IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 15, Issue 6, Ser. I (Nov - Dec 2020), PP 01-12
www.iosrjournals.org

Risk-driven Security Policy Enforcement: An Evidential Model

Bel G. Raggad[1], Abhi Pandya[2]
[1](Seidenberg School of CS&IS, Pace University, USA)
[2](College of Engineering and Computer Science, Florida Atlantic University, USA)

Abstract:
We propose a risk-driven security policy enforcement model using Dempster and Shafer's calculus of evidence. We define security policy, as in [15], to be the acceptable behavior of a system as defined by its owners. We assume information security management to have a corporate security policy made of auditable rules for which we apply security-monitoring sensors to detect violations of the security policy. We adopt security objectives and features on which security policy is structured and monitored using security sensors configured to generate raw data translated into belief structures needed to manage model uncertainty.

Belief structures are shaped using a tuned membership function that accounts for two generic types of evidence, favorable to corporate security or unfavorable to it. The raw data will be translated into belief structures on the security sensors, features, and objectives, which are later sequentially fused using Dempster rule to produce the corporate security posture and risks. Information security management can then evaluate security risks and devise a cost-effective risk-driven security program capable to bring current security risks below prescribed policy risks. Cost-effectiveness is achieved by stopping the search for security controls at the first set that produces a residual risk below policy risks.

Our model is intentionally designed in a generic manner, first, to allow for enough flexibility in defining the security policy, its features and objectives, and its monitoring process; and secondly, to provide a standardizedtemplate that information security management can adapt in the design of a security policy enforcement system capable of continual security. Our model will also serve as a mechanism for the validation of a security policy and as a risk-driven continual security process. We provide a numerical example to demonstrate the working of our evidential model and its claims.

Keywords: *Security policy, security features, layer-based, policy enforcement, Dempster and Shafer theory, evidence calculus, belief functions, basic belief assignment, security risk.*

Date of Submission: 16-11-2020 Date of Acceptance: 02-12-2020

I. Introduction

Security policy is often written to delineate the acceptable behaviors of systems as defined by owners. This policy has to be known to all components of the organization's computing environment, including its people, activities, data resources, technology, and networks [15]. It is the responsibility of information security management to configure all those components to fully comply with all security policy rules, and apply effective security monitoring systems to enforce the security policy. Moreover, all auditors have to be in agreement with information security management on the rules of compliance, theexact interpretations of security policy rules, and allowable security risks as prescribed by owners. This sort of alignment on security policy enforcement requirements should be observed by all stakeholders. All components of the computing environmentshould then be configured subject to this alignment. The security policy should be modeled with enough flexibility and structuredness to allow for feasible enforceability, compliance, and auditability.

Often we talk about customers and staff compliance with policy; but it is not always about people. However, this type of violations of security policy can be easily tracked and communicated, while on the other hand, there are plenty of other violations that take place in computing processes, as in data resources, networking activities, and technology that remain very difficult to define, communicate, monitor, detect, and correctas easy as in people-related security policy requirements.

When it comes to threats, there are then no differences between security breaches by people, or other components of the computing environment, or by information security management configurations faults. All these breaches will be translated into security risks that are compared to policy risks as defined by owners. When it comes to people, the direct reasons behind violations of the security policy may be attributed to, for example, people distrust, unawareness, carelessness, and so on; or there may be indirect reasons, like faults, misconfigurations, deficiencies, and vulnerabilities which are very common as violations of security policy. Neverthelsess, in order to plan their security policy enforcement effort, information security management employ a diverse range of monitoring and intrusion detection systems, like in SNORT, NIDS, HIDS [2], and similar systems that provide automatic real-time detection of incidents, including security attacks and breaches of security policy.There are certainly many challenges fronting information security management in designing the organization's information security policy itself and translating it into auditable security policy rules; and certainly even greater challenges in implementing, enforcing, and monitoring this security policy([3], [4], [11]).

This paper proposes a generic approach of security policy enforcement that allows for enough flexibility for defining the organization security objectives, the monitoring of its security policy rules, and the computing of its security posture and risks. We build a model that can serve as a template to design a well-structured security policy enforcement process. This paper proposes an evidential model for a risk-driven security policy enforcement program using Dempster and Safer's evidence theory ([16], [20]). Raw data is processed into evidence that is fused, using Dempster Rule, to produce the corporate security posture and the corporate security risk based on which information security management can devise a feasible risk-driven security program.We define security policy, as in [15], to be the acceptable behavior of a system as defined by its owners. Information security management is assumed to have a corporate security policy made of auditable rules for which monitoring sensors are applied to detect any violations of the security policy. We allow information security management to define a set of security objectives on the security policy, and define security features for which they implement monitoring sensors that generate data on policy enforcement [21].

