Towards automated incident handling: How to select an appropriate response against a network-based attack?

Sven Ossenbühl*, Jessica Steinberger*† and Harald Baier*

*da/sec - Biometrics and Internet Security Research Group University of Applied Sciences Darmstadt, Darmstadt, Germany Email:{Sven.Ossenbuehl, Jessica.Steinberger, Harald.Baier}@h-da.de

†Design and Analysis of Communication Systems (DACS), University of Twente, Enschede, The Netherlands Email:{J.Steinberger}@utwente.nl

Abstract

The increasing amount of network-based attacks evolved to one of the top concerns responsible for network infrastructure and service outages. In order to counteract these threats, computer networks are monitored to detect malicious traffic and initiate suitable reactions. However, initiating a suitable reaction is a process of selecting an appropriate response related to the identified network-based attack. The process of selecting a response requires to take into account the economics of an reaction e.g., risks and benefits. The literature describes several response selection models, but they are not widely adopted. In addition, these models and their evaluation are often not reproducible due to closed testing data. In this paper, we introduce a new response selection model, called REASSESS, that allows to mitigate network-based attacks by incorporating an intuitive response selection process that evaluates negative and positive impacts associated with each countermeasure. We compare REASSESS with the response selection models of IE-IRS, ADEPTS, CS-IRS, and TVA and show that REASSESS is able to select the most appropriate response to an attack in consideration of the positive and negative impacts and thus reduces the effects caused by an network-based attack. Further, we show that REASSESS is aligned to the NIST incident life cycle. We expect REASSESS to help organizations to select the most appropriate response measure against a detected network-based attack, and hence contribute to mitigate them.

Index Terms

cyber security; intrusion response systems; network security; automatic mitigation.

I. Introduction

Frequent occurrences of network-based attacks pose serious threats to the Internet, which has become the crucial infrastructure nowadays [1], [2]. Such attacks are feasible due to the historically evolved architecture of the Internet [3], [4]. For instance, many of the protocols in the TCP/IP stack do not provide security mechanisms to authenticate the source or destination of a network traffic packet. This absence of authentication enables to spoof the source address which is used in many attacks [5], [6].

Recent statistics show that reported network-based attacks are becoming larger, more sophisticated (e.g., multi-vector attacks), and getting more frequent [7]–[9]. According to a press release in February 2014, 39% of German organizations became victims of cyber attacks [10]. Furthermore, 21% are victims of cyber espionage accumulating the loss of 50 billion Euros [10]. However, these reports can only show a part of this magnitude. According to the Federal Office for Information Security (BSI) only 25% of incidents are communicated to external companies for help to recover normal operation. The number of reported incidents is even less [11]. At the same time, in-depth technical knowledge is no longer required to launch a network-based attack [12]. The reason is that launching these attacks have evolved as a business model [13], [14].

One approach to counteract the proliferation of network-based attacks is a generalized and automated process that initiates mitigation and response measures. Traditionally, automated mitigation and response processes make use of a Intrusion Response System (IRS) that provides a distinct response selection process [15], and is able to collaborate with other security appliances, such as firewalls to block and terminate suspicious traffic [8]. Although IRSs are promising systems, available solutions proposed by the scientific community are not widely adopted [16]. Further, each IRS uses different metrics to select an appropriate response and some of the metrics are only applicable for specific system environments. Moreover, most of the previous work is not reproducible due to closed testing data.

To overcome the shortcomings of closed source and system dependency, this paper presents a reaction strategy for response selection based on effectiveness assessment. The contribution REASSESS brings to the state of the art is that it is aligned to the National Institute of Standards and Technology (NIST) incident life cycle and it evaluates negative and positive impacts associated with the responses deployment to a given attack. In addition, the impact on the network without a response is also taken into account.

The remainder of this paper is organized as follows: Section II describes the context of our work. The requirements, derived from the context of this work, are specified within Section III. Section IV covers the related work by presenting an overview of published response selection models. In Section V, we introduce the concept of the reaction strategy model REASSESS, its alignment to the NIST incident life cycle and describe its implementation. In Section VI we analyze and evaluate the reaction strategy models. Finally, we conclude the paper and give an outlook on future work in Section VII.

II. Scenario

In this section, we describe the main focus of this paper. First, we describe the network in which we are going to place REASSESS. Second, we present how an alert occurs and initiates mitigation and response procedures within the IRS. The primary focus of this work are networks consisting of individual end hosts. Within these networks intrusion detection systems (IDSs), intrusion prevention systems (IPSs) and IRSs are often placed to protect the network against malicious traffic. The IDS sensor could be placed inline or in a promiscuous mode.

Figure 1 illustrates a simplified view of a network topology containing an IDS sensor in promiscuous mode. The advantage of an IDS in promiscuous mode is that it prevents forwarding delays or impacts caused by sensor failing. The traffic flow within a network with an IDS in promiscuos mode is shown in Figure 1. After the traffic enters the network (1) and passes the firewall, the traffic enters the demilitarized zone (DMZ). When the traffic arrives at the DMZ, copies of the packets are send to the IDS sensor (2). Therefore, IDS sensors in promiscuous mode analyzes the network traffic based on copies of the original packets and raises alerts (3) in case of detected malicious traffic. However, because the original packet is already on its way towards the destination, the IDS by itself cannot prevent that the original packet makes its way onwards its destination. Hence, the IDS is detecting the attack, but not preventing.

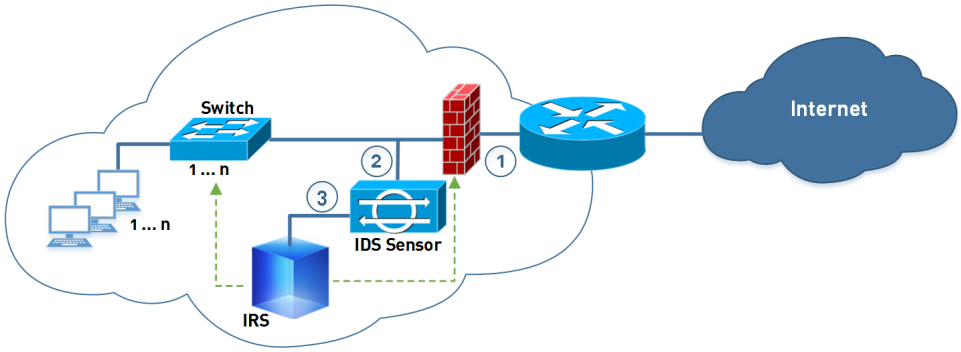


Fig. 1. Simplified network topology of a IDS in promiscous mode

In contrast to IDSs in a promiscuous mode, inline IDS sensors can be placed within the network as shown in Figure 2. Any traffic traversing the network (1) is forced to go in one physical or logical port on the sensor (2) and thus this topology adds a small delay before forwarding. In case of a benign packet, the IDS sensor forwards the packet through another logical or physical interface and continues its journey towards its destination (3). Inline IDS sensors are able to perform an automatic deployment of countermeasures e.g., drop a packet and deny that packet from reaching its final destination. Therefore inline IDSs are also called intrusion prevention systems. However, IPSs do not employ a distinct response selection process and thus rely on simple mappings of attacks to predefined responses [17].

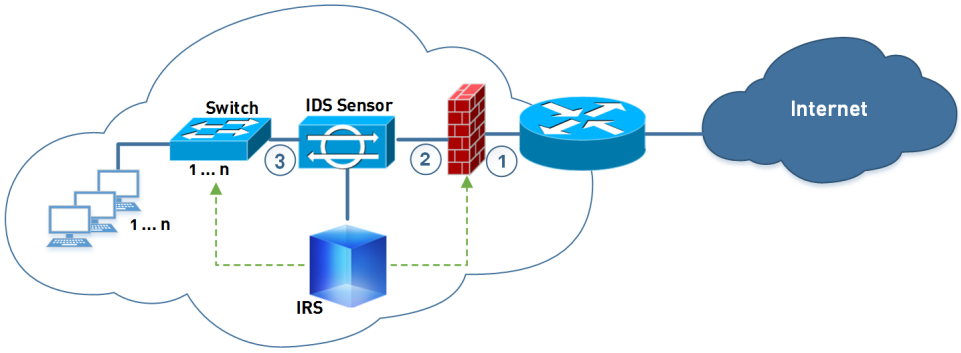


Fig. 2. Simplified network topology of a IDS in inline mode (IPS)

To overcome these shortcomings that just simple response actions or even no action can be taken, IRSs have been developed. IRSs provide the capability to select an appropriate response and perform both, automatic mitigation and response, to defend the network against malicious traffic. IRSs are located next to an IDS sensor as shown in Figure 1 and Figure 2. This work focuses on the reaction process deployed within an IRS. In particular, the response selection model within this reaction process that is initiated by an alert raised by IDSs. Therefore the type of IDS used within the network is not important.

III. Requirements and Assumptions

In this section, we define the requirements that an response selection model within an IRS should fulfill, as they emerged by the scenario described in Section II and defined in [18]. In the following, we will use these requirements to evaluate the response selection models and our proposed model REASSESS. In addition, we describe our assumptions to ensure that the work is not biased by tasks related to detection technologies.

A. Requirements

Automatic deployment: The selection of an appropriate response should be performed automatically to reduce delays caused by human interactions. Besides the reduction of delays caused by human interaction, the automatic deployment supports the timeliness of initial mitigation and response procedures, and the reduction of expert knowledge to choose an appropriate response.

Scalability: The response selection model should be able to cope with different network topologies e.g., a single network, distributed networks. Further, the response selection model should provide the capability to be placed in different network sizes and thus able to handle a different amount of alerts e.g., mid-size networks, backbone networks.

