Taxonomy of Intrusion Risk Assessment and Response System

Alireza Shameli-Sendi , Mohamed Cheriet, *Senior Member, IEEE*, and Abdelwahab Hamou-Lhadj, *Member, IEEE*

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.

Index Terms—Intrusion detection system, Intrusion response system, Intrusion risk assessment, Response time, Prediction, Response cost, Attack graph, Service dependency graph.

1 Introduction

TODAYS 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. To mitigate these threats, the deployment of an appropriate defense mechanism is needed. As Figure 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 behaviour.

IDSs vary depending on whether they monitor network traffic (Network-based IDS) or local hosts (Host-based IDS) [16]–[20]. IDSs are divided into two categories: *anomaly-based* and *signature-based*. Anomaly-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. [21]–[23]. 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) [24], 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

• *A. Shameli-Sendi and M. Cheriet are with the Department of Electrical and Computer Engineering, Ecole de Technologie Superieure (ETS), Montreal, Canada. E-mail: alireza.shameli@synchromedia.ca, mohamed.cheriet@etsmtl.ca. A. Hamou-Lhadj is with Department of Electrical and Computer Engineering, Concordia University, E-mail: wahab.hamou-lhadj@concordia.ca*

Manuscript received ...; revised ....

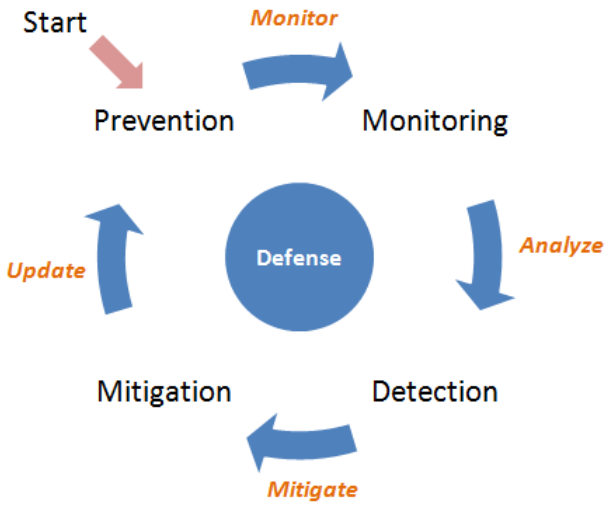

Fig. 1: Defense Life-cycle.

difficult to find the right balance between accuracy and generality, which may lead to false negatives and false positives [25], [26].

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 [1]. It also has to have an accurate evaluation of the cost of the response [2], [3]. 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 [4], [5]. 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 [6], [7]. 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) [9].

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. Also, a review of recent existing IRS and IRA is presented in this Section. Section 3, 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 4, 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 Figure 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.
- **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 life-time), 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 [10].

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 [13], [61]. 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 [34], [65]. The challenge of this approach is the delay between the intrusion and the human response [13], [28].

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 [31], [32]. One of the problems with this approach is the possibility that an inappropriate response will be executed when a problem arises [11]. 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 [14]. 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,