Raw data generated by security monitoring sensors are processed using a membership function, as shown in Figure 3, that accounts for two generic types of evidence, favorable to corporate security or unfavorable to it. Information security management can define the corporate security objectives and reconfigure sensors as needed. Sensors can directly relate to security features when security policy rules are directly violated by components of the computing environment. Sensors can, for example, also directly monitor hosts, or spanning ports that record traffics that pass through monitored switches. There are however, sensors that relate to security features indirectly, for example, those sensors that collect http log data on hosts, or those sensors that can be configured to sniff tcp traffic on hubs [2]. Security features may be monitored by configuring sensors to monitor interfaces connecting routers to the Internet, routers to switches, and routers to hosts [19]. Of course, there are not only network-based sensors to apply for security policy monitoring, information security management should also monitor for vulnerabilities that compromise security features that lead to violations of security policy rules. There are many other types of threats that relate to physical attacks, vulnerabilities that lead to data tampering withnetworking. There are alsothreats related to software attacks that exploit vulnerabilities inside IoT applications, and encryption attacks that involve breaking system encryptions [14]. That is, the monitoring of human interactions with other components of the computing environment is not only needed with computing and communication devices, but also needed with physical production assets to mitigate functional integrity risks.

The raw data will be transformed into belief structures on the security objectives that are later fused using Dempster rule to produce the corporate security posture. Information security management can then evaluate the company security risks and devise a feasible security program capable to bring the current security risks below policy security risks.

II. A brief review of DST evidential theory

The Dempster–Shafer theory (DST) provides a mathematical framework for uncertainty management where all analysts use the same frame of discernment in studying a finite set of mutually exclusive outcomes about their decision domain. This framework is capable of combining evidence from different sources and produces a degree of belief, as a belief function, that takes into account all available evidence.The Dempster-Shafer theory started with Dempster in 1968 as statistical inference, but has been later formalized by Shafer, in 1976, as a theory of evidence ([5], [6]). Later after the 1980's, Smets reshaped it in his Transferable Belief Model before it started to see growing development in diverse AI applications in most domains ([17], [18]).

In similar decision domains, Dempster and Shafer theory should produce the same decision support as in Bayesian reasoning, but it is capable of a superior expressive power when information is incomplete or data is not of good quality.In order to model a belief structure for a decision domain with a frame of discernment Ω, we let the power set 2^{Ω} contain every mutually exclusive subsets of the frame of discernment Ω. A basic probability assignment m is used to allocate a belief value in [0, 1] for every hypothesis defined by the subsets in the frame of the discernment, as follows:

$m: 2^{\Omega} \rightarrow [0, 1]$; $m(\varnothing)=0$; $m(A)\geq 0$ for any A in 2^{Ω}; $\sum_{A \square \Omega} m(A) =1$.

If x is an unknown quantity with possible values in our frame of discernment Ω, we can add a piece of evidence about x using a mass function m on Ω. Any subset A of Ω with a mass greater than zero is called a focal set of m. You can see that this is different than in Bayesian theory where probability distributions only have singleton focal sets. When we have no evidence on x, we use the vacuous mass function, defined by $m_{\Omega}(x) = 1$, which represents a completely uninformative piece of evidence.Upper and lower probability can be obtained which will enclose the precise traditional probability the decision maker is seeking. This decision maker’s target is then bounded by two non-additive continuous measures that Dempster and Shafer Theory refers to as belief and plausibility. The belief for a subset of interest A is the sum of all the masses of the subsets x residing in A; and the plausibility of a subset A is, on the other hand, the sum of all the masses of the subsets x intersecting A.

A great expansion of Dempster and Shafer Theory is Dempster’s Rule of combination of evidence that is capable of computing the fused evidence obtained from multiple sources and the modeling of conflicts.Often, bodies of evidence come in small pieces obtained from different independent sources. While these bodies of evidences are included in the decision process using belief functions, the totality of the evidence is computed by combining the belief functions using Dempster Rule and its extensions. This rule consists of a mapping that considers multiple sources and produces a composite source that represents the combined impact of sources as one combined measure of belief. Given two independent sources of evidence defined on the same frame of discernment Ω and with basic probability assignments m_1 and m_2, we combine evidence as follows:

$$m_{\Omega}(A) = \sum_{B \cap C=A} m_1(B)m_2(C)/(1-K);\ \text{for } A \neq \emptyset$$

$$\text{Where } K= \sum_{B \cap C=\emptyset} m_1(B)m_2(C) \text{ and } m_{\Omega}(\emptyset)=0$$