Adaptability: Due to the ever-changing nature of attacks the response selection model should have the ability to dynamically adjust the response selection and thus suit a particular type of attack identified by a IDS. The response selection model should provide automatic learning capabilities. Further, the response selection model should take into account previous responses and their effectiveness and thus make use of a knowledge base of previous responses.

System Independency: The response selection model should be used within IRSs of different vendors, communities etc. It should be able to handle different sources of alerts that have been detected by IDSs that work with flow data e.g., NetFlow, IPFIX, but also IDSs that work with raw packets e.g., Snort. Further, the response selection model should ensure the integratability with other security tools. Therefore it should initiate responses that interact with other security tools e.g., firewall, router.

Calculation Efficiency: A fast and efficient calculation is necessary in order to reduce the time window of an ongoing attack and thus reduce potential damages. Consequently, the identification of an appropriate response to a given alert should be calculated quickly.

Usability: The response selection model should provide a minimum number of configuration input parameters to reduce configuration efforts and the required expert knowledge to use the response model within an IRS. Network operators should be able to understand the process of selecting an appropriate response and performing the selection manually.

Security mechanisms: The selection of an appropriate response to a given attack often includes sensitive data (e.g., raw data, analyzed information of incident handling and its remediation). Due to this reason, the calculation process of the response selection model should prevent unauthorized access to this mitigation and response information to ensure that an attacker does not initiate a specific reaction with an attack. Further, the response selection model should ensure trustworthy origin, relevance and integrity of an alert.

B. Assumptions

This detection of malicious actions could be performed by monitoring technologies that classify malicious actions based on anomalies or signatures. Well-known signature based systems are Snort, Suricata [19], STAT or NetSTAT [20]. Current anomaly-based systems often use flow export technologies (e.g., Cisco NetFlow, IPFIX). Well-known anomaly-based systems using flow export technologies are BotTrack [21], BotFinder [22] and Disclosure [23]. Both, signature and anomaly-based systems rely on information that determine what is normal behavior and what is not normal. Due to the ever-changing nature of networks, applications, and malicious actions, false positives might be raised. However, the main focus of this article is the selection of an appropriate response after an malicious action was detected and thus the assumptions are as follows:

Aggregation: Each alert raised by a detection engine is treated as one attack. The aggregation and fusion of alarms should be taken into account by the detection system. This is assumption is in accordance to previous work [17], [24], [25].

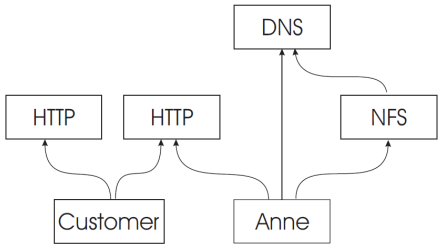
Confidence: We assume 100% confidence of the alerts. A detection system might raise false alerts but to ensure a hundred percent certainty of the alerts or a sanity check of the detection engine is out of scope of this work. This strong confidence is in accordance to [17], [24], [25].

IV. Response selection models

In this section, we present an overview of existing response selection models. For each model we elaborate its point of view and calculation methodology. Additionally, we describe mandatory input parameters and requirements of each model. Further, we analyze the response models with regard to their use-case context and to what type of attack a response is selected.

A. Evaluating the impact of automated intrusion response mechanisms

In the year 2002 researchers of the Technical University Vienna proposed a response selection model, hereafter referred to as IE-IRS [18]. IE-IRS is used to select different firewall configurations as a response to a given attack. These firewall configurations are evaluated with respect to their effectiveness. Therefore IE-IRS takes into account dependencies between network services offered by hosts, system users, network topologies and firewall rules [18]. These dependencies are modeled as a dependency tree and shown in Figure 3a.



(a) Dependency tree between the two users Anne and Customer of IE-IRS [18]

```
DNS is service at 132.100.98.11 53 udp
  { processName="bind";};

HTTP is service at 132.100.98.15 80,
  at 132.100.101.4 80
  { processName="httpd"; };

NFS is service at 132.100.100.4 2049
  { processName="nfsd"; };

anne is user at 132.100.100.27 { cost=5000; }
  requires (DNS at 132.100.98.11 53 udp 0.4)
   and
     ( NFS at 132.100.100.4 2049 0.4
       and  HTTP at 132.100.101.4 80 0.2 ));
```

(b) Modeling language of IE-IRS [18]

Fig. 3. IE-IRS dependency graph and its modeling language

Besides a dependency tree this response selection model utilizes a modeling language which is shown in Figure 3b. This modeling language consists of a Bison[1] grammar file as input and a fast lexical analyzer generator (FLEX[2]) file. The final output file contains C source code. The modeling language is used to define the importance for each network service as shown in Figure 3b. The importance for each network service has to be defined through a user or customer that uses this service. Further, the importance of a service is defined in terms of costs related with the degradation or termination of the service.

In order to select a meaningful response, the degradation of the working capability and penalty costs are determined. However, the authors declare the penalty costs as a constant value. In addition, the authors stated that IE-IRS is able to terminate processes and disable user accounts. As an example, IE-IRS has been used to mitigate and react on a Denial of Service (DoS) attack against a web server. To select a response the response selection algorithm takes into account penalty costs of a resource being unavailable and the capability of a resource. The capability is determined by using a depth-first search without a cyclic behavior. Therefore the authors introduce a capability value $c(e) \in [0..1]$ of an entity e. This capability value indicates the performance of an entity if the specified response strategy is triggered, compared to the situation when all resources are available. Let entity e be not dependent on any other entities, then the capability of $c(e) = 0$. In case of $c(e) = 0$, the entity does not provide any service to other system resources. In contrast, $c(e) = 1$ implies that the entity provides service to others. The capability reduction is defined as $cr(e) = 1 - c(e) \in [0..1]$. The penalty costs $p(e)$ for an entity e is a value representing the cost when e becomes unavailable. The penalty costs of a network service is calculated using the following equation: $p(e) = cr(e) \cdot penalty$ where the $penalty$ is a user-defined constant that reflects the importance of an entity. The response with the lowest penalty

[1]http://www.gnu.org/software/bison/
[2]http://flex.sourceforge.net/

cost has the least negative impact on the system and is selected to be deployed to a given intrusion.

B. Adaptive Intrusion Response using Attack Graph

The Adaptive Intrusion Response using Attack Graph (ADEPTS) [26] approach was published by researches of Purdue University in 2005. ADEPTS mainly considers host attacks (e.g., buffer overflows or privilege escalation) and was built to show that several intermediate attack goals need to be accomplished by an adversary to achieve the attack. Thus ADEPTS employs attack graphs to identify the actions required to achieve possible attack targets in a distributed system. This attack graph of ADEPTS is an Directed Acyclic Graph (DAG) [27] and used to show the objectives of suitable responses. Attacker goals are expressed as end states in the attack graph with intermediate steps leading to the fulfillment of those goals. Such an attack graph is shown in Figure 4.

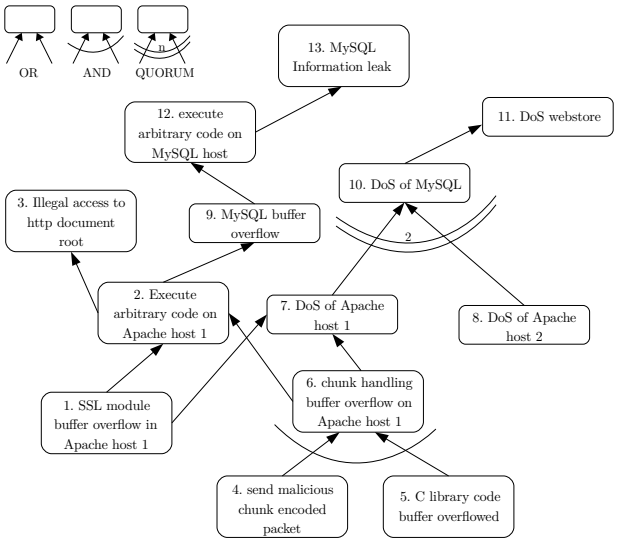


Fig. 4. A sample attack graph of ADEPTS [26]

The edges in the intrusion graph are categorized into the three types: OR, AND, and QUORUM. For a node with incoming OR edges, an intruder has to achieve at least one of its child nodes. In contrast, for AND edges, all the child nodes have to be achieved. For QUORUM edges a Minimum Required Quorum (MRQ) is assigned to it. This number represents the minimum number of child nodes in which goals need to be achieved in order for the node to be accomplished. For instance, node ten in Figure 4 has an MRQ of two. If two database servers are successfully under attack by a Distributed Denial of Service (DDoS) attack, the database is no longer usable for the system in a satisfying manner.

The intrusion graph has to be updated when changes to the system configuration or new known vulnerabilities related to the system architecture occur. Thus vulnerability information have be obtained, preprocessed or tailored to the specific environment in a regular manner. To update the intrusion graph, a semi automated method called Portable Intrusion Graph Generation (PIG) is employed. PIG requires the two inputs: vulnerability descriptions and system services description. Both, the vulnerability descriptions and system services description are system and environment dependent, and thus created manually.

The system service description is a directed graph, in which each node represents a service. The edges are considered as intrusion channels. These intrusion channels model how an intrusion can spread from a compromised node to the next node which is connected through the edge.

The vulnerability descriptions are obtained by querying common vulnerability databases, such as NIST's National Vulnerability Database (NVD)[3], the Bugtraq security database[4], the Open Source Vulnerability Database (OSVDB)[5], and the Common Vulnerabilities and Exposure (CVE) referencing standard. However, manually adjustments are also required in the generation algorithm. These manually adjustments cover the specification of the four characteristics: (i) name, (ii) affected services, (iii) manifestation, and (iv) dependent vulnerabilities and services.