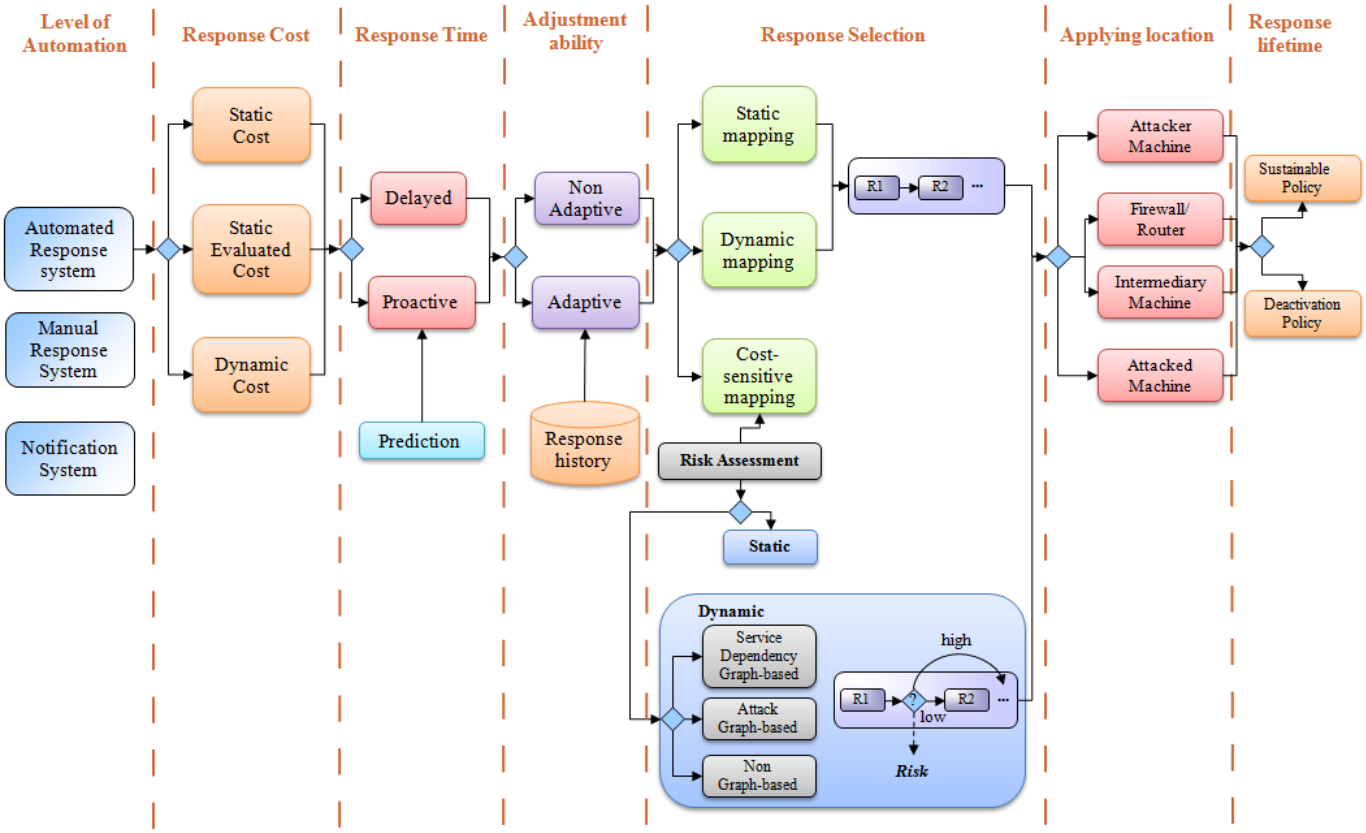

Fig. 2: Taxonomy of Intrusion Response Systems.

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 [28], [29]:

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 [30].
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 is obtained by assigning a static value based on an experts opinion. So, in this approach, a static value is considered for each response ($RC_s = CONSTANT$). Lee et al. [28] 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.

2.2.2 Static evaluated cost model

In this approach, a statically evaluated cost, obtained by an evaluation mechanism, is associated with each response ($RC_{sc} = f(x)$). The response cost in the majority of existing models is statically evaluated. A common solution is to evaluate the positive effects of the responses based on their consequences on confidentiality, integrity, availability, and performance. To evaluate the negative impacts, we can consider the consequences for the other resources in terms of availability and performance [2], [40]. 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 [11], obviously the higher RC, the better the response in ordering list:

$$RC_{se} = \frac{Positive_{effect}}{Negative_{impact}} \tag{1}$$

Papadaki and Furnell [67] 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. [2] 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.

2.2.3 Dynamic evaluated cost model

The dynamic evaluated cost is based on the network situation (RC_{de}). We can evaluate the response cost online based on the dependencies between resources [6], [48] 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.

Dynamic evaluated response cost approach is firstly proposed in [34]. Toth and Kruegel [34] 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.

2.3 Response time

In point of response time, IRSs can be classified into type categories: *Delayed* and *Proactive* [13], [41]. In the delayed mode, the responses are formulate only after an intrusion is detected. Most existing IRS use this approach (e.g., [2], [67]) 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 [18]. 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 [42] and it is often hard to guarantee that the prediction result is 100 accurate [13].

In [3], 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. [29] 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 [68]. 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.

2.4 Adjustment ability

There are two types of adjustment models: *Non-adaptive* and *Adaptive* [13], [66]. 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. Tanachaiwiwat et al. [65] 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 the adaptive model, the system has the ability to automatically and appropriately adjust the order of the responses based on response history [13].

