Open Access License Notice

Evidential Reasoning in Real-time Monitoring of Computing Systems

Bel G Raggad [1]
braggad@pace.edu

Alyssa Akermi [2]
alyssa.akremi@gmail.com

[1] Pace University, New York
[2] Technische Universität München, Germany

Abstract - We propose and demonstrate the construction of a belief structure based on data captured by a monitoring and intrusion detection system on the state variables defining the computing behavior of critical assets in a computing environment. We also propose a security risk model, using Dempster and Shafer theory, capable of predicting the occurrence of undesired events that can compromise the security of the entire computing environment through the compromising of one of its critical assets.

We adopt a more comprehensive definition of information security risks based on the notion of plausibility of undesired events which covers the potential for the realization of unwanted negative consequences of events, but also any other uncertain conditions that may involve both negative or positive effects due to the presence of ambiguity. This method facilitates incorporating the impact on security risk in the computing environment of planned incident responses that pertain to multiple and priory unknown threats. We can then, because of the real-time management of the asset's state variables, devise a security program without the exact knowing of all the threats that produce the undesired events. Our model can then predict undesired events and plan risk-driven responses without all the details of the threats currently menacing the computing environment.

Categories and Subject Descriptors

K.5.2 [Legal Aspects of Computing]: Governmental Issues – Regulation.

General Terms

Legal Aspects, Security

Keywords

Belief Function, Intrusion Detection System, Security Risk, Incident Response, Dempster and Shafer Theory.

1. INTRODUCTION

Security risk management in a computing environment is not that simple. The security manager has to make sure that this environment is secure with all its components including its personnel, its activities, its networks, its data resources, and its technology with its hardware and software. All those components are examples of assets that may be exposed to risk during information processing, information transmission, information storage, network operations, and so on. The security manager has to know all possible threats and their likelihood information, existing vulnerabilities and all different ways the threats can exploit those vulnerabilities, and resulting consequences, with business impacts. The security risk management process may then be very complex.

The security manager can easily identify those critical assets without which the computing environment cannot be secured unless they are secure. The security policy of the computing environment may be written in terms of the security policies of its critical assets. The security policy of a critical asset may be translated into a hyper table where we can record the values of a selected subset of critical state variables. The values of the state variables of an asset may serve as indicators that are triggered when undesired events occur.

We can then implement a monitoring hyper table that stores the values of state variables that we adopt to describe the computing behavioral activities of the asset. The values taken by the state variables will indicate all types of events characterizing the computing behavior of the critical asset.

In this article, as [4], a system's security policy is defined as the acceptable behavior of this system as defined by its owners. In this sense, the subset of security policy defining the acceptable computing behavior of a given asset is simply a subset of set-valued tuples defining all the known events which are known to produce acceptable outcomes.

The security policy is viewed as a live and an iterative rule base that stores the acceptable computing behavior of the asset that is revised in a periodic manner. The computing behavior of the asset is captured in live state variables the values of which may be associated with three subspaces of hyper (set-valued) tuples E+, E-, and E0. E+ denotes the known and desired events with acceptable outcomes. E- denotes the known but undesired events with undesirable events. E0 denotes the unknown events with unknown outcomes. The outcomes of the unknown events in E0 may be acceptable or undesirable.

The security policy will however in an iterative manner learn new computing behavioral patterns that can be recognized and added to the security policy as desired or undesired events. Over time, the subspace of unknown events will be reduced by classifying some its members as either desirable or undesirable.

We will throughout this paper model the security of a critical asset using a simple belief structure built on a simple frame of discernment $\Theta = \{\theta, \neg\theta\}$ where θ denotes the assertion that the asset is protected against the existing security policy as defined by its owners, and $\neg\theta$ denoting the opposite assertion.

2. OUR CONTRIBUTIONS

Security risk techniques can be quantitative, qualitative, or a combination of both. The quantitative techniques rely most often on expected value analysis where security risk exposure is expressed in terms of the likelihood of the incident and its consequences [5].

In addition to the intended major contribution of this article that demonstrates the construction of a belief structure based on the state values captured in a computing behavior data set of a monitored critical asset in a computing

environment, we also propose a security risk model capable of predicting the occurrence of undesired events that can compromise the security of the entire computing environment through the compromising of one of its critical assets. Our contribution uses Dempster-Shafer Theory [6, 7] to model uncertainties involved in the monitoring process of critical assets in a computing environment where belief functions are computed for all the undesired events that would compromise its critical assets.