The characteristic *Name* ensures human readability. *Affected services* represents an enumeration of services in the system service description affected by the vulnerability. The characteristic *Manifestation* describes the result of a successful exploitation.

[3]https://nvd.nist.gov/
[4]http://www.securityfocus.com/vulnerabilities
[5]http://osvdb.org/

Possible outcomes are considered as leaking of information, execution of arbitrary code, incorrect behavior of service, DoS, and service termination. *Dependent vulnerabilities* denotes the dependence on other vulnerabilities and services that have to be compromised to exploit this vulnerability.

The selection of a response is based on effectiveness to that particular attack in the past. Further, the selection takes into account the disruptiveness to legitimate users and a confidence level which indicates the probability that the attack is actually taking place. ADEPTS provides response measures to block IP addresses and host specific reactions such as rebooting the web server.

After a response has been deployed the system checks if the response was successful. If a response action is deployed on an edge that can be used to reach the attackers goal the effectiveness is decreased. After a certain amount of time or when an administrator deactivates a response the response is considered as successful if no further alerts were observed.

C. A Framework for Cost Sensitive Assessment of Intrusion Response Selection

In 2009 researchers of the Iowa State University proposed a framework to select the most appropriate response in balance of the potential damage of an attack and the costs for mitigation [24]. This approach is also described in several other publications of the authors [16], [17], [28] and will be referred as CS-IRS in the following.

The basic idea of CS-IRS is that response actions lead to positive and negative effects. CS-IRS is intended to minimize the damage of the attack and the negative effects of the response deployment. To achieve this, a set of measurements are introduced to characterize potential costs associated with the intrusion handling process in terms of the risk of potential intrusion damage, effectiveness of response action, and response cost for a system. In CS-IRS, response cost and intrusion cost are evaluated based on two factors: operational cost and impact on system resources. The assessment of system resources, response cost, and potential intrusions are modeled as static values. They are based on the expert judgment and are represented as values in the range $[0, 1]$. The final step is the response selection process at the time of intrusion. To estimate the response effectiveness, they calculate a response success rate against detected intrusions and a response coverage value. A success rate parameter (response goodness) is defined for each response. If the selected response succeeds in repelling the attack, its success factor is increased by 1. Otherwise, the failure factor of the response is increased.

CS-IRS requires the input of an IDS. In particular, CS-IRS expects an alert raised by an IDS. In addition, the model requires information about the systems security policy. The systems security policy contains a value between 0 and 1 that represents the importance for the overall system with respect to the security goals confidentiality, integrity, availability. CS-IRS also requires the enumeration of system resources with their importance to the system and definitions of known intrusions [17]. CS-IRS was designed to work as a host-based solution and thus provides response measures to reboot the system, block user accounts or restart a service. The authors of CS-IRS used attacks of the DARPA/Lincoln Lab offline evaluation data to evaluate CS-IRS and select an appropriate response to a given attack.

D. Topological Vulnerability Analysis

A preventive approach called Topological Vulnerability Analysis (TVA) was proposed by researchers of George Mason University in 2010 [29]. TVA is a modeling and simulation approach that relies on existing security tools to gather the required information. This information includes vulnerability information and network configuration. In addition, a continual monitoring of information sources regarding reported vulnerabilities is necessary to keep pace with emerging threats.

TVA combines vulnerability information from internal sources and external sources (e.g., NIST NVD, CVE, Symantec DeepSight). TVA also takes into account the attacker perspective to discover all attack paths within a network. As internal input, potential vulnerabilities an attacker might abuse and network configurations are used for generating an attack model. Network configuration data might include vulnerability scan reports and firewall rules. The exploits in the attack model are matched against the provided vulnerability information to predict multi step attacks. Using this information, an attack graph is generated similar to ADEPTS. In contrast to ADEPTS, the attack graph of TVA guides optimal strategies for preventing attacks, such as patching critical vulnerabilities and hardening systems and services. However, there usually remain some residual attack paths within a network. At this point, the attack graph provides the necessary context for dealing with intrusion attempts. For instance, the attack graph can guide the placement of intrusion detection sensors to cover all attack paths, while minimizing sensors redundancy. Finally, this graph is used to generate recommendations for vulnerability mitigation and computation of metrics to measure the overall network security.

V. REASSESS - Response Effectiveness Assessment

This section introduces our response selection model REASSESS (Response Effectiveness ASSESSment). We first explain its concept and its components, and then turn to its calculation methodology. Finally, we describe the use of REASSESS within three scenarios as proof of concept.

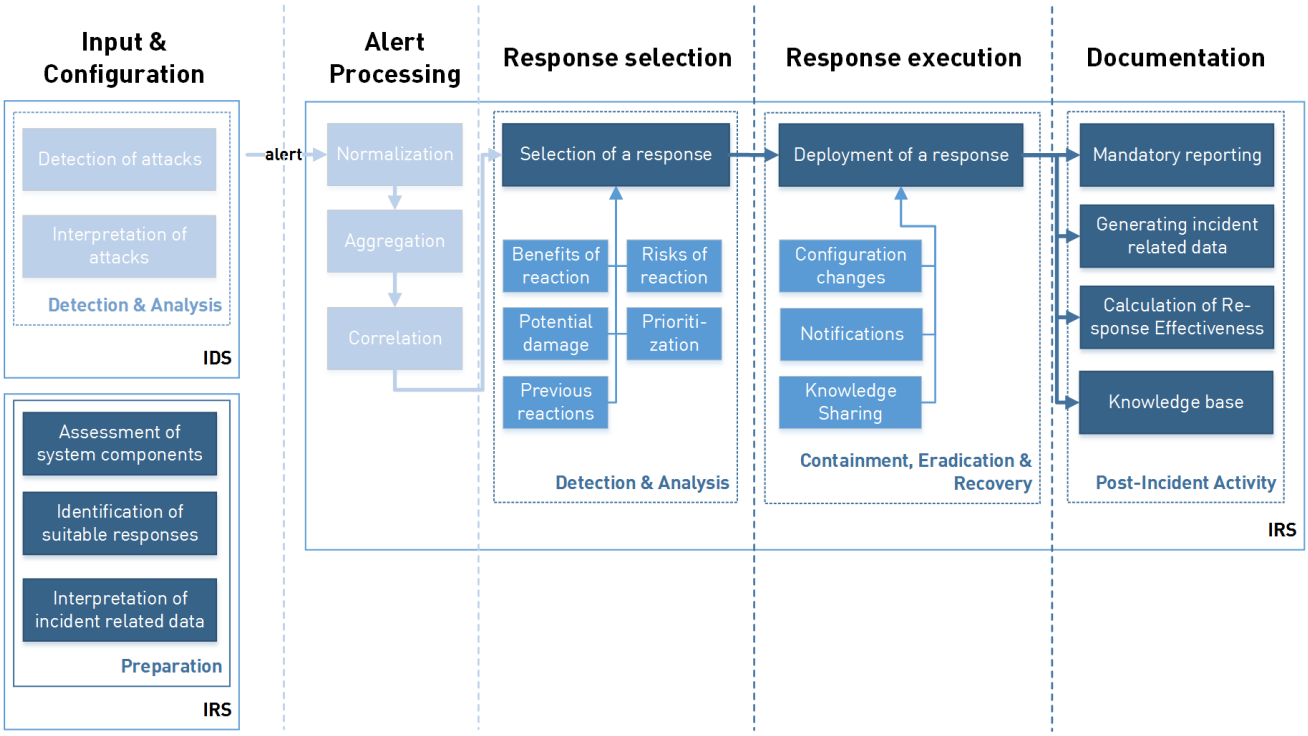


Fig. 5. Response selection process

A. *Reaction System Concept*

Our response selection model *Response Effectiveness Assessment* - REASSESS is based on CS-IRS [24] and IE-IRS [18]. In contrast to CS-IRS, REASSESS considers the negative effects of a response as an estimation of the degree of negatively affected legitimate service requests due to the response deployment. In addition, the negative impact of an attack without taking a responsive action is modeled with the usage of the priority of an alert. In contrast to the IE-IRS, REASSESS models penalty costs as Service Level Agreement (SLA) violation costs related to the importance of a provided service.

REASSESS consists of the five stages: Input & Configuration, alert processing, response selection, response execution and documentation, which are related to the NIST incident response cycle. The response selection process of REASSESS and its main components are shown in Figure 5.

Before REASSESS may be used in a productive environment, configuration steps need to be performed. These configuration steps are described within the Input & Configuration stage and the related NIST incident response cycle phase called preparation. Within this preparation phase, the importance of different system components are assessed e.g., assessment of provided services within the network. In addition, suitable responses that can be applied in the system environment are identified and defined. Besides the configuration of REASSESS, the input of an IDS is required and described withing the Detection & Analysis phase. Within this phase, an IDS performs the detection and interpretation of an attack. In case the IDS detects malicious traffic, it raises an alert that initiates the alert processing within an IRS. The alert processing stage consists of an alert normalization, aggregation and correlation. The alert processing stage is not covered by REASSESS, because of its own complexity. After completion of the alert processing stage, the response selection stage is initiated. The response selection stage is related to the Detection & Analysis phase of the NIST incident response cycle. The selection of a response takes into account the benefits and risks of a reaction, but also potential damages caused by the attack in case of no reaction. In addition, the selection of a response also looks up previous reactions and their effectiveness. Finally, possible responses are identified and prioritized. After the response selection stage, the response execution is started. The response execution stage is related to the Containment, Eradication & Recovery phase of the NIST incident response cycle. Within the response execution stage, configuration changes, notifications or knowledge sharing is performed. Finally, the documentation stage starts which is related to the Post-Incident-Activity of the NIST incident response cycle. Within the documentation stage, the alert and its response are reported, the response effectiveness is calculated and all alert related data is stored for later investigations within a knowledge base. The alignment of REASSESS to the NIST incident response cycle is shown in Figure 6 and thus ensures both reproducible and understandable results.

