Improving Attack Mitigation with a Cost-sensitive and Adaptive Intrusion Response System

Rodion Iafarov, Ruediger Gad, Martin Kappes
Frankfurt University of Applied Sciences
Frankfurt am Main, Germany
email: yafarovrs@gmail.com, {rgad, kappes}@fb2.fra-uas.de

***Abstract*—Because of the rise of the number of attacks in computer networks, mitigation measures have to be applied in an efficient manner. The time frame for attack mitigation is shortened what makes using classical manual intervention approaches less efficient. Even though the idea of Intrusion Response Systems (IRS) is not new, IRS are still not widely used. Potential users are typically afraid of inadequate reactions, which could worsen the situation or could even be used as a part of attacks. In this paper, we present a cost-sensitive, retroactive, adaptive, and preemptive IRS that is intended to support network administrators in the attack mitigation and decision making processes. Our approach aims on balancing the costs of responses and attacks, adapts to changing situations, and optimizes the selection of responses and response deployment locations. Experimental results obtained with an evaluation prototype show that our approach works and is feasible from a performance perspective.**

Keywords–Intrusion Response System; Risk Assessment; Impact Cost Assessment; Dynamic; Adaptive.

I. INTRODUCTION

The amount of attacks on Information and Communications Technology (ICT) increases, e.g., in 2013 an increase in the number of web-based targeted attacks of 25% and a 91% increase in targeted attacks campaigns could be observed [1]. Successful ongoing attacks may lead to severe consequences like significant monetary losses or may even endanger human health.

In order to avoid such consequences, it is paramount to mitigate attacks quickly and efficiently. Due to the increase of complexity and pace of attacks and intrusions, however, classical manual intervention is often not sufficient anymore. Weaknesses of classical manual intervention are the lack of speed, the requirement of expert knowledge, and the increasingly complicated response selection process.

Consequently, the necessity for more automated solutions has become obvious [2]. Intrusion Response Systems (IRS), which can be seen as an extension to Intrusion Detection Systems (IDS) and Intrusion Prevention Systems (IPS) [3], have gotten more attention in recent years, particularly the combination of IRS with other approaches like decision-making [4][5].

The aim of automatic response approaches is to deal with attacks faster and more efficiently [6]. Fully automated systems, however, may trigger erroneous actions that may worsen the situation. As one consequence, system administrators usually perceive fully automated systems sceptical and are unwilling to hand over control to fully automated systems. Semi-automated approaches aim on solving this issue by allowing manual control while still accelerating the mitigation.

In this paper, we present an approach which improves the response selection process by supplying the user with pre-selected optimized response suggestions. Our solution takes advantages of existing methods and combines them for efficiently mitigating attacks. Furthermore, our approach allows additional human interaction and intervention, e. g., rolling back applied countermeasures or applying alternative actions, which may better fit in case the situation changes. With these mechanisms, we believe that the classical manual process can be significantly improved with respect to quality, speed, and deployment of reactions. While our system belongs to the class of manual response systems it can be extended to operate fully automated.

We assume that information about attacks is readily provided, e. g., by an IDS. The detection of attacks is beyond the scope of this paper.

In the following, we first present related work. In Section III, we introduce important requirements. Afterwards in Section IV we present our response selection approach. In Section V, we perform an assessment of our approach with a prototype. Finally in Section VI, we provide a conclusion of our findings and present an outlook on future work.

II. RELATED WORK

One line of research in the field of automatic and semi-automatic IRS deals with the development and application of cost models [7]–[9]. The objective of cost-sensitive approaches is to define a consistent metric, e. g, to balance the costs of attacks and responses or for decision-making. The classification proposed by Shameli-Sendi, Ezzati-jivan, Jabbarifar, and Dagenais [10] shows that recent researches pay more and more attention to the risk assessment mechanisms, adjustment and prediction abilities.

The approach to intrusion response proposed by Stakhanova, Strasburg, Basu, and Wong [6] introduces 4 types of costs: intrusion/response impacts on the system and intrusion/response operational costs. The response impact is evaluated based on the defined system goals and their importance, and the intrusion impact is evaluated with respect to the response ability to counter this damage. The response operational cost includes the costs for the setup of responses, the costs for the deployment of responses, and the costs of the data processing overhead needed to analyze the results of responses. The intrusion operational cost includes the baseline cost present for an attack and the actual damage that can be potentially caused by a successful attack. The main disadvantage of the model is absence of probabilistic analysis. Additionally, this model does not take into account combinations of responses that mitigate

 ISBN: 978-1-61208-398-8

attacks partially and sets of existing attacks and already applied reactions.

