure response goodness (success or failure).

Another important limitation of existing studies lies in the assessment method used to evaluate the effectiveness of the approach. Most researchers only consider true positives (i.e., the number of correct responses). While true positive is an indication of accuracy, it only draws a partial picture. False positive must also be taken into account. It is important to know how responses for IRSs and risks for IRA have been wrongly identified.

In addition, most IRSs focus only on response activation. They do not consider response deactivation, which can take into account users needs in terms of quality of service. Finally, most attack graph methods look at the generation of complex attack graphs and the complexity of analyzing these large attack graphs. There has been little attention paid to real live

implementations for calculating damage costs. The response selection is also ineffective unless the attack context is taken into account, which is not the case is most studies.

We believe that these limitations are the main reasons that prevent these techniques from finding their place in commercial tools. To build on existing work, we propose, in this paper, to conduct further research in the following areas: 1) Adaptive IRS, 2) Attack context-aware response selection mechanism in IRS, 3) Dynamic response cost evaluation framework for IRS that meet network demands, 4) Elastic IRSs that consider response activation and deactivation by considering the rate of attack or network risk tolerance, and 5) Building dataset of single and multi-step attacks. Such a dataset is needed by all security researchers and will be useful for testing the IRSs and IRAs approaches.

Acknowledgment

This work is partly funded by Natural Sciences and Engineering Research Council of Canada Research Chair on Sustainable Smart Eco-Cloud, NSERC-950-229052 and by the NSERC CRDPJ 424371-11: ECOLOTIC Sustainable and Green Telco-Cloud.

REFERENCES

Adetunmbi AO, Falaki SO, Adewale OS, Alese BK. Network intrusion detection based on rough set and k-nearest neighbour. Int J Comput ICT Res 2008;2(1):60–6.

Ammann P, Wijesekera D, Kaushik S. Scalable, graph-based network vulnerability analysis. In: Proceedings of 9th ACM Conference on Computer and Communications Security (ACM-CCS 2002); 2002. pp. 217–24.

Anuar NB, Sallehudin H, Gani A, Zakaria O. Identifying false alarm for network intrusion detection system using hybrid data mining and decision tree. Malays J Comput Sci; 2008 ISSN: 0127-9084:110–5.

Anuar NB, Papadaki M, Furnell S, Clarke N. An investigation and survey of response options for intrusion response systems. In: Information Security for South Africa; 2010. pp. 1–8.

Arnes A, Sallhammar K, Haslum K, Brekne T, Moe M, Knapskog S. Real-time risk assessment with network sensors and intrusion detection systems. In: Computational Intelligence and Security, vol. 3802 of Lecture Notes in Computer Science; 2005. pp. 388–97.

Balepin I, Maltsev S, Rowe J, Levitt K. Using specification-based intrusion detection for automated response. In: Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection; 2003. pp. 136–54.

Berkhin P. Survey of clustering data mining techniques; 2001.

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; 2000. pp. 84–99.

Carver C, Pooch U. An intrusion response taxonomy and its role in automatic intrusion response. In: IEEE Workshop on Information Assurance and Security; 2000.

Carver C, Hill JM, Surdu JR. A methodology for using intelligent agents to provide automated intrusion response. In: IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop; 2000. pp. 110–6.

Chen YM, Yang Y. Policy management for network-based intrusion detection and prevention. In: IEEE Network Operations and Management Symposium; 2004.

Chen Y, Boehm B, Sheppard L. Value driven security threat modeling based on attack path analysis. In: 40th Hawaii International Conference on System Sciences, Big Island, Hawaii; January 2007.

Cuppens F, Ortalo R. Lambda: a language to model a database for detection of attacks. In: Proceedings of the Third International Workshop on Recent Advances in Intrusion Detection (RAID2000); 2000. pp. 197–216. Toulouse, France.

Curtis A, Carver J. Adaptive agent-based intrusion response [Ph.D. thesis]. USA: Texas A&M University; 2001.

Dantu R, Loper K, Kolan P. Risk management using behavior based attack graphs. In: Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04); 2004. pp. 445–9.

Davis JJ, Clark AJ. Data preprocessing for anomaly based network intrusion detection: a review. Comput Secur 2011;30(6):353–75.

Difference between Signature Based and Anomaly Based Detection in IDS, URL http://www.secguru.com/forum/difference between signature based and anomaly based detection in ids.