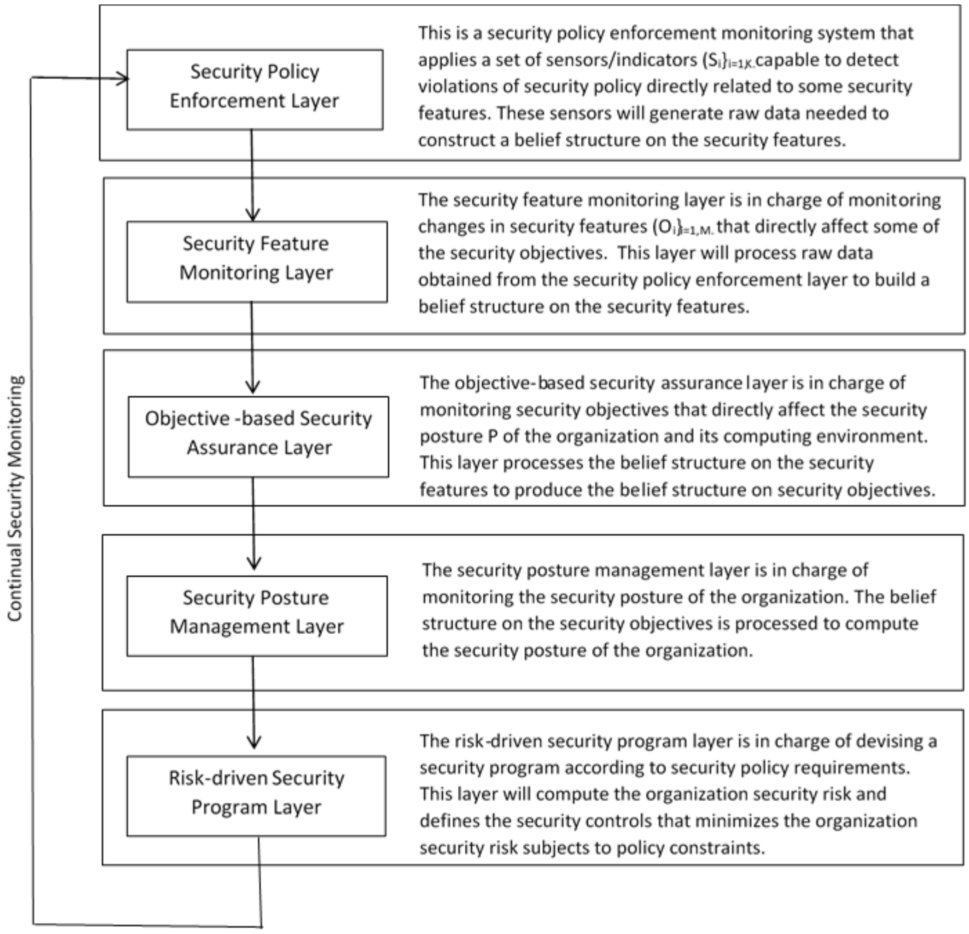
The parameter K represents the basic probability mass associated with the conflict between m_1 and m_2. It is computed as the sum of the products of the basic probability masses of all the disjoint sets from the tow sources of evidence.

III. Our layer-based security policy enforcement process

We are proposing a layer-based security policy enforcement approach, as depicted in Figure 1, thattranslates the steps information security management follows in adopting a risk-driven continual security program. This approach employs monitoring sensors to collect raw data on the security features used to define the organizational security objectives used to determine the security posture of the organization. Information security management will then assess security risks in terms of the security posture formerly computed. At a final step, information security management will recommend those security controls that are capable of cutting risks below the maximum security risks allowed by the corporate security policy as defined by owners. The five layers constituting the layered process, depicted in Figure 1, are introduced in this section.

The first layer is the security policy enforcement layerwhich is a security policy enforcement monitoring component. It applies a set of sensors/indicators $\{S_i\}_{i=1,N.}$capable to detect violations of security policy directly related to the security features $\{F_j\}_{j=1,M}$. These sensors will generate raw data q(i,j), i=1,N and j=1,M, needed to construct a belief structure on the security features.