The cost model proposed by Lee, Miller, Stolfo, Fan, and Zadok [11] considers the potential harm of an attack and attack operational costs for monitoring and detection. In [11], Lee et al. do not split the response cost into categories but calculate an overall cost of acting against the attack. The main disadvantage of this model is uncertainty in the cost analysis due to incomplete or imprecise estimation and limitations related to the reconstruction of the model in case of metric changes.

The approach proposed in [9], defines the cost of damage caused by intrusion as the sum of intrusion impact on the system and cost of daily maintenance of various aspects of the detection system. The response cost is calculated cognate as the sum of impact on system and operation cost value.

In [7], Yaorui Wu and Shufen Liu use a cost model, which depends on probabilities referring to detection methods. Probabilistic techniques help to address risks of inadequate response deployment in case of detection errors.

Forecasting techniques can be applied to predict the possible development of attacks. Preemptive IRS, like the one presented in [6], use forecasting techniques to increase the accuracy of the countermeasure selection.

In [12], two types of response executions were defined: a burst model, which has no risk assessment mechanisms once the response has been applied and a retroactive model, which includes feedback mechanism that assesses the response effect based on the result of the applied response. In [13], an adaptive IRS was proposed that additionally introduces a response effectiveness index, which is used as quality indicator for responses.

Obviously, IRS should behave differently for different kinds of attacks. Therefore, attack classifications are required to allow an adequate response selection. The classification presented in [11], divides attacks into four main categories: probe, denial of service (DoS), remote to local (R2L), and user to root (U2R). In [14], Wu, Xiao, Xu, Peng, and Zhuang introduce another categorization for attacks similar to the one presented in [11]. In addition to the attack type, the categorization by Wu et al. also takes a location property into account, e. g., privilege escalating can be local or remote, resource depletion can be applied to host and/or network.

Risk and cost assessment are based on resource dependencies. The confidentiality, integrity, and availability (CIA) triad [15], e. g., can be used to define the importance of particular system resource security properties. Dependencies can be defined using the idea of a resource type hierarchy, as introduced in [16]. An additional graph-based data structure, the "system map", can be used to carry information about specific instances of the system. Within the system map, system resources are represented as vertices and dependencies between resources are defined as edges.

Existing models lack consistency and do not take advantages of each other. With our approach, we combine the aforementioned approaches in order to create a cost-sensitive, retroactive, adaptive, and proactive IRS with risk assessment based on resource dependencies, a dynamically evaluated cost model, and sustainable countermeasures according to the classification proposed in [10]. To the best of our knowledge, no such combination was presented before. Additionally, we consider the applicability of our solution in real networks in order to make a step towards using such solutions in real scenarios. Thus, we propose a semi-automated solution instead of a fully automated one, as it can be considered as more reliable. Our proposed solution is flexibly such that it can be adapted for being applied in varying environments and with varying sets of responses. Our model also aims on removing the lack of consistency with existing solutions by combining multiple different approaches.

III. System Resources Categorization

In our proposed approach, we use resources dependencies as a risk assessment criterion. As the first step of categorization, we assign importance values to the system resource security properties. As in [6], the importance of a particular system instance with respect to the CIA principles is defined by float values in the range $[0, 1]$, where 0 denotes minimum and 1 denotes maximum importance.

Additionally, as described in [17], the attack impact is split into three categories: none, partial, complete. The importance of the security properties defines how critical the loss of a certain attribute is. However, it is important to distinguish complete and partial affection to avoid unnecessary risk elevation, which may lead to inadequate response deployment. For each security property, we define two values, which correspond to partial and complete loss.

Dependencies between system resources are used for the risk assessment and impact cost calculation. During response selection, we have to take possible impacts on dependent system resources into account. Dependencies between resources can be declared as a directed graph.

To deal with cycles, the following procedure is used: At first we use a depth-first search and mark the states of the observed resources. When we observe a system resource for the second time, we check if there is an additional impact on the security properties in the new state. If yes, we assess the additional impact and go through the dependent nodes according to the new impact. Otherwise, we do not process dependent nodes as the previous assessment remains valid. This algorithm ends because maximum possible impact is defined, when all security properties are affected, and we always accumulate impact. When the maximum impact is reached for a resource node, the algorithm stops observing this node.