Definition 2 (Response Goodness (G)). *Response goodness represents the history of success (S) and failure (F) of each response to mitigate attack over time*

The response goodness concept was introduced by [3], [66]. This parameter guarantees that our model will be adaptive and helps the IRS to prepare the best set of responses over time. The following procedure can be used to convert a non-adaptive model to an adaptive one [3]:

$$Goodness(t) = \frac{\sum_{i=1}^{n} S_i - \sum_{j=1}^{m} F_j}{\sum_{i=1}^{n} S_i + \sum_{j=1}^{m} F_j}$$

$$\begin{aligned} R_{effectiveness}(t) &= (RC_s | RC_{se} | RC_{de}) \times G(t) \\ R_{effectiveness}(t+1) &= R_{effectiveness}(t) \times G(t+1) \end{aligned} \quad (2)$$

Foo et al. [66] 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. [3] 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.

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 [34]. Chen et al. [58] proposed an 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.

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 [31]. 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 [32], [33]. Curtis et al. [31], [59], [60] 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*.

2.5.3 Cost-sensitive mapping

This is an interesting technique that attempts to attune intrusion damage and response cost [11], [34].

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 [8]".*

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., [3], [66], [67]) 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.

Lee et al. [28] 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. [50] 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 [48]. 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 [48].

Mu and Li [11] 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 [31]: *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.

Kheir et al. [48] 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"*) [49] in case of an openSSL attack. They targeted specific response that has negative effect on data confidentiality and integrity.

Risk Assessment in Cost-sensitive mapping

Many real-time risk assessment models have been proposed during the last decade. As illustrated in Figure 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 [6], [7], [72]. The attack graph is a useful model that can show the attack paths in a network based on service vulnerabilities [5], [69]. It not only correlates the intrusion detection system [70], [71] outputs, but also helps intrusion response systems to apply responses in a timely fashion, at the right place, and with the appropriate intensity [6], [7]. 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 [74], [75]: *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 [76]. The semi-explicit correlation type generalizes the explicit method by introducing preconditions and postconditions for each step in the attack graph [15].

Kanoun et al. [7] presented a risk assessment model based on attack graphs to evaluate the severity of the total risk of the monitored system. The LAMBDA [15] 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.

Jahnke et al. [6] 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 in [34] 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 [34], [50], these authors focus on the availability impacts.

(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 [48].

Kheir et al. [48] 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 [6]. In [48], 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.

(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) [1], [9], [47], [73].

In [47], Årnes et al. presented a 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, 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 et al. [1] presented a real-time risk man-

agement 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. [73] 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 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. [9] proposed an online 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.

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 [51]–[53], 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 [10]. 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. An efficient solution proposed by Kanoun et al. in [10] is to specify 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 [10], 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.

TABLE 1: Classification of existing IRSs based on proposed taxonomy.

IRS	Year	Response Selection	Risk Assessment	Manage False Positive	Response Time	Adjustment Ability	Response Cost	Response Lifetime
DC&A [54]	1996	Dynamic mapping		No	Delayed	Non-adaptive	Static Cost	Sustainable
CSM [32]	1996	Dynamic mapping		No	Delayed	Non-adaptive	Static Cost	Sustainable
EMERALD [33]	1997	Dynamic mapping		No	Delayed	Non-adaptive	Static Cost	Sustainable
BMSL-based response [55]	2000	Static Mapping		No	Delayed	Non-adaptive	Static Cost	Sustainable
SoSMART [56]	2000	Static Mapping		No	Delayed	Non-adaptive	Static Cost	Sustainable
PH [57]	2000	Static Mapping		No	Delayed	Non-adaptive	Static Cost	Sustainable
Lee's IRS [28]	2000	*Cost-sensitive*	Static Value	No	Delayed	Non-adaptive	Static Cost	Sustainable
AAIRS [31], [59]–[61]	2000	Dynamic mapping		No	Delayed	*Adaptive*	Static Evaluated Cost	Sustainable
SARA [62]	2001	Dynamic mapping		No	Delayed	Non-adaptive	Static Cost	Sustainable
CITRA [63]	2001	Dynamic mapping		No	Delayed	Non-adaptive	Static Cost	Sustainable
TBAIR [64]	2001	Dynamic mapping		No	Delayed	Non-adaptive	Static Cost	Sustainable
Network IRS [34]	2002	*Cost-sensitive*	Static Value	No	Delayed	Non-adaptive	*Dynamic Evaluated Cost*	Sustainable
Tanachaiwiwat's IRS [65]	2002	*Cost-sensitive*		No	Delayed	Non-adaptive	Static Cost	Sustainable
Specification-based IRS [50]	2003	*Cost-sensitive*	*Service Dependency Graph-based*	No	Delayed	Non-adaptive	*Dynamic Evaluated Cost*	Sustainable
ADEPTS [66]	2005	*Cost-sensitive*	Static Value	No	*Proactive*	*Adaptive*	Static Cost	Sustainable
FAIR [67]	2006	*Cost-sensitive*	Static Value	No	Delayed	Non-adaptive	Static Evaluated Cost	Sustainable
Stakhanova's IRS [3]	2007	*Cost-sensitive*	Static Value	No	*Proactive*	*Adaptive*	Static Evaluated Cost	Sustainable
DIPS [29]	2007	*Cost-sensitive*	Non Graph-based	No	*Proactive*	Non-adaptive	Static Cost	Sustainable
Jahnke [6]	2007	*Cost-sensitive*	Attack Graph-based	No	Delayed	Non-adaptive	*Dynamic Evaluated Cost*	Sustainable
Strasburg's IRS [2]	2008	*Cost-sensitive*	Static Value	No	Delayed	*Adaptive*	Static Evaluated Cost	Sustainable
IRDM-HTN [11]	2010	*Cost-sensitive*	Non Graph-based	*Yes*	Delayed	Non-adaptive	Static Evaluated Cost	Sustainable
OrBAC [10]	2010	*Cost-sensitive*	*Service Dependency Graph-based*	No	*Proactive*	*Adaptive*	Static Evaluated Cost	*Deactiveable*
Kheir's IRS [48]	2010	*Cost-sensitive*	*Service Dependency Graph-based*	No	*Proactive*	Non-adaptive	*Dynamic Evaluated Cost*	Sustainable