The second layer is the security feature monitoring layerwhich is a security feature monitoring component in charge of monitoring changes in security features that directly affect the security objectives $\{O_k\}_{k=1,K}$. The third layer is the objective-based security assurance layer whichis an objective-based security assurance component in charge of monitoring security objectives that directly affect the security posture P of the organization and its computing environment. This layer processes the belief structure on the security features to produce the belief structure on security objectives. The fourth layer is the security posture management layerwhichis a security posture management component in charge of monitoring the security posture of the organization. The belief structures on the security objectives computed based on the security features’ belief structures are processed to compute the security posture of the organization. The final risk-driven security program layer is a risk-driven security program component in charge of devising a security program according to security policy requirements. This layer will compute the organization security risk and will define the security controls that minimize the organization security risk subject to policy constraints [8].


Security Policy Enforcement Layer
Security Feature Monitoring Layer
Objective -based Security Assurance Layer
Security Posture Management Layer
Risk-driven Security Program Layer
Continual Security Monitoring

This is a security policy enforcement monitoring system that applies a set of sensors/indicators $(S_i\}_{i=1,K.}$capable to detect violations of security policy directly related to some security features. These sensors will generate raw data needed to construct a belief structure on the security features.

The security feature monitoring layer is in charge of monitoring changes in security features $(O_i\}_{i=1,M.}$ that directly affect some of the security objectives. This layer will process raw data obtained from the security policy enforcement layer to build a belief structure on the security features.

The objective-based security assurance layer is in charge of monitoring security objectives that directly affect the security posture P of the organization and its computing environment. This layer processes the belief structure on the security features to produce the belief structure on security objectives.

The security posture management layer is in charge of monitoring the security posture of the organization. The belief structure on the security objectives is processed to compute the security posture of the organization.

The risk-driven security program layer is in charge of devising a security program according to security policy requirements. This layer will compute the organization security risk and defines the security controls that minimizes the organization security risk subjects to policy constraints.

Figure 1: Security policy enforcement layers

Managing security in a company's computing environment is an important activity to assure business continuity. Nonetheless, most of the information involved in processing information security is just too ambiguous to apply formal statistical analysis or Bayesian reasoning. We will however apply Dempster and Shafer Theory to model information security management in a corporate computing environment. In order to do so, we adopt a risk-driven objective-based information security management approach as shown in Equation 1.

$$C = \{\{P(O(F,S)) \rightarrow C^*(P,R)\} \rightarrow R^*, P^*\} \qquad (1)$$

Where C = Security Program; P = Security posture of the computing environment; O = Set of security objectives as defined by owners; F = Set of security feature subsets for objectives; S: Sensors; C* = Retained set of risk-driven security controls; R = Security risks for objectives; P* = Overall security posture after applying C*, R* = Overall security risk corresponding to P*.

Also, in order to provide sufficient flexibility to adopters, we modeled security parameters in a generic manner where parameters only take two values, favorable to the security of the company or unfavorable to it. This is then an open model where the adopter can include any variables that are relevant to the security of the company.Our security policy enforcement model enforces security policy rules by maintaining security risks produced by security policy violations below a tolerated security risk R^*prescribed by owners. Security risks are computed in a continuous manner and compared to policy risk and every time those security risks surpass this security policy risk then information security management proceeds to devising sufficient security controls to bring down security risks below policy risk R^*. Information security management should be able to determine a security program C^* that minimizes risks below R^*. Security risks are computed based on evidence obtained on the security objectives O, obtained by fusing evidence on security features F, which are in turn assessed based on raw data obtained from security monitoring sensors (S).

The purpose of the security program C is to define the security controls that maintain a security posture P* and the associated risk R* as prescribed in the company's security policy. In order to do so, we set our security objectives O={O_k}$_{k=1,K}$ as defined in the security policy. All the security features that are relevant to the objective O_k, are monitored by the corporate intrusion detection system or its security policy enforcement system. We assume that the company intrusion detection system or its security policy enforcement system is capable to monitor all the security features {Fj}, j=1,M, for all security objectives included in the security program. Figure 2 depicts the exchange of evidence among the security policy enforcement layers.