Ekelhart A, Fenz S, Klemen M, Weippl E. Security ontologies: improving quantitative risk analysis. In: The 40th Hawaii International Conference on System Sciences, Hawaii; 2007.

Feng L, Wang W, Zhu L, Zhang Y. Predicting intrusion goal using dynamic Bayesian network with transfer probability estimation. J Netw Comput Appl 2009;32(3):721–32.

Fisch E. A taxonomy and implementation of automated responses to intrusive behavior [Ph.D. thesis]. Texas A&M University; 1996.

Foo B, Wu YS, Mao YC, Bagchi S, Spafford E. ADEPTS: adaptive intrusion response using attack graphs in an e-commerce environment. In: International Conference on Dependable Systems and Networks; 2005. pp. 508–17.

Gehani A, Kedem G. Rheostat: real-time risk management. In: Recent Advances in Intrusion Detection: 7th International Symposium, (RAID 2004); 2004. pp. 296–314. France.

Goubault-Larrec J. An introduction to logweaver. Technical report. LSV; 2001.

Han J, Kamber M. Data mining: concepts and techniques. 2nd ed. San Francisco: Elsevier; 2006.

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: Proceedings of the 3rd International Symposium on Information Assurance and Security; 2007. pp. 183–8. Manchester, United Kingdom.

Haslum K, Moe MEG, Knapskog SJ. Real-time intrusion prevention and security analysis of networks using HMMs. In: 33rd IEEE Conference on Local Computer Networks; 2008. pp. 927–34. Montreal, Canada.

Haslum K, Abraham A, Knapskog S. Fuzzy online risk assessment for distributed intrusion prediction and prevention systems. In: Tenth International Conference on Computer Modeling and Simulation. Cambridge: IEEE Computer Society Press; 2008b. pp. 216–23.

Hulitt E, Vaughn RB. Information system security compliance to FISMA standard: a quantitative measure. Telecommun Syst 2010;45(2–3):139–52.

Jahnke M, Thul C, Martini P. Graph-based metrics for intrusion response measures in computer networks. In: Proceedings of the 3rd LCN Workshop on Network Security. Held in