3 DISCUSSION

A complete list of overview 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.

3.1 Risk Assessment

As seen in Table 1, recent proposed approaches use either attack graph-based [6] or service dependency-based [10], [48] methods to calculate multi-step attack costs online. We propose to use both of these 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 permissions among services. In contrast, when we use the second method to calculate the risk separately, we do not have any information about the intruders 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.

3.2 Manage false positives

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. In terms of tolerance to false positive IDS alerts, only [11] proposes a model to control false positives in IRS. They define a risk threshold for each countermeasure. Then, an online risk assessment module measures the alert risk. Since the risk value for a false positive is not high, it cannot reach the countermeasures risk threshold.

3.3 Adjustment ability

As seen in Table 1, only five IRS supports adjustment ability [3], [10], [11], [31], [66]. The response goodness (G) concept plays a critical role in adaptive approach that was introduced by Stakhanova et al. and Foo et al. [3], [66]. This parameter shows the history of each response 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. G can be calculated as proposed by Stakhanova et al. in [3]: if the selected response succeeds in neutralizing the attack, its success factor (S) is increased by one, otherwise, its failure factor (F) is increased. Unfortunately, the current solutions to calculate response goodness do not consider the time in calculation. The important point to keep in mind is that the most recent results must be considered more valuable than the earlier ones. For example, assume the results of S and F for a response are 10 and 3 respectively, the most recent result being $F= 3$. If we calculate the response goodness based on Eq. 2, G is equal to 0.54. Unfortunately, although $G= 0.54$ 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 is not sufficiently strong to stage a counter attack.

3.4 Response Cost

As we can see in Table 1, the majority of the proposed IRSs use *Static Cost* or *Static Evaluated Cost* models [2], [3], [10], [11], [28], [29], [31]–[33], [54]–[57], [59]–[67]. Only three dynamic evaluated cost models have been used [6], [34], [48], [50]. Several works have been devoted to building a response selection mechanism based on the positive effects (P) and negative impacts (N) of the responses [2], [3], [11]. A common solution is to evaluate the positive effects based on their consequences for the CIA triad and for the performance metric. To evaluate the negative impact, we can consider the consequences for the other resources, in terms of availability and performance. There are two approaches for the response ordering mechanism: (i) the first approach is to order responses based on response cost (*RC*). RC can be obtained by combining the positive and negative factors. If the positive and negative factors are static, the sorted list of responses will remain static throughout an attack, and so it may be predictable by an intruder. We can use the Goodness factor to convert this list to a dynamic one, as illustrated in Eq. 3.