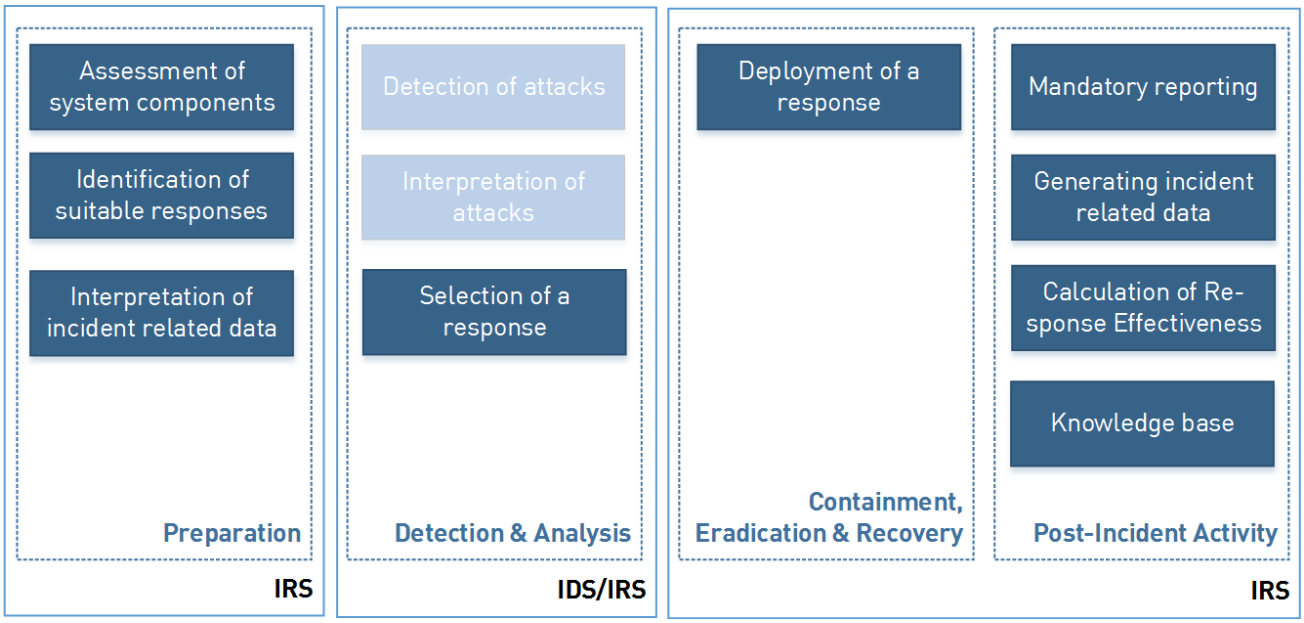


Fig. 6. The reaction strategy aligned with the NIST incident response cycle

B. Calculation Methodology

Traditionally, human experts are informed about the occurrence of an alert and begin to extract the necessary information. This information might include data about the targeted system component and the source of the potential attack. Afterwards the benefits and risks of each response are compared with potential damage of the attack and finally a response based on reasoning and experience is selected. The response selection model REASSESS facilitates this comparison and returns the most suitable response to a given alert. Table I provides the symbols used in the next sections.

Therefore the comparison is modeled within REASSESS with an effectiveness value E. Let $R = \{r_0, r_1, .., r_n\}$ be a set of appropriate responses r for the system environment, then the effectiveness for each response $E(r)$ is defined as in Equation (1)

$$E(r) = A_p - A_n \in [-1..1] \tag{1}$$

where $A_p \in [0, 1]$ are the positive effects and $A_n \in [0, 1]$ the negative effects of a response as explained in what follows. The model's aim is to select the response with the highest effectiveness E and execute it in case of a detected attack as presented in Equation (2).

$$\max(E) \Leftrightarrow \max(A_p - A_n) \in [-1..1] \tag{2}$$

In case of $E(r) = E(r')$ where $r = r'$ both responses are applicable. In the initial iteration of the response selection stage only negative effects are considered, thus $A_p = 0$. To model the negative effects of an attack or a response two factors are considered. First, the impact of the attack or response to the system and second, the importance of the attacked system component. Accordingly, the negative impact A_n can be determined as in Equation (3)

$$A_n = \frac{F_d(\epsilon) + S(\epsilon)}{2} \in [0..1] \tag{3}$$

where $S(\epsilon)$ denotes the importance of the service ϵ and $F_d(\epsilon)$ is the capability reduction. We define the capability value F to indicate the performance of an entity ϵ either due to a deployment of a response measure, or in the event of an attack. Thus, we are considering: $\exists r \in R$ that indicates no response. The capability reduction F_d is defined as in Equation (4)

$$F_d(\epsilon) = 1 - F(\epsilon) \in [0..1]. \tag{4}$$

We identified a casual loop between the priority of an incident and impact on available services. The more impact an incident has on normal services offered to users, the higher the priority assigned to the alert will be. The estimation of the alerts importance is done by IDSs. Hence, to model the case of an attack without a response, the capability reduction can be derived from the priority of the alert by dividing its priority value with the highest possible priority value. This division is

TABLE I
NOTATION TABLE.

Symbol	Description	Symbol	Description
ϵ	Entity (service)	α	Alert (Security event)
E	Effectiveness value	r	Response
$E(r)$	Effectiveness value for each response	$R = \{r_0, r_1, .., r_n\}$	Set of responses
A_p	Positives effects of a response	A_n	Negative effects of a response
$S(\epsilon)$	Importance of a service	$F_d(\epsilon)$	Capability reduction
D	Disruptive impact	A'	Set of alerts
$dps_{r_i}(a_j)$	Responses that successfully mitigate the attack	$dpt_{r_i}(a_j)$	Responses that do not successfully mitigate the attack

performed to normalize the priority of the alert to a value $\in [0 \dots 1]$. In order to model the negative impact of a response deployment, each available response has to have a disruptive impact assigned to it. We define $D \in \{none, low, medium, high\}$ as an estimation of the disruptive impact of negatively effected legitimate service requests due to the response deployment. Whereas $none = 0.0$, $low \in [0.01 \dots 0.33]$, $medium \in [0.331 \dots 0.66]$, and $high \in [0.66 \dots 1]$. For instance, isolating a service from the network would have a high impact and thus would have a capability reduction of $F_d(\epsilon) = 1 \in [0.661 \dots 1]$. The service importance can be derived from an SLA where penalty costs are associated with the degradation of the service. We identified a relation between the SLA defined penalty costs and the importance of the service. In theory, the higher SLA penalties have to be payed in case of unavailable services, the more important the service is in the system's context. The importance of a system component is defined by dividing the SLA costs of the incident by the highest SLA costs to normalize it to a value $S \in [0..1]$.

After a response is deployed, the positive effects are calculated. We define the positive impact A_p as in Equation (5)

$$A_p = rsr \in [0..1] \tag{5}$$

where rsr is the response success rate. Let $R = \{r_0, r_1, .., r_n\}$ be the set of responses and $A' = \{a_0, a_1, .., a_m\}$ the set of alerts, then $rsr(r_i, a_j)$ denotes the response success rate for a given response r_i and alert a_j. The rsr value is the degree of successful deployments of response r_i to counter an attack indicated by an alert a_j divided by the total number of deployments of r_i regardless of its success in countering attacks indicated by an alert a_j (cf. Equation 6)

$$rsr(r_i, a_j) = \frac{dps_{r_i}(a_j)}{dpt_{r_i}(a_j)} \in [0..1] \tag{6}$$

The number of deployments of response r_i in case of a_j that successfully mitigated the attack is denoted by $dps_{r_i}(a_j)$. The total number of deployments of response r_i in case of a_j without success in mitigating the attack is denoted by $dpt_{r_i}(a_j)$.

C. Proof of concept

To prove the concept of the response selection model REASSESS and its calculation methodology, a testbed is set up as shown in Figure 7. This testbed comprises two networks interconnected by the Internet. The IRS containing REASSESS is located in network A and is able to intercept ongoing data flows. The testbed involves malicious activity originating from the attacker residing in network B targeting the victim in network A. On each relevant network interface packet capture files of the attack are recorded.

With the aid of this testbed three different test scenarios are conducted to prove the performance, the correctness and the mitigation capabilities of REASSESS and can be described as follows:

Performance measurement: The performance of REASSESS is measured using different amounts of alerts (e.g., 125, 500, 2 000, 8 000). These alerts are passed to the IRS simultaneously in order to determine how fast REASSESS parses an alert, calculates and identifies an appropriate response and finally deploys this response.

Correctness: To verify the correctness of REASSESS, we launch a TCP SYN flooding attack that is detected by the IDS. Our response selection model REASSESS calculates, identifies and deploys the most suitable response. Afterwards the effects of this response are measured. In addition, we can evaluate if the calculation and identification of an appropriate response is fast enough to counter the attack.