For defining dependencies, causal links are used. Conditions specify which security properties have to be affected to create a defined impact on the dependent system resource. For each dependency, the probability of the event is specified in order to define how probable the occurrence of the impact is. The defined probability is used to perform forensic analysis and to predict the possible attack development.

The dependency structure is also used to optimize the deployment location, which aims on minimizing risks and impact cost, as the cost for a response depends on the location where it is deployed or implemented. E. g., isolating an entire network affects all instances in the network whereas isolating a host only affects the host and the services on the host. Thus, in addition to finding a response with adequate costs, the deployment location can also be optimized.

The type attribute was added, because responses can have different impacts depending on the location of the resource they are applied on. The parameter corresponds to the affected instances type parameter of the response and is used to improve the accuracy of the response impact determination.

 ISBN: 978-1-61208-398-8

Additionally, the system resource physical location attribute has to be configured for the deployment procedure. Not all properties are involved in the response selection, as they were considered as less important due to the less impact on the process. Nevertheless, our proposed approach can be extended to consider additional attributes.

IV. Response Selection

The response selection procedure is performed as follows: At first the system resources and the impact of attacks are assessed. Then, a set of possible responses and the corresponding locations are determined. Afterwards, the impact of the selected responses is assessed. Finally, responses and locations are optimized.

The attack impact is defined by the target(s) and security properties, which it can affect. Based on the description of the environment, we can assess the possible impact on the target(s) and dependent system resources. For each response, besides impact, we define which security properties it can protect. Using this information we form subset of possible responses, which can mitigate an attack.

In difference to [2], our proposed algorithm also considers responses that do not mitigate attacks completely, but mitigate the impact on security properties, which are crucial for attacked system resource. The possibility to combine multiple responses for the attack mitigation is also taken into account. A response is added to the subset of possible responses if: the response completely mitigates an attack; or the response protects properties that are relevant for the system resource; or the response mitigates an attack partially while other responses exists that can protect the remaining relevant security properties.

Our proposed method forms the subset of the possible responses to select an optimized countermeasure aiming on minimizing overall risks and costs. A trade-off between attack and response impacts is performed and it is avoided to worsen the situation by wrong response deployment.

The location where a response is applied has to be determined as well. Response costs differ depending on the location and the costs can be minimized by optimizing the deployment location. The environment description is used to find all possible locations for the deployment. Then, the costs are assessed in order to determine the one, which provides minimal cost.

In order to optimize the response selection when multiple attacks are present in the system, our proposed solution takes new attacks, the set of current attacks, and the set of already applied responses into account for the calculations. With this approach it can be, e. g., identified if new attacks can be mitigated by already applied response or if it is possible to reduce costs by replacing previously deployed countermeasures Furthermore, if an attack was stopped, it is required to reconsider the deployed reactions and possibly apply a new set of responses, or cancel responses, which mitigated stopped attack.

A. Attack Cost

The attack cost is based on the impact on system resources and operational costs as defined in [6]. As we mentioned, it is assumed that the required information about an attack, including attack target, is provided by the detection mechanisms. We evaluate the probable attack development and consider system resources dependencies during cost assessment. The cost of an attack, denoted by a, is assessed by the function $atCost(a)$ and is calculated in accordance to 1.

$$atCost(a) = p_{det}(a)(\sum_{s \in S} p_{imp}(a)\omega(s) + opCost(a)), \quad (1)$$

where S is a set of security properties affected by the attack and $s \in S$ denotes a security property of a system resource. $\omega(s)$ is a function that computes the importance value of the affected security property s. $p_{det}(a)$ is a function that calculates the probability of the correct detection of an attack. $p_{imp}(a)$ is a function that computes the probability that an attack a will actually impact the system. Additional weights can be added as extension to the provided solution. The operational cost of attack a, denoted by function $opCost(a)$, is assigned by value in the range $[0, 1]$, as proposed in [6].

The probability of an attack impact is one of the required parameters. It allows to evaluate possible attack development. Additionally, probabilities of the impact on the attacked system resource dependencies are calculated to evaluate possibility of the impact and assess attack effect cost. The probability of the affection creates a non-increasing sequence, as for each next step, previous steps have to be successfully performed. The probability of the next step is calculated as multiplication of the probabilities of all required previous steps. This approach minimizes the risk of overestimating an attack and it improves the adequacy of the reaction. Additionally, we decrease the attack cost according to the detection confidence in order to avoid inadequate reactions in case of a detection error. This approach allows to minimize risks and perform additional investigation before actual deployment.