$$RC = f(P, N) * G \tag{3}$$

Even though the strong response is not at the top of the ordered list when we initialize the response system, 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 G is that it blocks the response selection mechanism after a while. Since a strong response is better able to repel an attack, its Goodness attribute increases all the time. If we sort the responses based on G, we will be selecting the strong response all the time after a while, which is not what we want. Another drawback is that Quality of Service (QoS) in the network is not considered. As we know, many services are available and accessed by large numbers of users. It is extremely important to maintain the users' QoS, the response time of applications, and the critical services that are in high demand. Since, when we use G, the strongest response is selected in case of attack, we are restricting network functionality until the response is deactivated. (ii) The second approach that we propose for future research is not to consider G in the response cost formula, and instead start with a poor response when the response system decides to deactivate all the applied responses. It does not matter if a poor response is applied, because in this case the risk level slips under the threshold, based on the response Goodness, and brings us very close to the threshold again. This approach has two important benefits. The first is that all the 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 users 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.

3.5 Dataset

Almost all the security works are based on the very old Datasets [35], [37]. Their accuracy and ability to reflect real-world conditions has been a major concern. 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 is yet to exist [36]. In order to better test and optimize the selection of these parameters, and compare with other IRS systems, it would be interesting to assemble a large dataset of recent attacks. However, this dataset of attacks would need to be executable and include the attacking and attacked systems images (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 preparing 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 aspect of the intrusion response systems in real-time in the large environments. Shiravi et al. [36] proposed a set of guidelines to outline valid datasets, which set the basis for generating profiles.

4 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. Fals 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 MSERC RDC to GSTC project and by CRC on Sustainable Smart Eco-Cloud.

REFERENCES

[1] A. Gehani and G. Kedem, "Rheostat: Real-time risk management," In Recent Advances in Intrusion Detection: 7th International Symposium, (RAID 2004), pp. 296-314, France, 2004.

[2] C. Strasburg, N. Stakhanova, S. Basu, and J. S. Wong, "A Framework for Cost Sensitive Assessment of Intrusion Response Selection," *Proceedings of IEEE Computer Software and Applications Conference*, pp. 355-360, 2009.

[3] N. Stakhanova, S. Basu and J. Wong, "A cost-sensitive model for preemptive intrusion response systems," *Proceedings of the 21st International Conference on Advanced Networking and Applications*, IEEE Computer Society, Washington, DC, USA, pp. 428-435, 2007.

[4] P. Ammann, D. Wijesekera, and S. Kaushik, "Scalable, Graph-Based Network Vulnerability Analysis," Proceedings of 9th ACM Conference on Computer and Communications Security (ACM-CCS 2002), pp. 217-224, 2002.

[5] S. Jha, O. Sheyner, and J. Wing, "Two formal analyses of attack graphs," *Proceedings of the 15th Computer Security Foundation Workshop*, June 2002.

[6] M. Jahnke, C. Thul, and P. Martini, "Graph-based Metrics for Intrusion Response Measures in Computer Networks," *Proceedings of the 3rd LCN Workshop on Network Security. Held in conjunction with the 32nd IEEE Conference on Local Computer Networks (LCN)*, pp. 1035-1042, Dublin, Ireland, 2007.

[7] W. Kanoun, N. Cuppens-Boulahia, F. Cuppens, and J. Araujo, "Automated reaction based on risk analysis and attackers skills in intrusion detection systems," *Third International Conference on Risks and Security of Internet and Systems*, pp. 117-124, 2008.

[8] H. Wei, D. Frinke, O. Carter, and C. Ritter, "Cost-benet analysis for network intrusion detection systems," CSI 28th Annual Computer Security Conference, Washington, DC, 2001.

[9] C. P. Mu, X. J. Li, H. K. Huang, and S. F. Tian, "Online risk assessment of intrusion scenarios using D-S evidence theory," *Proceedings of the 13th European Symposium on Research in Computer Security*, pp. 35-48, Malaga, Spain, 2008.

[10] W. Kanoun, N. Cuppens-Boulahia, F. Cuppens, and S. Dubus, "Risk-Aware Framework for Activating and Deactivating Policy-Based Response," *Proceedings of the Fourth International Conference on Network and System Security*, pp. 207-215, 2010.

[11] C. P. Mu and Y. Li, "An intrusion response decision-making model based on hierarchical task network planning," *Expert systems with applications*, vol. 37, no. 3, 2010, pp. 2465-2472.

[12] G. N. Matni and M. Dagenais, "Operating system level trace analysis for automated problem identification," *The Open Cybernetics and Systemics Journal*, vol. 5, 2011, pp. 45-52.