Our contribution also adopts a more comprehensive definition of information security risks based on the notion of plausibility of undesired events which covers the potential for the realization of unwanted negative consequences of events, but also any other uncertain conditions that may involve both negative or positive effects due to the presence of ambiguity.

We also provide a structured approach to incorporating the impact of risk factors and the impact of the planned incident responses on information security risk associated with the computing environment and its critical assets. That is, our risk approach facilitates incorporating the impact on security risk in the computing environment of planned incident responses that pertain to multiple and priory unknown threats. We can then, because of the real-time management of the asset's state variables, devise a security program without the exact knowing of all the threats that produce the undesired events. Our model can then predict undesired events and plan risk-driven responses without all the details of the threats currently menacing the computing environment.

3. THE EVIDENTIAL MODEL

Our evidential model, as shown in Figure 1, consists of mainly four steps: 1) Capture of the computing behavior of the critical asset, 2) Construction of a belief structure based on the captured data set describing the current computing behavior of the critical asset, 3) Computation of security risks, and 4) Planning risk-driven incident responses.

At a first step, the monitoring system captures, in a timely manner, the values of state variables describing the computing behavior of the critical asset. We then assess the strength of evidence which evaluates the level of support the captured data gives to the space of state variables. As explained above, this space is divided into the three subspaces E-, E0, and E+ defined respectively as the subspace of undesired events hyper tuples, the hyper tuples with unknown outcomes, and the desired events hyper tuples. The strength of evidence is represented by the basic belief assignments defining the belief structure in question.

In a second step, we construct a belief structure using the Dempster and Shafer theory on possible outcomes associated with the captured data set expressed in terms of domain hyper tuples associated with undesired events, unknown events, and desired events.

In a third step, we compute the security risks associated with undesired and unknown events. Once security risks are obtained, the security manager can then start, in a fourth step, planning the most appropriate incident responses. A reverse search in the intrusion detection system will lead to the identification of the original threats that caused the occurrences of undesired events [3, 9]. Once the threats are known, the intrusion detection database will produce the responses that minimize the security risks. For those undesired events that cannot be associated with known threats, a maximum security response has to be planned for the critical asset. This maximum security plan will apply security controls that restricts the activities of the asset to a minimal configuration that does not trigger the state variables that are indicators of undesired events.

3.1 Processing of the evidential model

We can represent an asset security policy as a feasible rule subset structured as a hyper table A defined on 2A1 x ... x 2A|A| where the Ai's define the domains of the attributes constituting the policy table for an asset A. We will use the letter A to represent at the same time the asset and its security policy table when there is no confusion in doing so.

We then now consider an asset A in a computing environment and assume that we can capture, using its monitoring and intrusion detection system, its current behavior in a data set D of the same structure as the its security policy table as defined above. The data set D consists of |D| hyper (set-valued) tuples Dj, j=1,|D| and |V| hyper (set-valued) columns Vi, i=1,|V|. This means that the values of attributes are subsets in their domains and not necessarily single values as in relational databases.

That is, each hyper tuple Dj takes its values in Dom(D) denoting the domain of D which is equal to 2V1 x ... x 2V|V|. For example, a hyper tuple may look like this: ({1, 2, a}, {c}, {d, h, 17}).

For any critical asset in our computing environment we, in a real time manner, capture its computing behavior in a data set D, then, we use its content to build a belief structure on undesired events for the purpose of planning effective security responses. These incident responses are to be planned in a timely manner to prevent any imminent attacks on the asset that can consequently compromise the security of the computing environment. These incident responses are designed based on security risks associated with the asset in question.

3.2 Defining the frame of discernment of asset behaviors

The computing environment includes monitoring processes for all its critical assets. For each critical asset, its monitoring process is designed to capture values for critical state variables describing important behavioral aspects of the asset. Once captured, the data will be studied in terms of possible values of the attributes defining possible computing behavioral activities whether they are desired or undesired. The frame of discernment is then made of the Cartesian product of the domains of the attributes needed to define the critical behavioral aspects of the critical asset.

Let Ω denote the frame of discernment in question. The space Ω is then the product A1, x ... x $A_{|A|}$. The captured data set D will be used to construct a belief structure on the frame of discernment Ω. The basic belief assignment m_D is then defined on 2^{Ω} or the product $E=2^{A1}$, x ... x $2^{A|A|}$.