Mitigation capabilities: The mitigation capabilities of REASSESS are tested using a bandwidth attack. This bandwidth attack consists of 200 000 network traffic packets with an attack duration of 13.41 seconds. The attack is launched from network B

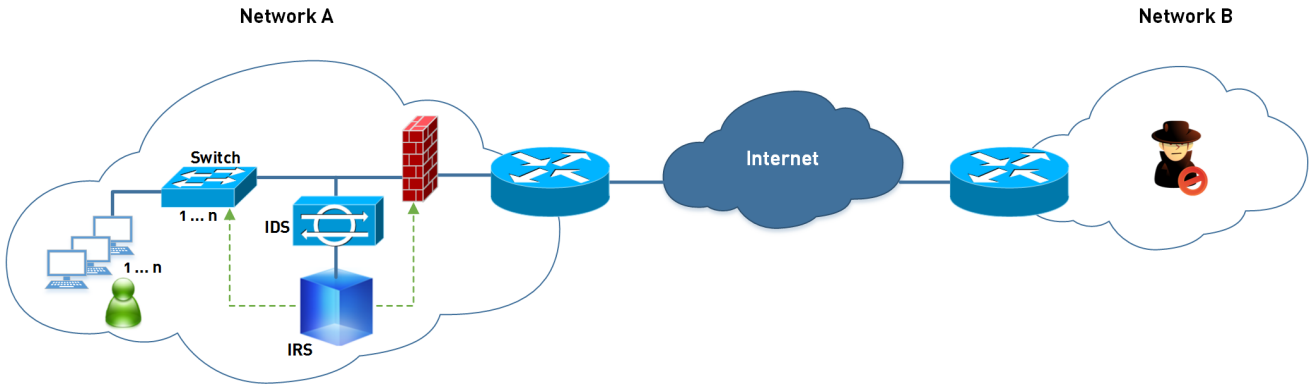


Fig. 7. Simplified view of the REASSESS testbed

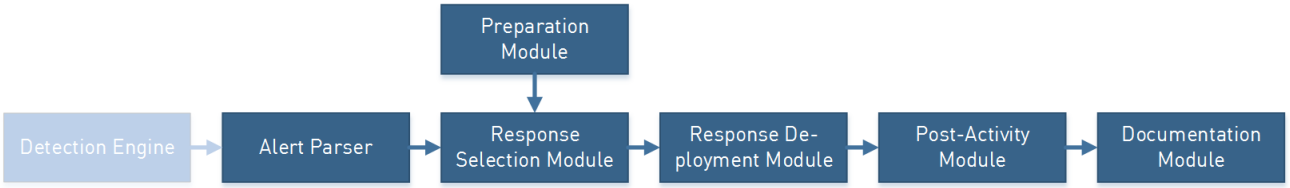


Fig. 8. Overview of the components of the REASSESS application

targeting network A as shown in Figure 7.

To perform these measurements in our three use-case scenarios, we developed an application that contains seven components as shown in Figure 8. The gray colored component represents the detection engine. In our application, we use the open-source IDS Snort. At a later point in time, the IDS Snort could also be replaced by other IDSs in order to keep the flexibility.

The *Preparation Module* parses the files that contain information about network services, and the disruptive impact of the response measures. This information is defined manually by a human expert within the preparation phase of REASSESS and is accessible to the *Response Selection Module* which utilize this information for its decision making process. The *Alert Parser* handles the alerts raised by the *Detection Engine* and extracts the necessary information for the *Response Selection Module*. Afterwards, the *Response Selection Module* identifies the attacked service on the basis of the information provided by the *Alert Parser* and *Preparation Module*. In addition, it calculates the effectiveness value E for each available response measure and selects the response with the highest E. Subsequently the *Response Deployment Module* gets instructed respectively which takes the required information and deploys the selected response by means of applying firewall rules. Each response is applied with a logging prefix to identify the discarded packets later. Thus, the success of the response can be evaluated by the *Post Activity Module*. Finally, the *Documentation Module* provides an interface to a database which includes the effectiveness of previously deployed responses. Additionally, the database may include incident related information.

The testbed of REASSESS was installed on three virtual machines interconnected by two virtual networks. Each virtual machine consists of one CPU of 2.5 GHz and 1 GB of RAM. The reaction system maintains two network interfaces (internal and external), enabling it to intercept ongoing data flows as shown in Figure 9. The IDS and REASSESS can be installed on the same physical device, but this is not a mandatory requirement. Besides the installation and configuration, one challenge is finding a suitable dataset for testing purposes. In contrast to IDSs, replaying public available dataset is not sufficient, because an IRS requires network traffic that can be replayed bidirectional [30]. This means that packets from the attacker should arrive on the IRS interface, and packets (and replies) from the target on another interface. In addition, configuration changes applied by the IRS need to be verified to ensure that the network components e.g., firewall are not misconfigured and still working properly. Thus, replaying recorded network data can not be used to evaluate the response deployment and mitigation capabilities of an IRS.

In our testbed, we identified and configured three suitable responses to defend the network: (i) no responsive action, (ii) rate-limiting a connection and (iii) IP traffic filtering. The response rate-limiting and IP traffic filtering make use of `iptables`[6] that provides the capability to throttle incoming connections and support basic firewall functionality. These responses are chosen

[6]http://netfilter.org/projects/iptables/index.html

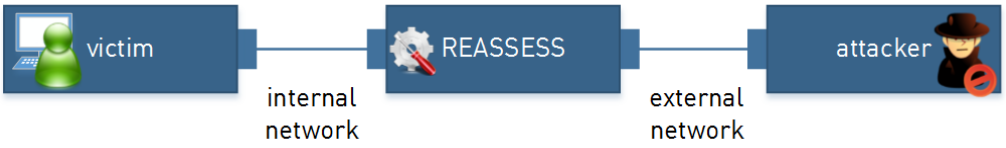


Fig. 9. Testbed for REASSESS application

in accordance to previous work [17], [25], [26]. However, our evaluation also validates the correctness of the response selection.

Performance measurement: The performance of REASSESS is evaluated using the three defined suitable responses: no responsive action, rate-limiting and IP traffic filtering. In different test steps a quantity of $125, 500, 2\,000, 8\,000$ alerts are generated and transmitted to the victims network. The IRS including REASSESS receives these alerts, calculates and identifies a suitable response. The results reveal that the runtime of the tests scale linearly as shown in Table II. Furthermore, the runtime of each test depends on the available CPU and RAM resources. Other processes running on the system can slow down the selection process significantly if they are consuming a large fraction of the CPU resources.

TABLE II
MEASURED RUNTIMES FOR DIFFERENT AMOUNT OF ALERTS.

	125 Alerts	**500 Alerts**	**2000 Alerts**	**8000 Alerts**
slowest runtime	13.147 sec.	50.353 sec.	195.386 sec.	838.901 sec.
fastest runtime	11.472 sec.	45.676 sec.	189.582 sec.	792.314 sec.
average runtime	11.957 sec.	48.550 sec.	192.435 sec.	804.555 sec.

Correctness: To verify the correctness of REASSESS, we assume that the DNS service of the victim network is target of an attack. Given $R = \{\text{none, rate-limit, blockIP}\}$, $S(\text{DNS}) = 1$, an alert α, $rsr(\text{rate-limit}, \alpha) = 0$, $rsr(\text{blockip}, \alpha) = 0$, $D(\text{rate-limit}) = F_d(\text{rate-limit}) = 0.66$, and $D(\text{blockIP}) = F_d(\text{blockIP}) = 0.2$. Let $\#$ denote the Iteration and $p(a)$ the priority p of the alert and let $p = 1$ be the lowest and $p = 3$ the highest priority. Then REASSESS calculates values as listed in Table III. The response blockIP was successful in every single test run. Thus, the rsr of response $blockIP$ changes to 1.0 and remains at this value.

TABLE III
CALCULATIONS WITHIN THE RESPONSE SELECTION MODULE.

#	$p(\alpha)$	r	$D(r)$	rsr	$A_n = (F_d(r) + S)/2$	$E = A_p - A_n$
1	3	none	1.0	0.0	$2.0/2 = 1.0$	-1.0
		rate-limit	0.66	0.0	$1.66/2 = 0.83$	-0.83
		blockIP	0.2	0.0	$1.2/2 = 0.6$	-0.6
1	2	none	0.66	0.0	$1.66/2 = 0.83$	-0.83
		rate-limit	0.66	0.0	$1.66/2 = 0.83$	-0.83
		blockIP	0.2	0.0	$1.2/2 = 0.6$	-0.6
1	1	none	0.33	0.0	$1.33/2 = 0.66$	-0.66
		rate-limit	0.66	0.0	$1.66/2 = 0.83$	-0.83
		blockIP	0.2	0.0	$1.2/2 = 0.6$	-0.6
2	1	none	0.33	0.0	$1.33/2 = 0.66$	-0.66
		rate-limit	0.66	0.0	$1.66/2 = 0.83$	-0.83
		blockIP	0.2	1.0	$1.2/2 = 0.6$	0.4

Beside the identification of the most suitable response, the effectiveness of a selected response needs to be verified. After REASSESS identified blockIP as the most suitable response, the response was deployed. The deployment of the response blockIP resulted in an additional rule added to the firewall. Further, the test results show that the ninth packet and tenth packet gets dropped due to the decision of the reaction system. Blocking is considered as successful if a logging entry is found that indicates the blocking of the IP address in one second difference compared to the timestamp of the alert. The firewall log file

reveals that the packets were logged and discarded.

Mitigation capabilities: The mitigation capabilities of REASSESS are evaluated by using a bandwidth attack that consists of 200 000 network traffic packets. One feasible way to verify that a replayed attack is blocked is to ensure that the packets of the attack are not received at the target [30]. Due to capturing packets transmitted through the attackers network interface, we observed that only 157 629 of the 200 000 packets are sent out by the attack tool and transmitted further towards the victim's network. During the attack, Snort raises 17 108 alerts and 15 438 packets are blocked due to the reaction to the alerts. Even though only 10.85 % of the packets traversing the interfaces were detected as an attack, 90.23% of them were blocked due to the decision of REASSESS.

