A defense-centric model for multi-step attack damage cost evaluation

Alireza Shameli-Sendi[1], Habib Louafi[2], Wenbo He[1], Mohamed Cheriet[2]
[1]*Department of Computer Science, McGill University, Montreal, Canada*
[2]*Synchromedia Lab, University of Quebec (ETS), Montreal, Canada*
{*alireza.shameli-sendi,wenbohe*}*@cs.mcgill.ca, habib.louafi.1@ens.etsmtl.ca, mohamed.cheriet@etsmtl.ca*

***Abstract*—Measuring the attack damage cost and monitoring the sequence of privilege escalations play a critical role in choosing the right countermeasure by Intrusion Response System (IRS). The existing attack damage cost evaluation approaches inherit some limitations, such as neglecting the dependencies between system assets, ignoring the backward damage of exploited non-goal services, or omitting the potential damage toward the goal service. In this paper, we propose a defense-centric model to calculate the damage cost of a multi-step attack. The main advantage of this model is providing an accurate damage cost by considering not only the damaged services (non-goal services) but also the potential damage toward the attacker target (goal service). To track the attacker's progress and find the attack path, an Attack-Defense Tree (ADT) is used. The model has been implemented in, but is not limited to, the cloud environment and tested with a multi-step attack scenario.**

***Keywords*-Multi-step attack; Vulnerability; Attack damage cost; Defense-centric;**

I. Introduction

Attackers are smart people in reality and they know that they may not achieve their malicious goals if they just follow the standard steps to compromise the target. They attempt to find the vulnerable services and defense points of the network and bypass them to launch multi-step attacks in order to compromise the target [1], [14]. To track the attacker's progress and find the attack path toward the attacker's target, there are two techniques based on network vulnerabilities: *attack graph* [2], [15] and *attack tree* [3], [16]. They provide an appropriate picture of different ways for compromising a target by exploiting a composition of vulnerabilities.

In both techniques, there are two models to calculate the multi-step attack damage cost [4]: *attack-centric* and *defense-centric*. In the former, there is one goal (e.g., a sensitive file modification) and all other completed steps are called non-goal. In this model, if the attacker does not reach the target, the attack damage cost is zero. In the defense-centric model, the damage cost is measured regardless of whether or not the attacker will compromise the target.

The main contribution of this paper is proposing a defense-centric model to take into account not only the current damage but also the potential damage of the multi-step attack. The former is the damage incurred on compromised services (non-goal services) and the latter represents the attack damage in the next steps of attack (non-goal/target).

The rest of this paper is organized as follows: Section II provides the background and literature review. The attack-defense tree and damage cost evaluation model will be discussed in Section III and IV, respectively. Section V discusses the experimental results. Finally, Section VI concludes the paper.

II. Related Work

Several tree and graph-based approaches have been proposed to quantitatively assess the damage cost of multi-step attack. Most of them use ”Common Vulnerability Scoring System (CVSS)” as the probability of successful vulnerability exploitation. The probability is propagated through the attack graph according to the relationship between exploits which can be disjunctive or conjunctive.

Wang et al. [9] uses dependency attack graphs rather than state-based attack graphs to represent network observations. The proposed approach systematically integrates attack graphs and Hidden Markov Models together for exploring the probabilistic relation between system observations and state. Kanoun et al. [10] presented a risk assessment model based on attack graphs to evaluate the severity of the total risk of the monitored system. The LAMBDA [11] language is used to model attack graphs when an attack is detected. Jahnke et al. [12] present a graph-based approach for modeling the effects of attacks against services, and the effects of the response measures taken in reaction to those attacks. The proposed model considers different kinds of dependencies between services, and derives quantitative differences between system states from these graphs. Kheir et al. [13] propose a service dependency graph to evaluate the confidentiality and integrity impacts, as well as the availability impact. The confidentiality and integrity criteria are not considered in [12].

Most of the existing works only consider the potential damage cost toward the target regardless of taking into account the backward damage impact on the dependent services to the compromised service [5]–[7]. In this paper, we propose a risk assessment model by taking advantage of the backward and forward propagation concepts, without the aforementioned limitation.