[13] N. Stakhanova, S. Basu, and J. Wong, "Taxonomy of Intrusion Response Systems," *Journal of Information and Computer Security*, vol. 1, no. 2, 2007, pp. 169-184.

[14] A. Shameli-Sendi, N. Ezzati-Jivan, M. Jabbarifar, and M. Dagenais, "Intrusion Response Systems: Survey and Taxonomy," *International Journal of Computer Science and Network Security*, vol. 12, no. 1, pp. 1-14, 2012.

[15] F. Cuppens and R. Ortalo, "Lambda: A language to model a database for detection of attacks," *Proceedings of the Third International Workshop on Recent Advances in Intrusion Detection (RAID2000)*, pp. 197-216 Toulouse, France, 2000.

[16] K. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems," Technical report, *NIST: National Institute of Standards and Technology*, U.S. Department of Commerce, 2007.

[17] G. Stein, C. Bing, A. S. Wu, and K. A. Hua, "Decision Tree Classifier For Network Intrusion Detection With GA-based Feature Selection," *Proceedings of the 43rd annual Southeast regional conference, Georgia*, ISBN:1-59593-059-0, pp. 136-141, 2005.

[18] N. B. Anuar, H. Sallehudin, A. Gani, and O. Zakaria, "Identifying False Alarm for Network Intrusion Detection System Using Hybrid Data Mining and Decision Tree," *Malaysian Journal of Computer Science*, ISSN 0127-9084, 2008, pp. 110-115.

[19] A. Lazarevic, L. Ertz, V. Kumar, A. Ozgur, and J. Srivastava, "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection," *Proceedings of the Third SIAM International Conference on Data Mining*, 2003.

[20] F. Xiao, S. Jin, and X. Li, "A Novel Data Mining-Based Method for Alert Reduction and Analysis," *Journal of Networks*, vol. 5, no. 1, 2010, pp. 88-97.

[21] P. Berkhin, "Survey of clustering data mining techniques," 2001.

[22] A. O. Adetunmbi, S. O. Falaki, O. S. Adewale, and B. K. Alese, "Network Intrusion Detection based on Rough Set and k-Nearest Neighbour," *International Journal of Computing and ICT Research*, vol. 2, no. 1, 2008, pp. 60-66.

[23] J. Han and M. Kamber, "Data Mining: Concepts and Techniques," 2nd ed., *San Francisco: Elsevier*, 2006.

[24] The Snort Project, Snort users manual 2.8.5, 2009.

[25] 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.

[26] M. F. Yusof, "Automated Signature Generation of Network Attacks," B.Sc. thesis, University Teknologi Malasia, 2009.

[27] H. Debar, D. Curry, and B. Feinstein, "The Intrusion Detection Message Exchange Format (IDMEF)," http://www.ietf.org/rfc/rfc4765.txt.

[28] W. Lee, W. Fan, and M. Miller, "Toward Cost-Sensitive Modeling for Intrusion Detection and Response," *Journal of Computer Security*, vol. 10, no. 1, 2002, pp. 5-22.

[29] K. Haslum, A. Abraham, and S. Knapskog, "DIPS: A framework for distributed intrusion prediction and prevention using hidden markov models and online fuzzy risk assessment," *Proceedings of the 3rd International Symposium on Information Assurance and Security*, pp. 183-188, Manchester, United Kingdom, 2007.

[30] M. Sabhnani and G. Serpen, "Formulation of a Heuristic Rule for Misuse and Anomaly Detection for U2R Attacks in Solaris Operating System Environment," *In Security and Management*, pp. 390-396, 2003.

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

[32] G. White, E. Fisch, and U. Pooch "Cooperating security managers: a peer-based intrusion detection system," *IEEE Network*, vol. 10, 1996, pp. 20-23.

[33] P. Porras and P. Neumann, "EMERALD: event monitoring enabling responses to anomalous live disturbances," *National Information Systems Security Conference*, pp. 353-365, 1997.

[34] T. Toth and C. Kregel, "Evaluating the impact of automated intrusion response mechanisms," *Proceedings of the 18th Annual Computer Security Applications Conference*, Los Alamitos, USA, 2002.

[35] MIT Lincoln Laboratory, 2000 darpa intrusion detection scenario specific data sets, 2000.

[36] A. Shiravi, H. Shiravi M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," Computers & Security, vol. 31, no, 3, pp. 357-374, 2012.