The critical asset owner can partition the space E into three subspaces E0, E+ and E- where E0 contains all hyper tuples with unknown events, where E+ contains all hyper tuples defining desired events, and where E- contains all hyper tuples defining undesired events. The subspace E+ then contains the security policy table A.

3.3 Constructing the belief structure on the frame Ω

Before we further procced, we need to define a partial order relation Δ in 2^{Ω}, as follows:

For x and y in $2^{Dom(Ai)}$, xΔy if and only x ≤ y : we say that x provides full evidence support to y.

Given the frame of discernment Ω = Dom(A1) x ... x Dom($A_{|A|}$), we construct a belief structure on Ω based on the captured data set D on the current computing behavioral activities of the critical asset A as follows:

m_D: $2^{\Omega} \rightarrow$ [0 1] m_D (x) = $|s_D(x)| / |s_D(E)|$

Where: $s_D(x) = \{y \varepsilon D \text{ such that } y \Delta x\}$

$s_D(E) = \{\{y \varepsilon D \text{ such that } y \Delta x\}, x \varepsilon E\}$

The evidence on hand cab further distributed to the three subspaces E0, E+ and E- constituting the main evidence space E. We have then three types of evidence that are used in reducing uncertainty about the beliefs on assertions that the asset would see undesired events, desired events, and unknown events. The strength of evidence is represented by the basic belief assignments defining the belief structure in question but it will depend on to what extent the captured data set support the assertions associated with the undesired events, the desired events, and the unknown events.

3.4 Distributing evidence on assertions

The evidence on hand cab further distributed to the three subspaces E0, E+ and E- constituting the main evidence space E. We have then three types of evidence

that are used in reducing uncertainty about the beliefs on assertions that the asset would see undesired events, desired events, and unknown events. The strength of evidence is represented by the basic belief assignments defining the belief structure in question but it will depend on to what extent the captured data set support the assertions associated with the undesired events, the desired events, and the unknown events.

Computing $m_D(E+)$:

Let e+ in E+. Then $m_D(e+)$ is equal to $|s_D(e+)| / |s_D(E)|$

where:

$s_D(e+) = \{y \varepsilon D \text{ such that } y \Delta x\}$ $s_D(E) = \{\{y \varepsilon D \text{ such that } y \Delta x\}, x \varepsilon E\}$

Once we obtained the value of $m_D(e+)$ for e+ in E+, we can then compute the basic belief assignment for the entire subspace E+ as follows:

$m_D(E+) = |s_D(E+)| / |s_D(E)|$, where:
$s_D(E+) = \{y \varepsilon D \text{ such that } y \Delta x, x \varepsilon E+\}$
$s_D(E) = \{\{y \varepsilon D \text{ such that } y \Delta x\}, x \varepsilon E\}$

Computing $m_D(E-)$:

Let e- in E-. Then $m_D(e-)$ is equal to $|s_D(e-)| / |s_D(E)|$

where:

$s_D(e-) = \{y \varepsilon D \text{ such that } y \Delta x\}$ $s_D(E) = \{\{y \varepsilon D \text{ such that } y \Delta x\}, x \varepsilon E\}$

Once we obtained the value of $m_D(e-)$ for e- in E-, we can then compute the basic belief assignment for the entire subspace E- as follows:

$m_D(E-) = |s_D(E-)| / |s_D(E)|$, where:
$s_D(E-) = \{y \varepsilon D \text{ such that } y \Delta x, x \varepsilon E-\}$
$s_D(E) = \{\{y \varepsilon D \text{ such that } y \Delta x\}, x \varepsilon E\}$

Computing $m_D(E_0)$:

Let e_0 in E_0. Then $m_D(e_0)$ is equal to $|s_D(e_0)| / |s_D(E)|$

where:

$s_D(e_0) = \{y \varepsilon D \text{ such that } y \Delta x\}$ $sD(E) = \{\{y \varepsilon D \text{ such that } y \Delta x\}, x \varepsilon E\}$

Once we obtained the value of $m_D(e_0)$ for e_0 in E_0, we can then compute the basic belief assignment for the entire subspace E_0 as follows:

$m_D(E_0) = |s_D(E_0)| / |s_D(E)|$, where:
$s_D(E_0) = \{y \varepsilon D \text{ such that } y \Delta x, x \varepsilon E_0\}$
$s_D(E) = \{\{y \varepsilon D \text{ such that } y \Delta x\}, x \varepsilon E\}$

3.5 Determining security risks and feasibility of responses

Before planning any incident responses, a cost-benefit analysis should performed to prove that those responses are economically feasible. Even for critical assets, they can only be secured at a level that is consistent with a sound cost-benefit analysis.