III. Attack-Defense Tree

The attack tree tackles the state space explosion problem, which is present in the graph, and consequently addresses the problem of visualization complexity [8]. In this paper, we use the *Attack-Defense Tree* (ADT) which consists of two types of actions: *attacker action* and *defender action*. We define our ADT as follows:

Definition 1 (Attack-Defense Tree (ADT)). *The attack-defense tree we propose is a 5-tuple ADT=$\langle s_g, S_{ng}, V, C, \mathcal{P}\rangle$, where:*

1) s_g is the root of the ADT. It represents the attacker-targeted service (goal service).
2) $S_{ng} = \{s_i\}_{i=1}^{d} - \{s_g\}$ represents the set of services that can be compromised (non-goal services) to reach the attacker's target.
3) $V = \overset{d}{\underset{i=1}{\cup}} V(s_i)$ is the set of vulnerabilities that are present in the network topology. The subset $V(s_i) = \{v_j(s_i)\}_{j=1}^{z}$ represents the set of s_i vulnerabilities.
4) $C = \overset{n}{\underset{i=1}{\cup}} C(v_i)$ is a set of conditions associated with all the vulnerabilities. For a given vulnerability v_i, let $C(v_i) = \{c_j(v_i)\}_{j=1}^{m}$ be the set of its conditions. To exploit v_i, the predicate $\underset{k}{\vee}\left(\underset{t}{\wedge} c_t(v_i)\right)$ should be true. That is, at least one term of this disjunction, which is a conjunction of conditions, should be satisfied.
5) $\mathcal{P} = \{DP_i\}_{i=1}^{p}$ is a set of defense points placed in the network to protect services. Note that, each DP_i may protect one or many different paths (services) toward s_g. We define χ as mapping between each defense point and the subset of services it protects. Formally, it is given by:

$$\begin{aligned} \chi : \mathcal{P} &\rightarrow S_{ng} \\ DP_i &\mapsto S_{ng}^{'} \end{aligned} \tag{1}$$

where $S_{ng}^{'} \subseteq S_{ng}$.

As depicted in Figure 2, generally, the attacker compromises a service (non-goal) to reach his target. To do so, he has to exploit one or more of the vulnerabilities of the non-goal service(s). For the latter to be exploited, some conditions have to be satisfied, such as *”running service x version y on machine A”* and *”connecting attacker machine to machine A”*. Each service, including the goal, possesses one child that represents the defense point protecting it. In some cases, the service may possess a path comprising a set of defense points.

IV. Damage Cost Evaluation

We evaluate the damage cost as the current and potential impact on services in terms of CIA (confidentiality, integrity, and availability). To this end, a service dependency graph is used. We model the relationship between services by means of a dependency weighted directed graph, the vertices being the services, and the edges being the functional dependency between them. To each vertex s_k in the graph, we associate one value, which is the importance $I(s_k)$ of the service.

$I(s_k)$ is a vector that contains the importance in CIA, from the point of view of the company. Besides, to each edge between s_i and s_k in the graph (s_i depends on s_k), we associate a vector that consists of three weight values that represent the dependency severity $d_{i,k}^{\{C,I,A\}} \in [0,1]$ in terms of CIA. The dependency severity is a score representing how strong the relationship between s_i and s_k is, which reveals how much the damage incurred on s_k will affect s_i. Thus, the total attack damage cost for the whole system (S) from the exploited vulnerability v_j in service s_k is the ratio of the importance of the services damaged by attacker to the total value (importance) of all services in terms of CIA. This cost is calculated as follows:

$$D(S, s_k, v_j) = \frac{\displaystyle\sum_{\delta \in \{C,I,A\}} D_\delta(S, s_k, v_j)}{\displaystyle\sum_{s_i \in S} \sum_{\delta \in \{C,I,A\}} I_\delta(s_i)} \tag{2}$$

where $D_C(S, s_k, v_j)$, $D_I(S, s_k, v_j)$, and $D_A(S, s_k, v_j)$ are attack damage cost in terms of confidentiality, integrity, and availability, respectively. $I_C(s_i)$, $I_I(s_i)$, and $I_A(s_i)$ represent the importance of the service s_i in terms of confidentiality, integrity, and availability, respectively. The attack damage cost for each attribute (C, I, A) is evaluated as follows:

$$D_\delta(S, s_k, v_j) = D_\delta^c(S, s_k, v_j) + D_\delta^p(S, s_k, v_j) \tag{3}$$

where $D_\delta^c(S, s_k, v_j)$ and $D_\delta^p(S, s_k, v_j)$ are the current and potential attack damage costs, respectively.