VI. Evaluation

In this section, we evaluate the response selection models. First, we describe the characteristics of the evaluation criteria derived from the requirements presented in Section III. Furthermore, we introduce eight evaluation criteria for the response selection models. Last, we present and summarize our findings.

A. Evaluation methodology

The response selection models are evaluated based on the following eight criteria: automatic deployment, scalability, adaptability, system independency, calculation efficiency, usability, security mechanisms and evaluation methodology. These criteria were chosen to provide a means of comparison between the response selection models and to measure if and how the use of this response selection model is viable for network operators.

Each response selection model is evaluated using the following method. For each evaluation criteria one of the following scores is determined: low (-), medium (0), and high (+). All response models start with a score of 0. In case a low score is determined for a criterion, the sum will be decreased by 1, and increased by 1 in case of a high score respectively. A medium score is equal to 0 meaning that the sum will not be changed. In the best case a response model can have a sum of 8 meaning that all criteria are considered as high, in the worst case a response model will score with the sum of -8.

The ability to select and deploy an appropriate response automatically to reduce delays which result due to human intervention is described by the 'automatic deployment' criterion. This also includes the ability to evaluate the effectiveness of the deployed response. The criterion 'scalability' describes the ability handle different sizes of network topologies. An response selection model is called scalable when it is capable of working properly with small networks e.g., company networks but also large networks e.g., high-speed networks. 'Adaptability' defines the aptitude to suit different use-cases. A response selection model is called adaptable when it is capable to automatically learn from previous responses and their effectiveness and take them into account during a response selection. The ability to use the response selection model in various system environments is described by the criterion 'system independency'. The criterion 'calculation efficiency' describes how fast the calculation of an appropriate response is. The calculation of a response must be fast in order to respond in an appropriate time. 'Usability' describes whether a response selection model requires expert knowledge to provide input parameters. A response selection model is called usable when the input parameters and their calculation process do not require a system expert. The use of security mechanisms e.g., encryption or signature, within a response selection model is described by the criterion 'security mechanisms'. The criterion 'evaluation methodology' describes the reproducibility of the evaluation of a response selection model. Further, the criterion 'evaluation methodology' focuses on real world scenarios that show the practical applicability.

B. Evaluation results

In this section, we present and discuss the evaluation results of the response selection models according to the above described eight criteria.

Automatic deployment: IE-IRS is described to be able to automatically evaluate and re-evaluate responses in the system context and thus allows to determine the effects of different responses. IE-IRS does not describe the ability to make use of an automatic deployment of responses and thus result in low automatic deployment capability. ADEPTS is able to automatically deploy responses. The system uses a feedback mechanism which adjusts the effectiveness of the response based on whether the response was successful in preventing further attacks or spread of the current attack. After a certain amount of time or when an administrator deactivates a response the response is considered as successful if no further alerts were observed. Therefore the capability to automatically deploy an appropriate response of ADEPTS is considered as high. CS-IRS is also able to automatically deploy responses and thus the automatic deployment is considered as high. Even though a metric response success rate is explained, there is no explanation given how the response is evaluated to be successful or not. In contrast to ADEPTS and CS-IRS, TVA does not provide an automatic deployment of responses and thus score low in this criterion. Instead of providing an automatic deployment of responses, TVA automatically generates graphs which show vulnerability interdependencies to guide

an administrator in finding vulnerabilities and preventive measures to harden the network. REASSESS is able to automatically deploy responses and verify if they had the desired effect. The automatic deployment of responses within the REASSESS application is achieved by parsing certain log entries of the firewall for discarded packets due to the reaction decision of REASSESS. Therefore the capability to automatically deploy an appropriate response of REASSESS is considered as high.

Scalability: The calculation and descriptions of the IE-IRS dependency graph is considered as complex in mid-size or large-scale networks. The reason is that IE-IRS builts a dependency graph of network services offered by hosts, system users, network topologies and firewall rules. Therefore the scalability capability of IE-IRS is considered as low. Similar to IE-IRS, ADEPTS creates an attack graph, but ADEPTS focuses on specific system environments, namely distributed web-based environments. Therefore the attack graph compared to IE-IRS is quite small. However, the use of ADEPTS in mid-size or large scale environments might require some adaption and thus the scalability of ADEPTS is considered as low. Besides IE-IRS and ADEPTS, TVA also uses an attack graph. However, the graph generation of the TVA approach is quadratic in the number of hosts involved and can be reduced to $\mathcal{O}(n)$ by grouping hosts into protection domains. Therefore the scalability of TVA is considered as low. CS-IRS and REASSESS are able to handle different sizes of network topologies and thus the scalability capability is considered as high.

Adaptability: During the selection of a response, IE-IRS is capable to take into account a response history. However, the computation time to select an appropriate response increases. Further, IE-IRS only considers negative side effects of a response. Therefore the aptitude to automatically learn from previous responses and thus suit different use-cased is considered as low. In contrast to IE-IRS, ADEPTS takes into account previous effectiveness of the response, but only considers positive effects as basis of the response selection process. The response effectiveness is a value indicating how successful the response was in mitigating the attack. ADEPTS does not consider negative side effects of the chosen response. Therefore the aptitude to automatically learn from previous responses and thus suit different use-cased is considered as medium. The response selection model CS-IRS introduces the response success rate. This response success rate provides the aptitude to learn from previous responses. Therefore the adaptability capability of CS-IRS is considered as high. TVA takes into account several external sources (NIST NVD, CVE) to built a multi-step attack graph for the system environment. As these external sources also store the reported vulnerabilities, TVA contains both, a history and the latest reported ones. By its nature, TVA does not consider negative or positive side effects of a response to a given attack. Therefore the aptitude to automatically learn from previous responses and thus suit different use-cased is considered as medium. Similar to CS-IRS, REASSESS stores a response success rate that represents positive effects of a response. REASSESS also provides a response history stored within a database and thus is much more efficient than IE-IRS. The aptitude to automatically learn from previous responses and thus suit different use-cased is considered as high.

System independency: The IE-IRS response model requires information about the network in terms of a list of services with their IP address and port. In addition the users or customers have to be defined along with the costs related to service degradation or termination. These definitions can be done for any network. Thus, the system independency of IE-IRS is considered as high. The attack graph of ADEPTS can be computed for other environments as well. However, the attack graph of ADEPTS focuses on distributed web-based environments. Therefore an implicit assumption is, that the structure of the intrusion graph is fairly stable. The authors in study [16] reported, deploying ADEPTS in a different setting might require significant modifications. Therefore the system independency of ADEPTS is considers as medium. CS-IRS relies on measures based on expert judgment and the system security policy. The expert judgment and the system security policy could be defined in any network. Similar to CS-IRS, TVA takes into account network scans and vulnerability information of different external sources. These external sources provide system independent vulnerability information. Therefore the system independency of CS-IRS and TVA is considered as high. REASSESS relies on information about the services and the disruptive potential of a response. This information can be collected in any environment. Therefore the system independency of REASSESS is considered as high.

Calculation efficiency: In principle, the IE-IRS response selection model provides a fast calculation methodology to identify an appropriate response, but over time the calculation methodology has to analyze the length of the response history and the number of alternative response actions. A huge amount of alternatives in a long history of response actions lead to an explosion of the number of sequences that IE-IRS has to evaluate. The authors of IE-IRS stated that even though it is possible to optimize the algorithms used within IE-IRS, the number of possible responses increases exponentially with the length of the history of response actions. The computation time for the best response strategy of IE-IRS with 35 resources and a dependency graph of depth of up to 8 computes 34 seconds using an Pentium III with CPU of 550 MHz and 512 MB of RAM. Therefore the calculation efficiency of IE-IRS is considered as low. ADEPTS uses a directed acyclic graph (DAG) to create an attack graph. Even so the attack graph of ADEPTS has to be updated on a regular basis, algorithms for constructing them in linear time

are known. Therefore the calculation efficiency of ADEPTS is considered as high. Similar to IE-IRS, the CS-IRS response selection model provides a fast calculation methodology to identify an appropriate response. The computation time for the best response strategy of CS-IRS in inline mode with $10\,000$ system resources, 100 intrusion responses available in the system and 100 suspected intrusion computes 0.015 seconds using an Intel Core 2 Duo CPU U7700 with CPU of 1.33 GHz and 1 GB of RAM. Significantly less computation time is required with $1\,000$ system resources, 100 intrusion responses available in the system and 100 suspected intrusion. With these settings, CS-IRS computes less than 5μs. Therefore the calculation efficiency of CS-IRS is considered as high. The graph generation of the TVA approach is quadratic in the number of hosts involved and can be reduced to $\mathcal{O}(n)$ by grouping hosts into protection domains. However, a graph for a network containing 100 subnets and $20\,000$ hosts computes 14 minutes using a quad-core Intel Xeon CPU at 1.86 GHz, with 4 GB RAM. Taking into account, that this approach is intended to be used for preventive measures the computation time is less important. Therefore TVA results in a medium scalability capability. As REASSESS is based on CS-IRS, the calculation methodology is similar to CS-IRS. In contrast to CS-IRS, REASSESS was evaluated in conjunction with an IDS in promiscuous mode. Further REASSESS analyzes the impact of no reactive response, positive and negative effect of a response. Therefore the calculation methodology requires more steps to be completed in comparison with CS-IRS. The computation time for the best response strategy of REASSESS in promiscuous mode with 125 alerts computes 12 seconds using a CPU of 2.5 GHz and 1 GB of RAM. Therefore the calculation efficiency of REASSESS is considered as medium.