conjunction with the 32nd IEEE Conference on Local Computer Networks (LCN); 2007. pp. 1035–42. Dublin, Ireland.
Jha S, Sheyner O, Wing J. Two formal analyses of attack graphs. In: Proceedings of the 15th Computer Security Foundation Workshop; June 2002.
Kanoun W, Cuppens-Boulahia N, Cuppens F, Autrel F. Advanced reaction using risk assessment in intrusion detection systems. In: Proceedings of the Second international conference on Critical Information Infrastructures Security; 2007. pp. 58–70. Spain.
Kanoun W, Cuppens-Boulahia N, Cuppens F, Araujo J. Automated reaction based on risk analysis and attackers skills in intrusion detection systems. In: Third International Conference on Risks and Security of Internet and Systems; 2008. pp. 117–24.
Kanoun W, Cuppens-Boulahia N, Cuppens F, Dubus S. Risk-aware framework for activating and deactivating policy-based response. In: Proceedings of the Fourth International Conference on Network and System Security; 2010. pp. 207–15.
Karabacak B, Sogukpinar I. ISRAM: information security risk analysis method. Comput Secur 2005;24(2):147–59.
Kheir N. Response policies and counter-measures: Management of service dependencies and intrusion and reaction impacts; 2010 [Ph.D. thesis].
Kheir N, Debar H, Cuppens-Boulahia N, Cuppens F, Viinikka J. Cost evaluation for intrusion response using dependency graphs. In: IFIP International Conference on Network and Service Security; 2009.
Kheir N, Cuppens-Boulahia N, Cuppens F, Debar H. A service dependency model for cost sensitive intrusion response. In: Proceedings of the 15th European Conference on Research in Computer Security; 2010. pp. 626–42.
Lazarevic A, Ertz L, Kumar V, Ozgur A, Srivastava J. A comparative study of anomaly detection schemes in network intrusion detection. In: Proceedings of the Third SIAM International Conference on Data Mining; 2003.
Lee W, Fan W, Miller M. Toward cost-sensitive modeling for intrusion detection and response. J Comput Secur 2002;10(1):5–22.
Lee S, Chung B, Kim H, Lee Y, Park C, Yoon H. Real-time analysis of intrusion detection alerts via correlation. Comput Secur 2006;25(3):169–83.
Lewandowski SM, Hook DJV, OLeary GC, Haines JW, Rossey ML. SARA: survivable autonomic response architecture. In: DARPA Information Survivability Conference and Exposition; 2001. pp. 77–88.
Lin YD, Lai YC, Ho CY, Tai WH. Creditability-based weighted voting for reducing false positives and negatives in intrusion detection. Comput Secur 2013;39:460–74.
Lo CC, Chen WJ. A hybrid information security risk assessment procedure considering interdependences between controls. Expert Syst Appl 2012;39(1):247–57.
MIT Lincoln Laboratory. 2000 DARPA intrusion detection scenario specific data sets; 2000.
Mu CP, Li Y. An intrusion response decision-making model based on hierarchical task network planning. Expert Syst Appl 2010;37(3):2465–72.
Mu CP, Li XJ, Huang HK, Tian SF. Online risk assessment of intrusion scenarios using D–S evidence theory. In: Proceedings of the 13th European Symposium on Research in Computer Security; 2008. pp. 35–48. Malaga, Spain.
Musman S, Flesher P. System or security managers adaptive response tool. In: DARPA Information Survivability Conference and Exposition; 2000. pp. 56–68.
Noel S, Jajodia S. Understanding complex network attack graphs through clustered adjacency matrices. In: Proceedings of the 21st Annual Computer Security Conference (ACSAC); 2005. pp. 160–9.
Papadaki M, Furnell SM. Achieving automated intrusion response: a prototype implementation. Inf Manag Comput Secur 2006;14(3):235–51.
Porras P, Neumann P. EMERALD: event monitoring enabling responses to anomalous live disturbances. Natl Inf Syst Secur Conf; 1997:353–65.
Ragsdale D, Carver C, Humphries J, Pooch U. Adaptation techniques for intrusion detection and intrusion response system. In: IEEE International Conference on Systems, Man, and Cybernetics; 2000. pp. 2344–9.
Sabhnani M, Serpen G. Formulation of a heuristic rule for misuse and anomaly detection for U2R attacks in Solaris operating system environment. In: Security and Management; 2003. pp. 390–6.
Savage S, Wetherall D, Karlin A, Anderson T. Practical network support for IP traceback. In: ACM SIGCOMM; August 2000. pp. 295–306.
Scarfone K, Mell P. Guide to intrusion detection and prevention systems. Technical report. NIST: National Institute of Standards and Technology, U.S. Department of Commerce; 2007.
Schnackenberg D, Holliday H, Smith R, Djadhandari K, Sterne D. Cooperative intrusion traceback and response architecture (CITRA). In: IEEE DARPA Information Survivability Conference and Exposition; 2001. pp. 56–68.
Shameli-Sendi A, Ezzati-Jivan N, Jabbarifar M, Dagenais M. Intrusion response systems: survey and taxonomy. Int J Comput Sci Netw Secur 2012a;12(1):1–14.
Shiravi A, Shiravi H, Tavallaee M, Ghorbani AA. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput Secur 2012;31(3):357–74.
Somayaji A, Forrest S. Automated response using system-call delay. In: Proceedings of the 9th USENIX Security Symposium; 2000. pp. 185–98.
Spathoulas GP, Katsikas SK. Reducing false positives in intrusion detection systems. Comput Secur 2010;29(1):35–44.
Stakhanova N. A framework for adaptive, cost-sensitive intrusion detection and response system (Ph.D. thesis). USA: Iowa State University; 2007.
Stakhanova N, Basu S, Wong J. A cost-sensitive model for preemptive intrusion response systems. In: Proceedings of the 21st International Conference on Advanced Networking and Applications. Washington, DC, USA: IEEE Computer Society; 2007a. pp. 428–35.
Stakhanova N, Basu S, Wong J. Taxonomy of intrusion response systems. J Inf Comput Secur 2007b;1(2):169–84.
Stein G, Bing C, Wu AS, Hua KA. Decision tree classifier for network intrusion detection with GA-based feature selection. In: Proceedings of the 43rd annual Southeast regional conference, Georgia, ISBN 1-59593-059-0. pp. 136–41.
Strasburg C, Stakhanova N, Basu S, Wong JS. A framework for cost sensitive assessment of intrusion response selection. In: Proceedings of IEEE Computer Software and Applications Conference; 2009. pp. 355–60.
Strasburg C, Stakhanova N, Basu S, Wong JS. The methodology for evaluating response cost for intrusion response systems. Technical Report 08-12. Iowa State University; 2008.
Tanachaiwiwat S, Hwang K, Chen Y. Adaptive intrusion response to minimize risk over multiple network attacks. ACM Trans Inf Syst Secur; 2002:1–30.
The Snort Project. Snort users manual 2.8.5; 2009.
Totel E, Vivinis B, Mé L. A language driven intrusion detection system for event and alert correlation. In: Proceedings at the 19th IFIP International Information Security Conference, Kluwer Academic, Toulouse; 2004. pp. 209–24.