[37] University of California. KDD Cup 1999 data, http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[38] M. Gaber, A. Zaslavsky, and S. Krishnaswamy, "Mining Data Streams: A Review," *ACM SIGMOD Record*, vol. 34, 2005.

[39] C. Aggarwal, J. Han, J. Wang, and P. Yu, "A Framework for Projected Clustering of High Dimensional Data Streams," *Proceedings of the 30th VLDB Conference*, Toronto, pp. 852-863, Canada, 2004.

[40] C. Strasburg, N. Stakhanova, S. Basu, and J. S. Wong, "The Methodology for Evaluating Response Cost for Intrusion Response Systems," Technical Report 08-12, Iowa State University.

[41] N. B. Anuar, M. Papadaki, S. Furnell, and N. Clarke, "An investigation and survey of response options for intrusion response systems," *Information Security for South Africa*, pp. 1-8, 2010.

[42] L. Feng , W. Wang , L. Zhu, and Y. Zhang, "Predicting intrusion goal using dynamic Bayesian network with transfer probability estimation," *Journal of Networks and Computer Applications*, vol. 32, no. 3, 2009, pp. 721-732.

[43] D. Yu and D. Frincke, "Improving the quality of alerts and predicting intruder's next goal with Hidden Colored Petri-Net," *Computer Networks*, pp. 632-654, 2007.

[44] A. Shameli-Sendi, M. Dagenais, M. Jabbarifar, and M. Couture "Real Time Intrusion Prediction based on improving the priority of alerts with Hidden Markov Model," *Journal of Networks*, vol. 7, no. 2, February 2012, pp. 311-321.

[45] Z. Li, Z. Lei, L. Wang, and D. Li, "Assessing attack threat by the probability of following attacks," *Proceedings of the International Conference on Networking*, Architecture, and Storage, IEEE, pp. 91-100, 2007.

[46] P. Arnes, F. Valeur, and R. Kemmerer, "Using hidden markov models to evaluate the risk of intrusions," *Proceedings of the 9th international conference on Recent Advances in Intrusion Detection*, pp. 145-164, Hamburg, Germany, 2006.

[47] A. Arnes, K. Sallhammar, K. Haslum, T. Brekne, M. Moe, and S. Knapskog, "Real-time risk assessment with network sensors and intrusion detection systems," *In Computational Intelligence and Security*, vol. 3802 of Lecture Notes in Computer Science, pp. 388-397, 2005.

[48] N. Kheir, N. Cuppens-Boulahia, F. Cuppens, and H. Debar, "A service dependency model for cost sensitive intrusion response," *Proceedings of the 15th European Conference on Research in Computer Security*, pp. 626-642, 2010.

[49] N. Kheir, H. Debar, N. Cuppens-Boulahia, F. Cuppens, and J. Viinikka, "Cost evaluation for intrusion response using dependency graphs," In IFIP International Conference on Network and Service Security, 2009.

[50] I. Balepin, S. Maltsev, J. Rowe, and K. Levitt "Using

specification-based intrusion detection for automated response," *Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection*, pp. 136-154, 2003.

[51] Y. Chen, B. Boehm, and L. Sheppard, "Value Driven Security Threat Modeling Based on Attack Path Analysis," *In 40th Hawaii International Conference on System Sciences*, Big Island, Hawaii, January 2007.

[52] Y. Zhang, X. Fan, Y. Wang, and Z. Xue, "Attack grammar: A new approach to modeling and analyzing network attack sequences," *Proceedings of the Annual Computer Security Applications Conference (ACSAC 2008)*, pp. 215-224, 2008.

[53] S. Savage, D. Wetherall, A. Karlin and T. Anderson "Practical network support for IP traceback," *In ACM SIGCOMM*, pp. 295-306, August 2000.

[54] E. Fisch, "A Taxonomy and Implementation of Automated Responses to Intrusive Behavior," Ph.D. thesis, Texas A&M University, 1996.

[55] T. Bowen, D. Chee, M. Segal, R. Sekar, T. Shanbhag, and P. Uppuluri, "Building survivable systems: an integrated approach based on intrusion detection and damage containment," *In DARPA Information Survivability Conference and Exposition*, pp. 84-99, 2000.

[56] S. Musman and P. Flesher, "System or security managers adaptive response tool," *In DARPA Information Survivability Conference and Exposition*, pp. 56-68, 2000.