To calculate the current damage cost, direct and backward propagation are considered. In the backward sub-graph of service s_k, $B(s_k)$, the damage propagation is considered for all services that have mandatory dependency on s_k directly or indirectly. Thus, the current damage cost is calculated as follows:

$$D_\delta^c(S, s_k, v_j) = \overline{D_\delta^c}(S, s_k, v_j) + \overleftarrow{D_\delta^c}(S, s_k, v_j) \tag{4}$$

where $\overline{D^c_\delta}(S, s_k, v_j)$ and $\overleftarrow{D^c_\delta}(S, s_k, v_j)$ are the direct and backward propagated attack damages, respectively. They are given by:

$$\begin{aligned}\overline{D^c_\delta}(S, s_k, v_j) &= A^c_m(s_k, A^c_t(v_j))I(s_k)\\ \overleftarrow{D^c_\delta}(S, s_k, v_j) &= \sum_{s_i \in B(s_k)} A^{bc}_m(s_{i+1} : s_i)d_{i,i+1}I(s_i)\end{aligned} \tag{5}$$

where $A^c_m(s_k, A^c_t(v_j))$ is the attack impact on s_k, which depends on the type of attack (A^c_t), when vulnerability v_j is exploited (as seen in Eq. 6).

$$A^c_m(s_k, A^c_t(v_j)) = \begin{cases}[1,0,0] \text{ if } A^c_t(v_j) = \text{information leakage}\\ [1,1,1] \text{ if } A^c_t(v_j) = \text{remote-2-root}\\ [0,1,0] \text{ if } A^c_t(v_j) = \text{integrity}\\ [0,0,1] \text{ if } A^c_t(v_j) = \text{denial-of-service}\\ \ldots\end{cases} \tag{6}$$

$A^{bc}_m(s_{i+1} : s_i)$ is the backward attack impact propagation from s_{i+1} to s_i. It is calculated as follows:

$$A^{bc}_m(s_{i+1} : s_i) = \begin{cases}A^c_m(s_{i+1}, A^c_t(v_j)) \text{ if } s_{i+1} = s_k\\ A^{bc}_m(s_{i+1})d_{i+1,i+2} \text{ else}\end{cases} \tag{7}$$

Since the attack starts from s_k, the impact on all the parents of s_k (in service dependency graph) is equal to the impact on s_k, $A^c_m(s_k, A^c_t(v_j))$. For the other services (e.g., s_i) that depend to the parents of s_k, the backward attack impact propagation depends on the type of attack impact on the service s_{i+1} and CIA dependency vector between s_{i+1} and s_{i+2}. The attack impact on CIA in s_k cannot be propagated to all services in the backward sub-graph. At each step of the calculation, the impact propagation value should be moderated by the CIA dependency between services.

To perform the potential damage cost calculation, we follow the forward dependency direction from the service s_k, $F(s_k)$, to identify all the services that can be affected in the next step of the multi-step attack. The potential damage cost is calculated as follows:

$$D^p_\delta(S, s_k, v_j) = \sum_{s_i \in F(s_k)} \overline{D^p_\delta}(S, s_i, v^*_h) + \overleftarrow{D^p_\delta}(S, s_i, v^*_h) \tag{8}$$

where $\overline{D^p_\delta}(S, s_i, v^*_h)$ and $\overleftarrow{D^p_\delta}(S, s_i, v^*_h)$ are the potential attack damages on service s_i, which is exactly one step ahead from the compromised service (s_k), and the backward propagated attack damage from it, respectively. They are given by:

$$\begin{aligned}\overline{D^p_\delta}(S, s_i, v^*_h) &= A^p_m(s_i, A^p_t(v^*_h))I(s_i)\\ \overleftarrow{D^p_\delta}(S, s_i, v^*_h) &= \sum_{s_j \in B(s_i) - \{s_k\}} A^{bp}_m(A^p_m(s_{j+1} : s_j))d_{j,j+1}I(s_j)\end{aligned} \tag{9}$$

where $A^p_m(s_i, A^p_t(v^*_h))$ is the attack impact on s_i when v^*_h is exploited. v^*_h is the vulnerability that creates the highest damage on s_i once it is exploited. It is given by:

$$v^*_h = \arg\max_{v_h \in V(s_i)} A^p_t(v_h) \tag{10}$$

$A^{bp}_m(A^p_m(s_{j+1} : s_j))$ is the backward attack impact propagation from s_{j+1} to s_j calculated using Eq. 7.