In general, any incident response or a combination of countermeasures that are prescribed against predicted undesired events, should not cost more than the total revenues generated by the continuity of operations of the asset or higher than the expected cost of tolerating the compromise for which the responses are planned.

That is, in order to prescribe the most appropriate incident responses, a cost-benefit study is mandatory to mitigate any security risks associated with the predicted incident. We then need to proceed with reducing known uncertainties and resolving existing ambiguities before performing an expected value analysis that minimizes expected losses. [1, 2, 7]. We use Strat's approach [8] because it provides the worst and the best case scenarios which, in turn, provide the security manager a choice between the two extremes.

As in [7], we express security risk using the plausibility of undesired events. As you can see, this is a conservative measure of risk because it is based on the worst

case scenario where the ambiguity is automatically added to any direct evidence on the security risk.

Given the computations performed above, we then have $m_D(E-)$ and $m_D(E_0)$ expressing respectively the belief that undesired incidents or unknown events would take place. On the other hand, $m_D(E+)$ expresses the belief that desired events would occur.

As in [8], we use a parameter ρ, $0 \leq \rho \leq 1$, to represent the security manager attitude towards risks. A value of $\rho = 1$ indicates that the security manager loves risks while a value of $\rho = 0$ indicates that he / she is extremely risk averse. The parameter ρ allows the security manager to resolve the ambiguity by allocating part of the ambiguity, $\rho\ m_D(E_0)$ in favor of desired events and the remaining ambiguity, $(1-)m_D(E_0)$ to undesired events affecting the asset.

We now then need to revise the m values of basic belief assignments after the introduction of the ρ parameter representing the security manager's attitude towards risks as follows:

$$b_D(E+) = m_D(E+) + \rho m_D(E_0)\ b_D(E-) = m_D(E-) + (1-\rho)m_D(E_0)$$

We used the parameter ρ to plan how ambiguity is resolved. The security manager may apply $\rho = 0$ to plan for the worst scenario case, where the ambiguity $m_D(E_0)$ is assigned to undesired events. The security manager, on the other hand, may plan for the best scenario case, where all the ambiguity is assigned to the support of desired events, by setting $\rho = 1$.

Of course, incident responses cannot be planned if we cannot find out the threats at the origin of the undesired events. This article assumes that once an undesired event is identified we can determine the suspected threats of causing them. Most often, available intrusion detection systems include threat definition databases that determine the threats in terms of the values of state variables defining the undesired events [3, 9]. That is, we assume that the incident responses are planned in accordance with threat information stored in intrusion detection databases.

The plausibility of an assertion is the degree to which the assertion is plausible. In accordance with the evidence on hand. In Dempster and Shafer theory, the plausibility function for an assertion A, in a frame Ω, is defined as the highest possible belief that could be assigned to A if all future evidence were in support of A. That is, $Pl(A) = \Sigma_{A \cap B \neq \emptyset} m(B)$. The plausibility function may also be written as $Pl(A) = 1 - Bel(\neg A)$.

We now go back to our original discussion of asset protection. We modeled the security of a critical asset using a simple belief structure built on a simple frame of discernment $\Theta = \{\theta, \neg\theta\}$ where θ denotes the assertion that the asset is protected against the existing security policy as defined by its owners, and ¬θ denoting the opposite assertion.

We then need to compute the security risk of our critical asset A. This risk may be computed as the plausibility that the asset is not protected. We can compute security risks in two ways: (1) without taking into account of the security manager's attitude towards risks, or (2) with revising the security risks after adding the security manager's attitude towards risks:

(1) Risk (Asset not Protected) = $Pl(\neg\theta) = 1 - Bel(not(\neg\theta))$

$= 1 - Bel(\theta) = 1 - m_D(E+) = m_D(E-) + m_D(E_0)$.

(2) Revised Risk (Asset not Protected) = $b_D(E-)$

$= m_D(E-) + (1 - \rho)m_D(E_0)$.

4. PLANNING THE INCIDENT RESPONSE

The planning of incident responses was triggered by the prediction of upcoming undesired events. At the same time, when the security risk was computed, as above, the value of risk was higher than the maximum tolerated risk. Two situations may be present: the security manager identified the known threats that produced the evidence for the upcoming undesired events, or he / she could not identify known threats that have been at the origin of the evidence on hand. In the first situation, the threats are known to the monitoring and intrusion detection system and also

any incident responses that apply. The security manager will plan the incident responses accordingly.