B. Response Cost

After the subset of possible responses is formed and the possible locations for the deployment were defined, the effects of each response in its possible deployment locations are assessed. The cost of a response, denoted by r, is calculated by the function $respCost(r)$. To avoid negative cost values, the base cost is initially set to the sum of all attacks persisting in the system, including the current attack, as shown in 2.

$$\sum_{a \in A} atCost(a), \quad (2)$$

where A denotes the set of all persisting attacks in the system. We also introduce an efficiency factor, denoted by the function $respEff(r)$. The efficiency property is specified for each response and is changed according to the results of response application. The efficiency is calculated as ratio of the number of successfully mitigated attacks by the response over the number of overall number of attacks we tried to mitigate by the response in accordance to 3.

$$respEff(r) = \frac{\#ofSuccessfullyMitigated}{overall\#ofTriesToMitigate}. \quad (3)$$

The response cost is decreased by the ability to mitigate the current attack and other persisting attacks influenced by the efficiency factor, as shown in 4:

$$respEff(r) \sum_{m \in M_r} atCost(m), \quad (4)$$

where M_r is the set of all attacks in the system which can be mitigated by the response r including the current attack.

 ISBN: 978-1-61208-398-8

Afterwards, we increase the response cost due to negative impact on the system expressed as in 5.

$$\sum_{s \in S_r} \omega(s), \tag{5}$$

where $\omega(s)$ denotes a function that calculates the importance of the security property s, which is affected by the response r and S_r is the set of security properties affected by the response r. The response impact is calculated in the same way as it is done for the attack, whereas the probability component is excluded as, unlike to intrusions, we can precisely define the impact of responses. Finally, we include the operational cost of the response r, denoted by function $opCost(r)$, in the same way as it is done for the attacks. Consequently, the response cost can be calculated as follows:

$$\begin{aligned} respCost(r) = &\sum_{a \in A} atCost(a) + \sum_{s \in S} \omega(s) + opCost(r) \\ &- respEff(r) \sum_{m \in M} atCost(m). \end{aligned} \tag{6}$$

Based on $respCost(r)$, we choose the response with the lowest cost value that is lower than the sum of the costs of existing and current attacks. If the response cost value is higher or equal than the overall cost, it means that the reaction can worsen the situation and additional investigation is required. The system resource state is considered healthy if for every security property of the system resource the following is true: there is no negative response impact and if there is an attack impact, it is mitigated by the deployed response(s).

V. Evaluation

We considered the following parameters as specified in [18] for the evaluation procedure: flexibility, dynamic, efficiency, ease of use, minimization of negative impact.

a) Flexibility: Flexibility of the proposed IRS is achieved by the system resources description method, which models dependencies between system resources as directed graph and is generally applicable for various environments. Response object properties can be changed as well to adapt priorities if it is required. The set of the system resources and response properties is not fixed and can be extended for additional flexibility.

b) Dynamic: Static IRS can be less efficient as they do not adapt to changes in the environment. Our proposed method tracks changes in the environment caused by attacks and responses and adapts to the current state of the system. This is achieved by getting feedback from the system after countermeasure deployment and consideration of already deployed responses.

c) Efficiency: The performance is one of the factors that affects the efficiency. We used the time for computing the results as measure of the performance. As our proposed IRS is intended to be used in small to medium sized enterprises (SMEs), the performance evaluation was performed with a desktop class computer with an Intel(R) Core(TM) i5-3330 CPU with 3 GHz and 8 GB RAM.

In Figures 1, 2 and 3, the average computation time for different numbers of attacks, responses and system resources is shown. During the analysis we varied the number of the analyzed type and fixed the number of the other types to 100. For example, if we perform an experiment by varying the number of system resources, the numbers of the attacks and responses was fixed to 100. The calculated computation time determines how long it took to process all generated attacks.

The computation time depends on the complexity of the system structure. The following system structure was used: a root resource is connected to all other resources, all dependent resources are connected to every node except the root node. Attacks are always performed on root node, so all resources are affected and are considered during computation. For each setup 10 experiments were performed and the average time was computed. For each experiment, the specified number of system resources and responses is created and then attacks are generated concurrently in multiple threads.