V. Experimental Results

In the following, we present our experimental results and show the feasibility of our approach. We also discuss the implementation of our framework and its integration within an open source cloud framework, i.e. OpenStack [1].

A. Simulation Setup and Integration in Cloud

In order to show the feasibility of our approach and test it in a real cloud environment, we integrated our framework in Openstack. To validate the proposed security framework, we use a cloud network topology consisting of three servers connected to the Internet through a physical openFlow-capable switch[2], as seen in Figure 1. The cloud server consists of virtual machines (VM), open vSwitchs (OVS)[3], network-based intrusion detection systems (NIDS). The NIDS is installed in either Dom0 or DomU in each cloud server and sniffs a mirroring port in the OVS. The control functions of both OVS and OFS are integrated into the SDN controller. Our security modules are implemented in the centralized control layer.

The first cloud server hosts two VMs, used as two web servers ($WS1$ and $WS2$), that are connected to a virtual switch ($OVS1$). The second cloud server hosts two VMs, the first one hosts a web server ($WS3$); the second one hosts a mail server ($MS1$). Similarly, these VMs are connected to a virtual switch $OVS2$. The third cloud server hosts two VMs, used as a database server ($DBS1$) and a file server ($FS1$) that are connected to a virtual switch $OVS3$. As seen in Figure 1, the physical OpenFlow-capable Switch (OFS) is responsible for managing the communication between cloud servers. Each VM possesses a set of vulnerabilities, which are represented in Table I.

[1] http://www.openstack.org

[2] http://www.openflow.org/wp/learnmore/

[3] http://openvswitch.org/

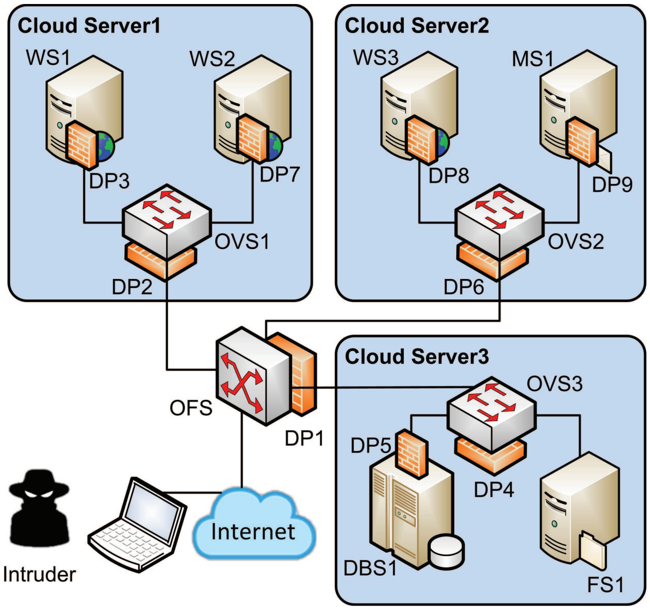

Figure 1: Virtual network topology used in the validation

Table I: VMs and their vulnerabilities

Server	VM	CVE	Vulnerability
Cloud Server 1	WS1 (Apache)	v_1: CVE-2007-1741 v_2: CVE-2014-6271 v_3: CVE-2012-4558 v_4: CVE-2014-0098	Execute code gain privileges Remote code execution XSS vulnerability DoS
	WS2 (IIS)	v_5: CVE-2009-1535 v_6: CVE-2009-3023	WebDAV Authentication Bypass Memory Corruption
Cloud Server 2	WS3 (Apache)	v_7: CVE-2014-6271 v_8: CVE-2012-0883 v_9: CVE-2014-0098	Remote code execution Gain privilege DoS
	MS1	v_{10}: CVE-2004-0840 v_{11}: CVE-2001-1030	Remote code execution Squid port scan
Cloud Server 3	FS1	v_{12}: CVE-2007-5616 v_{13}: CVE-2001-0755	Buffer overflow Buffer overflow
	DBS1	v_{14}: CVE-2008-5416 v_{15}: CVE-2008-0107	DoS Memory Corruption