In the second situation, there is no information available of possible known threats that could have triggered the undesired events. The security manager, in this case, has to conduct a brief audit to identify possible causes in the computing environment that affected the computing behavioral activities of the critical asset in question. In case more evidence could not be collected, the security manager has to reconfigure the critical asset to operate under a safety mode that does not push the values of the state variables back to abnormal states and that reduces security risk below an accepted. If such a safe configuration is not possible, the critical asset has to be reconfigured in any way that mitigate its security risks below the tolerated risk level as defined in its security policy, even if it has to be switched of.

5. NUMERICAL EXAMPLE

Let us consider a small computing environment in a travel agency consisting of a local area network connecting 3 servers and 2 workstations. We assume a monitoring and detection system that records selected primary key indicators on the performance of the computing environment, as follows:

Buffer Cache Hit Ratio: a: '>90%'; n: 'between 70% and 90%, included; and r: '<70% included'.

Avg Disk Que Length: a: '<2 included'; n: 'between 2% and 5%, included'; and r: '>5'.

Buffer Cache Hit Ratio: a: '<500 ms included'; n: 'between 500 ms and 700 ms, included; and r: '>700 ms'.

Full Scans / Sec: a: 'Baseline'; r: 'Baseline violated'

The data and the computed values are provided in Tables 1 to 7. We can compute security risks in two ways: (1) without taking into account of the security manager's attitude towards risks, or (2) with revising the security risks after adding the security manager's attitude towards risks:

(1) Risk (Asset not Protected) = 0.36

(2) Revised Risk (Asset not Protected) = $b_D(E-)$

$= 0.20 + 0.16(1- \rho)$

Table 1: Security Risk after taking into account of Risk Attitude		
Risk Attitude		Security Risk
Extreme Risk Seeker	ρ=1.00	0.20
Risk Seeker	ρ=0.75	0.24
Risk Neutral	ρ=0.50	0.28
Risk Averse	ρ=0.25	0.32
Extreme Risk Averse	ρ=0.00	0.36

Table 2: Undesired Events: E-		
	$\lvert s_D \rvert$	m_D
e_{-1}	1	0.04
e_{-2}	2	0.08
e_{-3}	2	0.08
$m_D(E-)$		0.20

Table 3: Unknown Events: E0		
	$\lvert s_D \rvert$	m_D
e_{01}	1	0.04
e_{02}	1	0.04
e_{03}	1	0.04
e_{04}	1	0.04
$m_D(E0)$		0.16

Table 4: Undesired Events: E-				
e_{-1}	{r}	{a, n, r}	{a, n, r}	{a, r}
e_{-2}	{a, r}	{a, n, r}	{a, n, r}	{a, r}
e_{-3}	{n, r}	{a, n, r}	{a, n, r}	{a, r}

Table 5: Unknown Events: E0				
e_{01}	{n, r}	{a, n, r}	{a, n, r}	{a, r}
e_{02}	{a, n}	{a, n, r}	{a, n, r}	{a, r}
e_{03}	{a, r}	{a, n, r}	{a, n, r}	{a, r}
e_{04}	{a, r}	{a, n, r}	{a, n, r}	{a, r}

Table 6: Hyper Data Set Capturing State Values of Assets				
A1	{n}	{a, n, r}	{a, n, r}	{a, r}
A2	{a, n, r}	{a, n, r}	{a, n, r}	{a, r}
A3	{r}	{a, n, r}	{a, n, r}	{a, r}
A4	{a}	{a, n, r}	{a, n, r}	{a, r}
A5	{n, r}	{a, n, r}	{a, n, r}	{a, r}
A6	{a, n}	{a, n, r}	{a, n, r}	{a, r}

Table 7: Captured Data on Critical State Variables on Assets in the Computing Environment				
Buffer Cache Hit Ratio	Avg Disk Que Length		Avg Wait Time in ms	Full Scans / Sec: Baseline?
A1	75	1	412	y
A1	78	3	460	n
A1	74	2	642	n
A1	78	5	820	y
A2	65	1	418	y
A2	92	3	260	n
A2	94	2	642	y