In Figure 1, it can be seen that the number of system resources affects computation time more than the size of the set of responses as depicted in Figure 2. Our proposed method aims on SMEs, which limits the number of system resources. So, if 1.6 seconds are required to process 100 attacks in the environment with 2500 system resources, on average it takes only 0.016 seconds to process one attack. We consider this time as acceptable as it significantly reduces the gap between intrusion detection and deployment of the response in comparison to classical manual approaches. Whenever, results for big environments are slower than ones demonstrated by Stakhanova, Strasburg, Basu, and Wong in [6], growth is near to linear.

For the results of the measurements as shown in Figure 3, the number of responses and attacked system resources was fixed to 100. The number of intrusions affects calculation time most of all, as was also concluded in [6]. The response selection procedure takes 47.34 seconds to process 2500 concurrent attacks, on average it takes only 0.019 seconds to process one attack. This result is close to the time required to process one attack in case of 2500 system resources in the environment. The rapid computational time escalation occurs as all attacks persisting in the system are involved in the countermeasure selection, because we also look for the responses which can minimize costs for mitigating not only the impact of the new attack, but also for mitigating all other attacks persisting in the system.

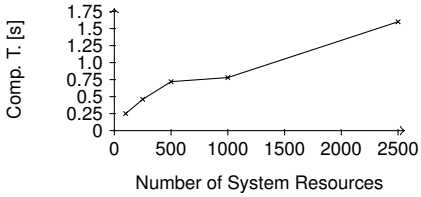

Figure 1. Performance (System Resources)

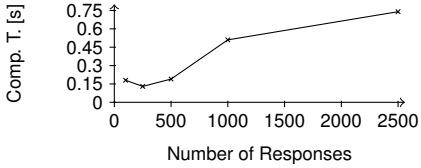

Figure 2. Performance (Responses)

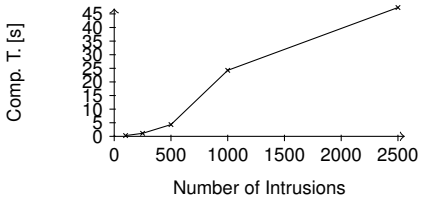

Figure 3. Performance (Intrusions)

d) Ease of use: One of the objectives for the research was to help system administrators of SMEs to deal with intrusions more efficiently. Our proposed system can be used both in automatic and semi-automatic modes. Consequently, system administrator can use an advantage to choose from the list of proposed responses and deployment locations. Additionally, we provide deployment and feedback mechanisms, which help in evaluating results of the deployment.

e) Minimization of negative impact: To illustrate minimization of the negative impact, we use an example assuming a simple environment with 2 hosts in the same sub-network. One of them contains web service for which availability is crucial, the second one contains FTP server for which integrity and confidentiality are important security properties (see Table I). Note that importance differs for the cases of partial and complete loss. Dependencies between resources with required conditions and possible effects are defined in the Table II. The set of the responses, including description of mitigation abilities and impact, is provided in the Table III. For simplification, we assume that the efficiency of the responses equals to 1 and that only one attack persists in the system. Additionally, the impact of a response is always either complete or none. We also assume that the attack impact and detection probabilities are equal to 1.0 and that the attack operational cost is equal to 0.5.

Consider the case, when sub-network instance is under attack and is intrusion entry point. The attack can affect confidentiality and integrity of the instance and the impact is complete. The probability of the impact on dependent resources is calculated as multiplication of the attack probability and probability of the transition to new state according to dependencies (see Table II). In accordance to (1), we compute the attack cost:

$$atCost(a) = 1 * (1 * 0.9 * 0.8 + 1 * 0.7 * 1.0 + 1 * 0.7 * 0.4 + 1 * 0.8 * 0.7 + 0.5) = 2.76.$$