[57] A. Somayaji and S. Forrest, "Automated response using system-call delay," *Proceedings of the 9th USENIX Security Symposium*, pp.185-198, 2000.

[58] Y.M. Chen and Y. Yang, "Policy management for network-based intrusion detection and prevention," In IEEE Network Operations and Management Symposium, 2004.

[59] C. Carver and U. Pooch, "An intrusion response taxonomy and its role in automatic intrusion response," *IEEE Workshop on Information Assurance and Security*, 2000.

[60] C. Carver, J. M. Hill, and J. R. Surdu, "A methodology for using intelligent agents to provide automated intrusion response," *IEEE Systems, Man, and Cybernetics Information Assurance and Security Workshop*, pp. 110-116, 2000.

[61] D. Ragsdale, C. Carver, J. Humphries, and U. Pooch, "Adaptation techniques for intrusion detection and intrusion response system," *IEEE International Conference on Systems, Man, and Cybernetics*, pp. 2344-2349, 2000.

[62] S. M. Lewandowski, D. J. V. Hook, G. C. OLeary, J. W. Haines, and M. L. Rossey, "SARA: Survivable autonomic response architecture," *In DARPA Information Survivability Conference and Exposition*, pp. 77-88, 2001.

[63] D. Schnackenberg, H. Holliday, R. Smith, K. Djadhandari, and D. Sterne, "Cooperative intrusion traceback and response architecture citra," *In IEEE DARPA Information Survivability Conference and Exposition*, pp. 56-68, 2001.

[64] X. Wang, D. S. Reeves, and S. F. Wu, "Tracing based active intrusion response," *Journal of Information Warefare*, vol. 1, 2001, pp. 50-61.

[65] S. Tanachaiwiwat, K. Hwang, and Y. Chen, "Adaptive Intrusion Response to Minimize Risk over Multiple Network Attacks," *ACM Transactions on Information and System Security*, 2002, pp. 1-30.

[66] B. Foo, Y. S. Wu, Y. C. Mao, S. Bagchi, and E. Spafford, "ADEPTS: adaptive intrusion response using attack graphs in an e-commerce environment," *International Conference on Dependable Systems and Networks*, pp. 508-517, 2005.

[67] M. Papadaki and S. M. Furnell, "Achieving automated intrusion response: a prototype implementation," *Information Management and Computer Security*, vol. 14, no. 3, 2006, pp. 235-251.

[68] K. Haslum, M. E. G. Moe, and S. J. Knapskog, "Real-time intrusion prevention and security analysis of networks using HMMs," *In 33rd IEEE Conference on Local Computer Networks*, pp. 927-934, Montreal, Canada, 2008.

[69] L. Wang, T. Islam, T. Long, A. Singhal, and S. Jajodia, "An attack graph-based probabilistic security metric," *Proceedings of The 22nd Annual IFIP WG 11.3 Working Conference on Data and Applications Security (DBSEC08)*, 2008.

[70] S. Noel and S. Jajodia, "Understanding complex network attack graphs through clustered adjacency matrices, "*Proceedings of the 21st Annual Computer Security Conference (ACSAC)*, pp. 160-169, 2005.

[71] L. Wang, A. Liu, and S. Jajodia, "Using Attack Graph for Correlating, Hypothesizing, and Predicting Intrusion Alerts," *Computer Communications*, vol. 29, no. 15, 2006, pp. 2917-2933.

[72] R. Dantu, K. Loper, and P. Kolan, "Risk Management Using Behavior Based Attack Graphs," *Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)*, pp. 445-449, 2004.

[73] K. Haslum, A. Abraham, and S. Knapskog, "Fuzzy Online Risk Assessment for Distributed Intrusion Prediction and Prevention Systems," *Tenth International Conference on Computer Modeling and Simulation*, IEEE Computer Society Press, pp. 216-223, Cambridge, 2008.

[74] W. Kanoun, N. Cuppens-Boulahia, F. Cuppens, and F. Autrel, "Advanced reaction using risk assessment in intrusion detection systems," *Proceedings of the Second international conference on Critical Information Infrastructures Security*, PP. 58-70., Spain, 2007.

[75] E. Totel, B. Vivinis, and L. Mé, "A language driven intrusion detection system for event and alert correlation," *Proceedings at the 19th IFIP International Information Security Conference*, Kluwer Academic, Toulouse, pp. 209-224, 2004.

[76] J. Goubault-Larrec, "An introduction to logweaver," Technical report, LSV, 2001.