Table 7: Captured Data on Critical State Variables on Assets in the Computing Environment				
Buffer Cache Hit Ratio	Avg Disk Que Length	Avg Wait Time in ms	Full Scans / Sec: Baseline?	
A2	78	5	860	y
A3	65	1	718	y
A3	68	3	462	n
A3	54	2	684	n
A4	95	1	418	y
A4	92	3	760	n
A4	94	2	642	y
A5	62	3	260	n
A5	74	2	742	y
A5	74	5	642	y
A6	52	3	368	n
A6	74	2	642	n
A6	94	6	742	y

6. CONCLUSION

This article proposed and demonstrated the construction of a belief structure based on data captured by a monitoring and intrusion detection system on the state variables defining the computing behavior of critical assets in a computing environment. We also proposed a security risk model, using Dempster and Shafer theory, capable of predicting the occurrence of undesired events that can compromise the security of the entire computing environment through the compromising of one of its critical assets.

We adopted a more comprehensive definition of information security risks based on the notion of plausibility of undesired events which covers the potential for the realization of unwanted negative consequences of events, but also any other uncertain conditions that may involve both negative or positive effects due to the presence of ambiguity. This method facilitates incorporating the impact on security risk in the computing environment of planned incident responses that pertain to multiple and priory unknown threats. We can then, because of the real-time management of the asset's state variables, devise a security program without the exact knowing of all the threats that produce the undesired events. Our model can then predict undesired events and plan risk-driven responses without all the details of the threats currently menacing the computing environment.

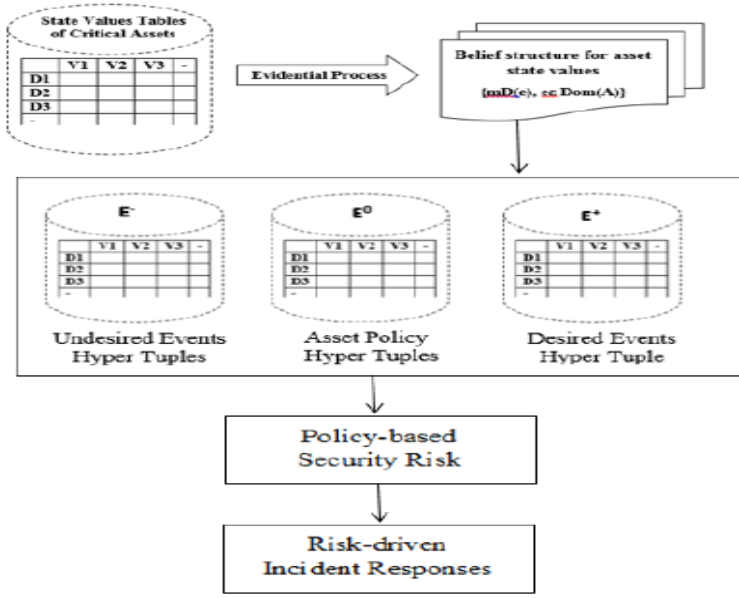

Figure 1: Word cloud of comments on the NPRM.

REFERENCES

[1] Jaffray, J-Y. Dynamic decision making with belief functions. In, Yager, R. R.; Fedrizzi, M.; and Kacprzyk, J., (eds.), Advances in the Dempster-Shafer Theory of Evidence, New York, NY: Wiley, 1994, 331-352. 24.

[2] Jaffray, J-Y. Utility Theory for belief functions. Operations Research Letters, 8 (1989), 107- 12. 47.

[3] Peddabachigari, S., Ajith Abraham, C. Grosan, J. Thomas, "Modeling intrusion detection system using hybrid intelligent systems", Journal of Network and Computer Applications, Volume 30, Issue 1, January 2007, Pages 114–132.

[4] Raggad, B., Information Security Management: Concepts and Practice, CRC Press, New York, 2010.

[5] Rainer, R. K.; Snyder, C. A., and Carr, H. H. Risk Analysis for information technology. Journal of Management Information Systems 8, 1 (1991), 129-147.

[6] Smets, P. The Combination of evidence in the transferable belief model. IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 5 (May 1990), 447-458. 50.

[7] Srivastava, R. P., and G. Shafer. Belief-Function formulas for audit risk. The Accounting Review, Vol. 67, No. 2 (April 1992), 249-283.

[8] Strat, T. M. Decision analysis using belief functions. International Journal of Approximate Reasoning, 4 (1990), 391-417.

[9] Zamboni, D."Using Internal Sensors For Computer Intrusion Detection". Center for Education and Research in Information Assurance and Security, Purdue University. August 2001.