The "isolate sub-network" is one of the possible responses. At first we set the value of the response cost to attack cost value: $respCost(r) = 2.76$. The impact of the response is equal to 2.1, as it affects the availability of the entire network, so we add this value to the cost. Additionally, we add the operational cost value. As a last step, we decrease the response cost due to efficiency against attack. Note that the efficiency coefficient equals to 1.0 for the sake of simplicity. According to (6), we get the response cost:

$$respCost(r) = 2.76 + 1.0 + 0.1 + 1.0 + 0.5 - 1 * 2.76 = 2.6.$$

Thereby, this response is already worth deploying. The "block port" reaction gives slightly better result $respCost(r) = 2.2$ due to lower operational cost. Additionally, we can deploy "isolate host" reaction to both hosts. Cost values will be $respCost(r)_{FTP} = 0.1 + 0.3 = 0.4$ for FTP server and $respCost(r)_{web} = 1.0 + 0.3 = 1.3$ for web server, with overall cost value $respCost(r) = 1.7$, which is better than both previous responses deployed on sub-network instance. Such cases are not considered by existing models. This example shows the importance of the deployment location. Nevertheless, the "block connection" response minimizes overall costs with response cost $respCost(r) = 0.1$.

Now, let us consider the case, when an attack partially affects confidentiality and integrity of the sub-network. The attack cost in this case is calculated as follows:

$$atCost(a) = 1 * (1 * 0.7 * 1.0 + 0.5) = 1.2.$$

The "network isolation" reaction has the following cost:

$$respCost(r) = 1.2 + 1.0 + 0.1 + 1.0 + 0.5 - 1 * 1.2 = 2.6.$$

The determined cost value is higher than the attack cost, thus it is better to keep the attack in the system as the deployment of this response will worsen the situation. The "block connection" response again gives the minimal cost $respCost(r) = 0.1$, while protecting all security properties. Another option is to disable an account. For this response, the cost will be $respCost(r) = 0.2$. In case of close cost values, the response efficiency coefficient, evaluated according to (3), allows making a decision based on the history of the response deployments in order to find the reaction with best chances to mitigate an attack.

TABLE I. SYSTEM RESOURCES

Sys. Res.	Confidentiality (Part./Compl.)	Integrity (Part./Compl.)	Availability (Part./Compl.)
Sub-network	0/0	0/0	0.7/1.0
FTP server	0.8/1.0	0.6/1.0	0/0.1
Web server	0/0.4	0.7/0.85	0.8/1.0

TABLE II. SYSTEM RESOURCES DEPENDENCIES

Sys. Res.	Dep. Sys. Res.	Dep. Cond.	Dep. Eff.	Prob.
Sub-net.	FTP	Conf. (Compl.)	Conf. (Part.)	0.9
		Int. (Part.)	Int. (Compl.)	0.7
		Avail. (Compl.)	Avail. (Compl.)	0.2
	Web	Conf. (Compl.)	Conf. (Compl.)	0.7
		Int. (Compl.)	Int. (Part.)	0.8
		Avail. (Part.)	Avail. (Compl.)	0.6

TABLE III. RESPONSES

Response	Confi. (Miti./Imp.)	Integr. (Miti./Imp.)	Avail. (Miti./Imp.)	Op. Cost
Isolate sub-net.	1/0	1/0	0/1	0.5
Isolate host	1/0	1/0	0/1	0.3
Block connection	1/0	1/0	1/0	0.1
Block port	1/0	1/0	0/1	0.1
Delay connection	0/0	0/0	1/0	0.2
Shutdown host	1/0	0/1	0/1	0.3
Disable account	1/0	1/0	0/0	0.2
Stop service	1/0	0/1	0/1	0.2

 ISBN: 978-1-61208-398-8

f) Limitations: While semi-automatic solutions are one step into the direction they still require human interaction and thus are slower than fully automated systems. Fully automated IRS, however, bring additional risks like erroneous responses or attackers misusing an IRS to trigger inadequate response deployment. Additional research on this topic is required.

The environment description process is a limitation of our proposed approach as it requires expert knowledge, is labor-intensive, and may be expensive. Our approach aims on SMEs, which typically lack resources and may not be able to afford personnel with expert knowledge in the field of information security. Notwithstanding, in case of not rapidly changing environment this limitation is not crucial, as the configuration is done once and reconfiguration is not required until any changes are introduced.

Assumptions according to the required information about the attack also imply limitations for the implementation in real environments. Our approach requires as precise and as detailed information as possible.

VI. Conclusion and Future Work

In this paper, we present an approach for optimizing the selection as well as the location of responses for mitigating attacks in computer networks. We have shown that multiple factors have to be taken into account. Existing solutions typically only consider subsets of factors that affect response selection mechanism in reality.