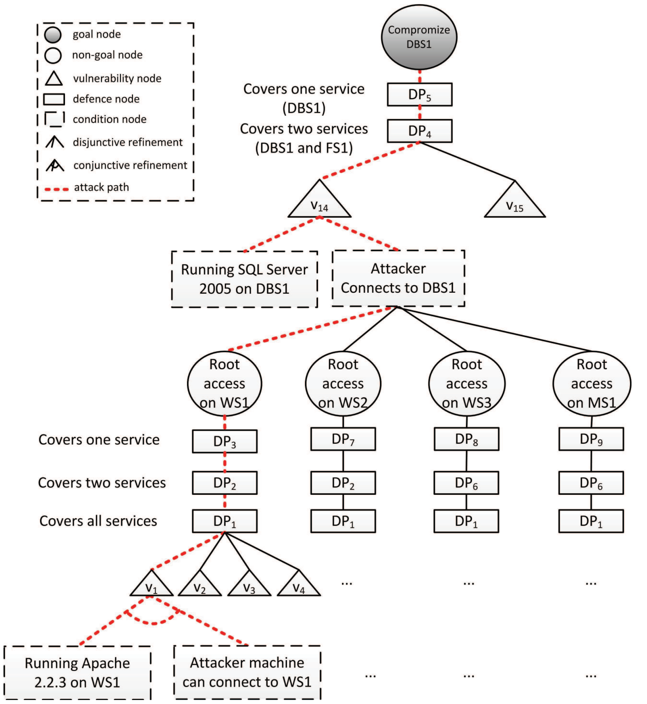

Figure 2: Attack Scenario in Attack-Defense Tree.

B. Attack Scenario

Figure 2 shows the Attack-Defense Tree for the virtual network topology. As seen, there are many attack paths to compromise the database. We programmed the Snort IDS to generate alerts with CVE [4] id, which can be captured and mapped by our model to the ADT. In this scenario, we consider an attack path in which v_1, CVE-2007-1741, is exploited in the first step in Apache 2.2.3. Then, the attacker uses the connection to $DBS1$ to compromise it.

C. Damage Cost Evaluation

Figure 3 shows the damage cost calculation result, which consists of current and potential damages, in details. Figure 4 illustrates the service dependency graph of the proposed virtual network topology. Since no service uses HR, the current backward damage from HR is zero, while the backward damage from DB is high. When the vulnerability v_1 in $WS1$ is exploited, the current and potential damages are 2.4 and 7.45 in terms of CIA, respectively. So, the total damage cost is 9.85. As shown in Figure 4, the total value of services in our framework is 14.1, in terms of CIA. Therefore, the total damage cost for the whole system from the exploited vulnerability is 69% (see Eq. 2), which represents the high damage cost. Thus, for this attack scenario, a strong countermeasure should be selected and deployed on the appropriate defense points DP_1 to DP_5. Note that the selected countermeasure and defense point should maximize the network security and minimize the impact on services.

VI. Conclusion

Balancing the countermeasure and attack damage costs efficiently leads a reliable defense framework. If we fail to do so, our automated intrusion response system will reduce network performance and wrongly disconnect legitimate users from a network. As the main contribution in this paper, we propose the backward and forward propagation concepts to assess the multi-step attack damage cost precisely. Measuring the right propagation of attack damage in the network is a key to select the appropriate countermeasure so as to minimize the impact on service.

[4] http://cve.mitre.org/

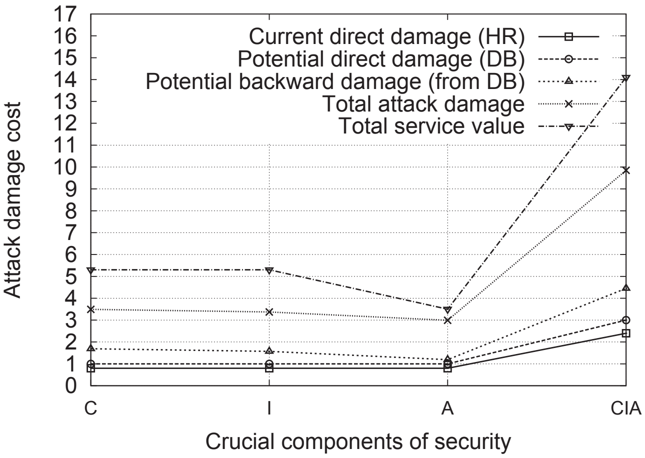

Figure 3: Attack damage cost, based on the current and potential damage costs on services, compared to the service value in terms of CIA.