Toth T, Kregel C. Evaluating the impact of automated intrusion response mechanisms. In: Proceedings of the 18th Annual Computer Security Applications Conference, Los Alamitos, USA; 2002.
University of California. KDD Cup 1999 data, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
Wang X, Reeves DS, Wu SF. Tracing based active intrusion response. J Inf Warefare 2001;1:50–61.
Wang L, Liu A, Jajodia S. Using attack graph for correlating, hypothesizing, and predicting intrusion alerts. Comput Commun 2006;29(15):2917–33.
Wang L, Islam T, Long T, Singhal A, Jajodia S. An attack graph-based probabilistic security metric. In: Proceedings of The 22nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DBSEC08); 2008.
Wang S, Zhang Z, Kadobayashi Y. Exploring attack graph for cost-benefit security hardening: a probabilistic approach. Comput Secur 2013;32:158–69.
Wei H, Frinke D, Carter O, Ritter C. Cost-benet analysis for network intrusion detection systems. In: CSI 28th Annual Computer Security Conference, Washington, DC; 2001.
White G, Fisch E, Pooch U. Cooperating security managers: a peer-based intrusion detection system. IEEE Netw 1996;10:20–3.
Xiao F, Jin S, Li X. A novel data mining-based method for alert reduction and analysis. J Netw 2010;5(1):88–97.
Yusof MF. Automated signature generation of network attacks [B.Sc. thesis]. University Teknologi Malasia; 2009.
Zhang Y, Fan X, Wang Y, Xue Z. Attack grammar: a new approach to modeling and analyzing network attack sequences. In: Proceedings of the Annual Computer Security Applications Conference (ACSAC 2008); 2008. pp. 215–24.
Zhang Z, Ho PH, He L. Measuring IDS-estimated attack impacts for rational incident response: a decision theoretic approach. Comput Secur 2009;28(7):605–14.
Zhang Z, Nat-Abdesselam F, Ho PH, Kadobayashi Y. Toward cost-sensitive self-optimizing anomaly detection and response in autonomic networks. Comput Secur 2011;30(6):525–37.
Zhou CV, Leckie C, Karunasekera S. A survey of coordinated attacks and collaborative intrusion detection. Comput Secur 2010;29(1):124–40.

Dr. Alireza Shameli-Sendi received his B.Sc. and M.Sc. with honors from Amirkabir University of Technology (Tehran Polytechnic). He received his Ph.D degree in computer engineering from École Polytechnique de Montréal, Montreal, Canada. He is currently doing PostDoc at Ecole de Technologie Superieure (University of Quebec) in collaboration with Ericsson Research Security Department (Montreal, Canada) with the aim to develop a new defence framework in Cloud Computing. His primary research interests include information security, vulnerability analysis, intrusion response system, and cloud computing.

Dr. Mohamed Cheriet received M.Sc. and Ph.D. degrees in Computer Science from the University of Pierre et Marie Curie (Paris VI) in 1985 and 1988 respectively. Dr. Cheriet is expert in cloud computing and network virtualization. In addition, he is an expert in Computational Intelligence, Pattern Recognition, Mathematical Modeling for Image Processing, Cognitive and Machine Learning approaches and Perception. Dr. Cheriet has published more than 300 technical papers in the field. He holds Canada Research Chair Tier 1 on Sustainable Smart Eco-Cloud.

Dr. Abdelwahab Hamou-Lhadj is an Associate Professor and the Undergraduate Program (Curriculum) Director at the Department of Electrical and Computer Engineering (ECE), Concordia University, Montreal, Canada. He has been working for many years in the area of software tracing and its applications to software maintenance and evolution. He leads a research lab that investigates techniques and tools to help software analysts understand the behavior of complex systems. Recently, he has been investigating the application of tracing techniques to cyber security and surveillance. He holds a Ph.D. degree in Computer Science from the University of Ottawa, ON, Canada. He is a member of IEEE and ACM. He can be reached at Wahab.hamou-lhadj@concordia.ca