Usability: The IE-IRS response selection model requires a dependency tree with all available network services offered by hosts, system users, network topology and firewall rules. The importance of each network services has to be defined though a user or a customer that uses this service. As the judgment of users regarding the importance of a network service depends on subjective opinion, the importance ranking results in a diversity of interpretations. Therefore the usability of IE-IRS is considered as medium. ADEPTS uses a semi automated way to re-generate the attack graph. As IE-IRS, ADEPTS requires a vulnerability and a system service description that are created manually. Besides the vulnerability and the system service description, adjustments are required to tailor the attack graph to the system environment. Therefore the usability of ADEPTS is considered as medium. Similar to IE-IRS, CS-IRS the response and intrusion costs, and the system policy are modeled as static values. Further these values are based on expert judgment. Besides the a high amount of manual input, the response and intrusion costs often depend on SLAs and thus should not be modeled as static values. Therefore the usability of CS-IRS is considered as medium. TVA provides a graphical user interface (GUI) to analyze the attack graph, but the TVA attack graphs can be much too complex for easy understanding even in a small setting of 20 hosts in four subnets. To overcome this shortcoming, the GUI of TVA supports a graph navigation with high-level overviews and detail drilldowns. Due to the complexity of the attack graphs of TVA and the fact that they are not easy to understand without the GUI the usability of TVA is considered medium. Similar to CS-IRS, REASSESS requires information about the services, SLA penalty fees, and how disruptive a response is considered in the system environment. In contrast to CS-IRS, all this information can be defined without expert judgment. In REASSESS the penalty costs are modeled in consideration of the SLA violation costs. The remaining information are either known or can be obtained by using network management tools such as SNMP. Consequently, the usability of REASSESS is considered as high.

Security mechanisms: None of the response selection models provide any mechanisms to secure the sensitive incident handling information. Moreover, none of the response selection models prevent unauthorized access to the mitigation and response information. The response selection models that take into account external sources do not ensure that the external sources is a trustworthy origin. Therefore the security mechanisms of all response selection models is considered as low.

Evaluation methodology: IE-IRS was evaluated using 13 different response actions that consists of up to ten firewall rule changes, user accounts and process status modifications. The authors of IE-IRS conclude that selecting different response actions can be done quickly, but this is only due to the fact that only crucial network services are modeled and thus the number of network services is limited. Further, the evaluation relies on optimized data structures. In addition, the authors of the IE-IRS basically measured the performance of their proposed evaluation method instead of the calculation efficiency of IE-IRS. Therefore the evaluation methodology of IE-IRS is considered as poor. In order to evaluate ADEPTS, a metric called survivability is introduced. The value of this metric relies on two dependencies. First, the set of high level system transactions that can be achieved. Second, the set of high level system goals which are not violated in the case of an intrusion. A high level transaction depends on successfully operating services and their interactions between them. Preventing information leakage, for instance is referred as a high level system goal, thus, assuring high level system goals imply preventing certain intrusion goals from being successful. For testing, real attack scenarios were injected into the system. The survivability of the system was compared to no response mechanism, static responses only, and their system. Static responses are applied with a mapping of attacks patterns to pre-defined responses. The authors of ADEPTS have shown that ADEPTS was able to outperform the static response. The introduction of the survivability metric hinders the comparison of results and thus the evaluation methodology

TABLE IV
EVALUATION SUMMARY OF THE RESPONSE SELECTION MODELS.

Criterion	IE-IRS	ADEPTS	CS-IRS	TVA	REASSESS
Automatic deployment	-	+	+	-	+
Scalability	-	-	+	-	+
Adaptability	-	0	+	0	+
System independency	+	0	+	+	+
Calculation efficiency	-	+	+	0	0
Usability	0	0	0	0	+
Security mechanisms	-	-	-	-	-
Evaluation	-	0	-	-	+
Sum	-5	0	3	-3	5

Legend: high (+), medium (0) and low (−)

is considered as medium. CS-IRS was evaluate using the 1998 DARPA/Lincoln Lab offline evaluation data, in particular week 5 for days Monday, Thursday, and Friday, and week 6 on Thursday. The tcpdump data was replayed and Snort, an open source signature-based IDS, is used to generate alerts. In consideration of the outdated evaluation data and the unidirectional generated traffic that was created to analyze and evaluate IDSs, CS-IRS results in a poor evaluation methodology. The TVA publication only provides data about the runtime for the graph generation, but no further insights into its evaluation. Therefore the evaluation methodology of TVA is considered as poor. REASSESS was evaluated using a testbed and three use-case scenarios. The testbed of REASSESS was installed on three virtual machines interconnected by two virtual networks. The reaction system maintains two network interfaces (internal and external), enabling it to intercept ongoing data flows. The test data was replayed bidirectional and the effects of an response have been analyzed. Therefore the evaluation methodology of REASSESS is considered as good.

C. Evaluation summary

In this section, we provide an aggregated overview of the key evaluation results. We have summarized the information presented in Section VI in Table IV.

Although detecting network-based attacks and its patterns generated considerable recent research interest [31]–[33], research in automated attack reaction has not yet been studied as in depth [15]. However, methods of calculating the impact of an attack [34] or the costs related to a response [35] have been proposed by the scientific community.

We found that the comparison of these approaches is difficult because some approaches are only applicable for specific system environments [16]. Another circumstance that affirm this problem is that each of the presented approaches is using different metrics, e.g the response model IE-IRS proposed by [18] only considers negative side effects of a response deployment, while the response model ADEPTS [26] considers the positive effects. Moreover, the majority of reviewed studies include insufficient insights in order to reproduce their approaches and results. In addition, there is no public available source code of these systems.

While each of the approaches employed reasonable metrics that aid in selecting an appropriate response, the evaluation methodology of all response selection models except REASSESS lacks in an evaluation coverage of the entire process of intrusion handling.

This isolated view is also shown in the evaluation of each response model. The majority of the tests are designed to compare the respective proposed approach to simple security measures. Other tests only measure the time which is needed to select a response. Performance measurements are reasonable since the response has to be applied in a small time window, otherwise IRSs are ineffective. However, without validating that the applied response measures are successful, we put this test criterion into question. Despite using simple response measures such as blocking IP addresses, the CS-IRS and IE-IRS studies do not provide any information on their testing to ensure if the selected response has its desired affect.

Another notable fact is that the response selection models are developed to increase the security. However, there are no security mechanisms explained in any of the publications.

While all of these approaches can be used in different environments, ADEPTS, IE-IRS and TVA require building intricate dependency graphs which hinders the scalability of these approaches. Further, the graphs of ADEPTS and TVA require vulnerability information and other data reducing the usability of the approaches.

TVA is a preventive approach and does not deploy any response measure. Neither IE-IRS, which only evaluates the response based on the negative side effects to other services and users. Both, TVA and IE-IRS do not automatically deploy a response. The residual response models are able to automatically deploy a response to a given attack. However, only ADEPTS and REASSESS describe how the feedback on the success of a deployed response is implemented.

VII. Conclusion and Future Work

This work presents REASSESS, a reaction strategy model based on effectiveness assessment as a solution against the growing threat of network-based attacks. We enumerate the requirements of intrusion response systems based on a common scenario and give an overview of existing response selection models. The published response selection models and REASSESS are evaluated in regard to eight evaluation criteria. Based on the findings of this evaluation, we show that REASSESS fulfills the requirements presented in Section III.

Through the implementation of REASSESS, we show that it is practical applicable. Taking into account the modular structure of the system, future work can target on individual aspects of the implementation. To the best of our knowledge, REASSESS is the first response selection model which considers the whole process of incident handling. Further, we demonstrate that REASSESS is aligned to the NIST incident life cycle.

While published response models employed reasonable metrics that aid in finding an appropriate responses, they have shortcomings in their evaluation. In addition, several other challenges in automated response selection have been identified. The first challenge is to determine the set of eligible responses for a system. To the best of our knowledge, there is no general method to solve this problem. Information security standards can solely be used as a starting point for establishing an incident response process, because they are universal in scope. Second, defining the impact of an attack on a given system is another challenging task of automated response selection. Proposed solutions by the scientific community rely on building intricate dependency graphs, including vulnerability interdependencies, intrusion paths, or service dependencies to express the impact of an attack. In REASSESS, we make use of the priority specified within an alert to estimate the impact on the system.

A further key question is how to identify the impact of a response measure. The approach IE-IRS of [18] relies on a simple service dependency graph, but does not describe practical examples. In the framework CS-IRS of [24], responses are ranked based on historical data and expert judgment. To the best of our knowledge, there is no other possibility to overcome this problem. In REASSESS, we consider the impact of a response as negative effects on legitimate requests to a service with relation to the service's importance. Thus, the value can be intuitively estimated with the importance of a service linked to SLA penalty fees.

The last challenge is to evaluate the success of a response. The reviewed studies tend to leave this verification out of scope. Rather, theoretical models to calculate the effectiveness of response measures are applied. Besides, we claim that there is a scarcity of meaningful IRSs tests. The majority is designed to compare the respective presented approach to simple security measures. Other tests solely measure the time which is needed to select a response. We have to stress that replaying recorded network data can not be used to evaluate the response deployment and mitigation capabilities of an IRS.

We believe it to be promising a approach to use penalty costs defined in SLA to derive the importance of a system component. With the developed formulas, it is also possible to estimate the impact of an attack based on the priority of a given alert. Since the response success rate (rsr) can be stored in a database and updated with every response deployed, the system is able to adapt.