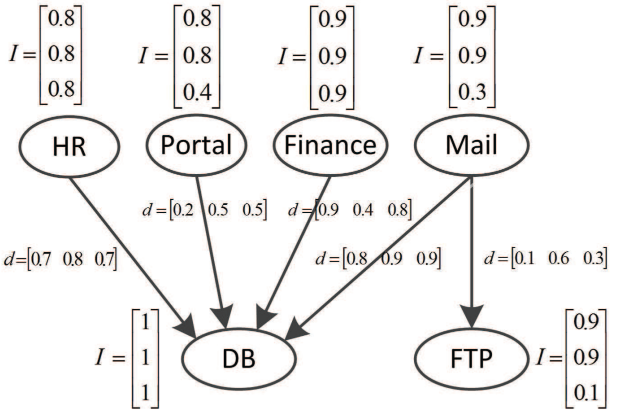

Figure 4: Service dependency graph of the proposed virtual network topology.

Acknowledgment

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

References

[1] A. Shameli-Sendi, M. Cheriet, and A. Hamou-Lhadj, "Taxonomy of intrusion risk assessment and response system," Computers & Security, vol. 45, pp. 1-16, 2014.

[2] P. Ammann, D. Wijesekera, and S. Kaushik, "Scalable, graph-based network vulnerability analysis," In Proceedings of the 9th ACM Conference on Computer and Communications Security, pp. 217-224, 2002.

[3] A. Roy, D. S. Kim, and K. S. Trivedi, "Attack countermeasure trees (ACT): towards unifying the constructs of attack and defense trees," Security and Communication Networks, vol. 5, no. 8, pp. 929-943, 2012.

[4] S. Zonouz, R. Berthier, H. Khurana, W. Sanders, and T. Yardley, "Seclius: An Information Flow-based, Consequence-centric Security Metric," IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 2, 2013.

[5] C.-J. Chung, P. Khatkar, T. Xing, J. Lee, and D. Huang, "NICE: Network intrusion detection and countermeasure selection in virtual network systems," IEEE Transactions on Dependable and Secure Computing, vol. 5, no. 4, pp. 198-211, 2013.

[6] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic security risk management using bayesian attack graphs," IEEE Transactions on Dependable and Secure Computing, vol. 9, no. 1, pp. 61-74, 2012.

[7] S. Noel, S. Jajodia, L. Wang, and A. Singhal, "Measuring security risk of networks using attack graphs," International Journal of Next Generation Computing, vol. 1, pp. 135-147, 2010

[8] R. Dewri, I. Ray, N. Poolsappasit, and D. Whitley, "Optimal security hardening on attack tree models of networks: a cost-benefit analysis," International Journal of Information Security, vol. 11, no. 3, pp. 167-188, 2012.

[9] S. Wang, Z. Zhang, and Y. Kadobayashi, "Exploring attack graph for cost-benefit security hardening: A probabilistic approach," Computers & Security, vol. 32, pp. 158-169, 2013.

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

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

[12] M. Jahnke, C. Thul, and P. Martini, "Graph-based Metrics for Intrusion Response Measures in Computer Networks," Proceedings of the 3rd LCN Workshop on Network Security, pp. 1035-1042, 2007.

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

[14] A. Gehani and G. Kedem, "Rheostat : Real-time risk management," Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection, pp. 15-17, 2004.

[15] S. Jha, O. Sheyner, and J. Wing, "Two formal analyses of attack graphs," In Proceedings of the 15th Computer Security Foundations Workshop, pp. 49-63, 2002.

[16] B. Kordy, S. Mauw, S. Radomirovi, and P. Schweitzer, "Foundations of AttackDefense Tree," In: FAST. LNCS. Springer, Heidelberg, 2010.