Our research focused on the practical application. Our proposed solution aims at balancing the costs of attacks versus the costs of countermeasures. We developed a model based on existing solutions that combines expert knowledge used for the resource assessment, increases accuracy of the selected response, and significantly reduces the gap between attack detection and response deployment with automated mechanism. Our approach allows to perform evaluation and adapt costs to the specific environments and security policies.

The concept was evaluated with a prototype implementation. The performance of the prototype showed acceptable results, even though the implementation can still be optimized. A possible optimization, e. g., is the pre-calculation of risk assessment values. Nevertheless, our prototype showed results that are sufficient for being used in real environments.

As future prospects, we plan to take more input parameters into account for the response selection. Additionally, we are going to research adaptable components in order to add flexibility and provide more optimized and efficient reactions to attacks and attack combinations.

Acknowledgment

This work was supported in part by the German Federal Ministry of Education and Research in scope of grant 16BY1201C. Responsible for the content are the authors.

References

[1] Symantec Corporation, “2013 trends,” Internet Security Threat Report 2014, vol. 19, April 2014.

[2] N. Stakhanova, S. Basu, and J. Wong, “A taxonomy of intrusion response systems,” International Journal of Information and Computer Security, vol. 1, no. 1/2, January 2007, pp. 169–184.

[3] N. B. Anuar1, M. Papadaki1, S. Furnell, and N. Clarke, “An investigation and survey of response options for intrusion response systems (irss),” Information Security for South Africa (ISSA), 2010, pp. 1–8.

[4] C. Mu and Y. Li, “An intrusion response decision-making model based on hierarchical task network planning,” Expert Systems with Applications: An International Journal, vol. 37, no. 3, March 2010, pp. 2465–2472.

[5] X. Zan, F. Gao, J. Han, X. Liu, and J. Zhou, “Nair: A novel automated intrusion response system based on decision making approach,” Information and Automation (ICIA), 2010, pp. 543–548.

[6] N. Stakhanova, C. Strasburg, S. Basu, and J. S. Wong, “Towards cost-sensitive assessment of intrusion response selection,” Journal of Computer Security, vol. 20, no. 2-3, 2012, pp. 169–198.

[7] Y. Wu and S. Liu, “A cost-sensitive method for distributed intrusion response,” Computer Supported Cooperative Work in Design (CSCWD), April 2008, pp. 760–764.

[8] N. Stakhanova, S. Basu, and J. Wong, “A cost-sensitive model for preemptive intrusion response systems,” Advanced Information Networking and Applications (AINA), May 2007, pp. 428–435.

[9] A. Ikuomola and A. S. Sodiya, “A credible cost-sensitive model for intrusion response selection.” in CASoN. IEEE, 2012, pp. 222–227.

[10] A. Shameli-Sendi, N. Ezzati-jivan, M. Jabbarifar, and M. Dagenais, “Intrusion response systems: Survey and taxonomy,” IJCSNS International Journal of Computer Science and Network Security, vol. 12, no. 1, January 2012, pp. 1–14.

[11] W. Lee, M. Miller, S. J. Stolfo, W. Fan, and E. Zadok, “Toward cost-sensitive modeling for intrusion detection and response,” Journal of Computer Security, vol. 10, 2002, pp. 5–22.

[12] A. Shameli-Sendi, J. Desfossez, M. Dagenais, and M. Jabbarifar, “A retroactive-burst framework for automated intrusion response system,” Journal of Computer Networks and Communications, vol. 2013, 2013.

[13] 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,” in Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on, June 2005, pp. 508–517.

[14] Z. Wu, D. Xiao, H. Xu, X. Peng, and X. Zhuang, “Automated intrusion response decision based on the analytic hierarchy process,” Knowledge Acquisition and Modeling Workshop, December 2008, pp. 574–577.

[15] Standards for Security Categorization of Federal Information and Information Systems, “Fips pub 199 standards for security categorization of federal information and information systems,” Federal Information Processing Standarts Publication, 2004.

[16] I. Balepin, J. Rowe, and K. Levitt, “Using specification-based intrusion detection for automated response,” Recent Advances in Intrusion Detection, vol. 2820, 2003, pp. 136–154.

[17] P. Mell, K. Scarfone, and S. Romanovsky, “A complete guide to the common vulnerability scoring system version 2.0,” 2013.

[18] T. Toth and C. Kruegel, “Evaluating the impact of automated intrusion response mechanisms,” Computer Security Applications Conference, 2002, pp. 301–310.

 ISBN: 978-1-61208-398-8