However, several challenges are worth future research. In particular, we intend to integrate common standards for exchanging incident related information. To make the model more realistic, future work should elaborate a confidence metric. In addition, developing an alert correlation mechanism would aid in handling large amounts of alerts raised by the detection engine and would also allow to assess the success of applied responses in more advanced attack scenarios. The development of a secured communication channel between the IDS and the reaction system would enable the usage of multiple IDS nodes. Finally, methods of parallel programming can be applied to speed up the calculation efficiency of REASSESS.

As mentioned in Section I this paper aims to overcome closed source and system dependency of this research domain. Thus, we share the test scenario scripts used by REASSESS as well as the source code of REASSESS itself as public available data which can be downloaded from https://www.dasec.h-da.de/staff/jessica-steinberger/.

Acknowledgements

This work was partly supported by the German Federal Ministry of Education and Research (BMBF) under grant number 03FH005PB2 (INSAIN) and CASED.

References

[1] "Die Lage der IT Sicherheit in Deutschland 2011," Online, Bundesamt für Sicherheit in der Informationstechnik, May 2011.

[2] Symantec. (2013, April) Internet Security Threat Report 2013. Online. Symantec Corporation. http://www.symantec.com/content/en/us/enterprise/other_resources/b-istr_main_report_v18_2012_21291018.en-us.pdf Last accessed on October 15th, 2013.

[3] P. Hunter, "Distributed Denial of Service (DDOS) Mitigation Tools," *Network Security*, no. 5, pp. pages 12 – 14, May 2003. [Online]. Available: http://dx.doi.org/10.1016/S1353-4858(03)00510-5

[4] T. Peng *et al.*, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *ACM Computing Surveys*, vol. 39, no. 1, pp. pages 1 – 42, April 2007. [Online]. Available: http://dx.doi.org/10.1145/1216370.1216373

[5] A. Bremler-Barr and H. Levy, "Spoofing prevention method," *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, 2005. [Online]. Available: http://dx.doi.org/10.1109/INFCOM.2005.1497921

[6] "Akamai's State of The Internet: Q1 2014 Report," Online, Akamai. [Online]. Available: http://www.akamai.com/dl/akamai/akamai-soti-q114.pdf

[7] "Akamai's State of the Internet: Q3 2014 Report," Online, Akamai Technologies, Inc., 2014. [Online]. Available: http://www.stateoftheinternet.com/downloads/pdfs/2014-internet-security-report-q3.pdf

[8] N. Anuar *et al.*, *An investigation and survey of response options for Intrusion Response Systems (IRSs)*. Institute of Electrical and Electronics Engineers, August 2010, pp. 1 – 8. [Online]. Available: http://dx.doi.org/10.1109/ISSA.2010.5588654

[9] B. Endicott-Popovsky and D. Frincke, "Adding the Fourth "R": A Systems Approach to Solving the Hacker's Arms Race," in *Proceedings of the 2006 Symposium 39th Hawaii International Conference on System Sciences*, January 2006.

[10] M. Dadomo. (2014, February) Gute Perspektiven für Standort Deutschland durch Industrie 4.0. Online. Verein Deutscher Ingenieure. Last accessed on February 7th, 2014. [Online]. Available: http://www.vdi.de/artikel/gute-perspektiven-fuer-standort-deutschland-durch-industrie-40/

[11] "Die Lage der IT Sicherheit in Deutschland 2014," Online, Bundesamt für Sicherheit in der Informationstechnik, Jan 2015. [Online]. Available: https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/Lageberichte/Lagebericht2014.pdf?__blob=publicationFile

[12] "DDoS Survival Handbook," Online, Radware, May 2013. [Online]. Available: http://security.radware.com/uploadedFiles/Resources_and_Content/DDoS_Handbook/DDoS_Handbook.pdf

[13] "The Risk vs. Cost of Enterprise DDoS Protection - How to Calculate the ROI from a DDoS Defense Solution," Online, Arbor Networks, November 2012. [Online]. Available: http://pages.arbornetworks.com/rs/arbor/images/Whitepaper_Risk_vs_Cost_Enterprise_DDOS_Protection.pdf

[14] J. Kindervag *et al.* (2013, May) Develop A Two-Phased DDoS Mitigation Strategy. Online. Forrester Research, Inc. Last accessed on October 23rd, 2013. [Online]. Available: http://www.forrester.com/pimages/rws/reprints/document/86101/oid/1-KWRIUQ

[15] N. Stakhanova *et al.*, "A taxonomy of intrusion response systems," *International Journal of Information and Computer Security*, vol. 1, no. 1/2, pp. pages 169 – 184, February 2007. [Online]. Available: http://dx.doi.org/10.1504/IJICS.2007.012248

[16] N. Stakhanova *et al.*, "Towards cost-sensitive assessment of intrusion response selection," *Journal of Computer Security*, vol. 20, pp. pages 169 – 198, June 2012. [Online]. Available: http://dx.doi.org/10.3233/JCS-2011-0436

[17] C. Strasburg, "A framework for cost-sensitive automated selection of intrusion response," Master's thesis, Iowa State University, 2009.

[18] T. Toth and C. Kruegel, "Evaluating the impact of automated intrusion response mechanisms," *18th Annual Computer Security Applications Conference, 2002. Proceedings.*, pp. 301 – 310, 2002. [Online]. Available: http://dx.doi.org/10.1109/CSAC.2002.1176302

[19] C. Sanders and J. Smith, *Applied Network Security Monitoring - Collection, Detection, and Analysis*, 1st ed. Syngress: Elsevier, December 2013.

[20] M. A. Faysel and S. S. Haque, "Towards cyber defense: research in intrusion detection and intrusion prevention systems," *IJCSNS International Journal of Computer Science and Network Security*, vol. 10, no. 7, pp. 316–325, 2010.

[21] J. François, S. Wang, R. State, and T. Engel, "BotTrack: Tracking Botnets Using NetFlow and PageRank," in *NETWORKING 2011*, ser. Lecture Notes in Computer Science, J. Domingo-Pascual, P. Manzoni, S. Palazzo, A. Pont, and C. Scoglio, Eds. Springer Berlin Heidelberg, 2011, vol. 6640, pp. 1–14. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20757-0_1

[22] F. Tegeler, X. Fu, G. Vigna, and C. Kruegel, "BotFinder: Finding Bots in Network Traffic Without Deep Packet Inspection," in *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies*, ser. CoNEXT '12. New York, NY, USA: ACM, 2012, pp. 349–360. [Online]. Available: http://doi.acm.org/10.1145/2413176.2413217

[23] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, and C. Kruegel, "Disclosure: Detecting Botnet Command and Control Servers Through Large-scale NetFlow Analysis," in *Proceedings of the 28th Annual Computer Security Applications Conference*, ser. ACSAC '12. New York, NY, USA: ACM, 2012, pp. 129–138. [Online]. Available: http://doi.acm.org/10.1145/2420950.2420969

[24] C. Strasburg *et al.*, *A Framework for Cost Sensitive Assessment of Intrusion Response Selection*. Institute of Electrical and Electronics Engineers, July 2009, pp. 355 – 360. [Online]. Available: http://dx.doi.org/10.1109/COMPSAC.2009.54

[25] F. P. Stanley, "Intursion detection and respone for system and network attacks," Master's thesis, Iowa State University, 2009. [Online]. Available: http://lib.dr.iastate.edu/cgi/viewcontent.cgi?article=1730&context=etd

[26] B. Foo *et al.*, *ADEPTS: Adaptive Intrusion Response Using Attack Graphs in an E-Commerce Environment*. Institute of Electrical and Electronics Engineers, June 2005, pp. 508 – 517. [Online]. Available: http://dx.doi.org/10.1109/DSN.2005.17

[27] Y. Wu *et al.* (2004) ADEPTS: Adaptive Intrusion Containment and Response using Attack Graphs in an E-Commerce Environment. [Online]. Available: https://engineering.purdue.edu/~sbagchi/Research/Papers/adepts_dsn04_submit.pdf

[28] C. Strasburg, S. Basu, N. Stakhanova, and J. Wong, "The Methodology for Evaluating Response Cost for Intrusion Response Systems," Online, Iowa State University, 2008. [Online]. Available: http://www.cs.unb.ca/~natalia/TR08-12.pdf

[29] S. Jajodia and S. Noel, *Topological Vulnerability Analysis*. Springer-Verlag, September 2010. [Online]. Available: http://dx.doi.org/10.1007/978-1-4419-0140-8_7

[30] Z. Chen *et al.*, "A Pragmatic Methodology for Testing Intrusion Prevention Systems," *The Computer Journal*, vol. 52, no. 4, pp. pages 429 – 460, June 2009. [Online]. Available: http://dx.doi.org/10.1093/comjnl/bxn043

[31] J. Francois *et al.*, *BotTrack: Tracking Botnets Using NetFlow and PageRank*. Springer, May 2011, pp. 1 – 14. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-20757-0_1

[32] S. Silva *et al.*, "Botnets: A survey," *Computer Networks*, vol. 57, no. 2, pp. pages 378 – 403, February 2013. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2012.07.021

[33] A. Sperotto *et al.*, "An Overview of IP Flow-Based Intrusion Detection," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 3, pp. pages 343 – 356, July 2010. [Online]. Available: http://dx.doi.org/10.1109/SURV.2010.032210.00054

[34] R. Vasudevan *et al.*, "MIDAS: An Impact Scale for DDoS attacks," in *Local & Metropolitan Area Networks, 2007. LANMAN 2007. 15th IEEE Workshop on*, June 2007, pp. 200 – 205. [Online]. Available: http://dx.doi.org/10.1109/LANMAN.2007.4295999

[35] W. Lee *et al.*, "Toward Cost-sensitive Modeling for Intrusion Detection and Response," *Journal of Computer Security*, vol. 10, no. 1-2, pp. pages 5 – 22, Jul. 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=597917.597919