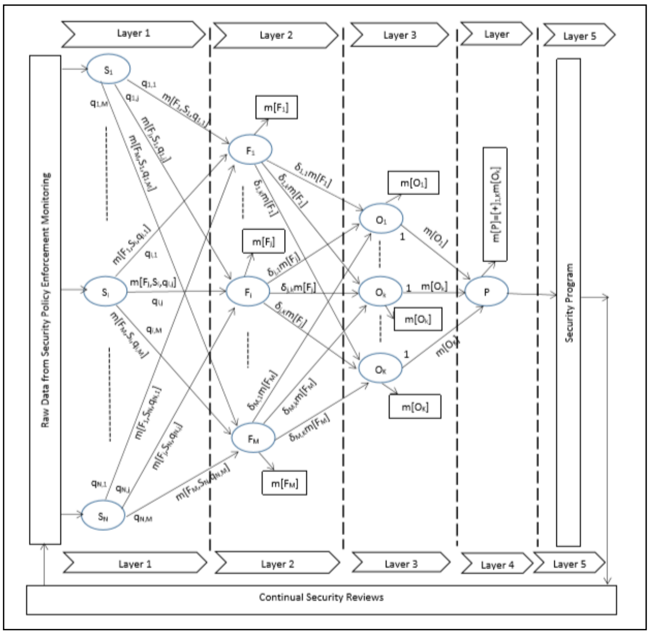


Figure 2: Evidence exchange among layers

IV. Buildingbeliefstructures

We are using is a simple method to construct basic belief assignments from raw data given by sensors that generate data associated with various security features planned in our monitoring system. The security incidents that triggered the monitoring sensors are associated to the relevant security features of relevant security objectives. In this section, we describe a simple way to build mass functions from raw data given by sensors and also a way to add temporization to take into account timed evidence which aligns with our efforts of continual security management. There are a variety of other methods that can be used in building mass functions, like in ([1] [7]), but ours is very simple and intuitive that fits well with our generic evidential model. The masses for our generic security parameters are generated directly from the sensors embedded in the security monitoring system. The values q(i,j), i=1,N and j=1,M associated with the security feature$F_j$can be entered in the membership functions of Figure 3 in order to produce the belief structure. If we let Ω={v, u}, where v means 'favorable to corporate security,' and u means 'unfavorable to corporate security,' and where m_Ω is defined as m_Ω: $2^\Omega \rightarrow [0, 1]$.Then the security monitoring system will generate the sequence <q(i,j), i-1,N; j=1,M>> that are entered in the graphs of Figure 3 to produce the belief structure of the security features Fj, j=1,M.

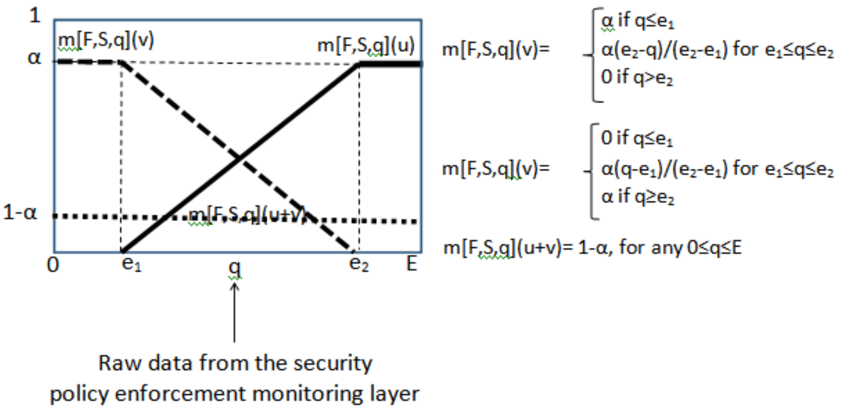


Figure 3: Timed construction of basic belief assignments from raw data

V. Model computations

Our evidential model achieves two objectives: the validation of security policy, and the enforcement of the security policy. We apply a continuous monitoring process for security policy enforcement and a continuous assessment of security risks. If the security risk is lower than the security policy tolerated risk as defined by owners, then both security policy validation and enforcement are achieved. If the security risks are assessed as higher than policy risk then we declare a security problem that could be either due to a security policy problem or security policy enforcement problem.We are assuming a set of auditable security policy rules that can be monitored using reliable sensors. A feasible combination of a diversified set of intrusion detection subsystems can do the job. The sensors will generate raw data needed to produce belief structures on the security features, as follows:

$\Omega=\{v, u\}$; $m[F_j]: 2^\Omega \rightarrow [1, 2]$

For any $1 \leq j \leq M$, x in 2^Ω, $m[F_j](x) = [[+]_{i=1,N}\ ](x)$ and $m[F_j](\Omega) = 1 - m[F_j](v) - m[F_j](u)$

Where:

v:'Favorable to corporate security'
u:'Favorable to corporate security'
F_j: Security objective j, $1 \leq j \leq M$
S_i: Security monitoring sensor i
N: Number of security policy monitoring sensors
M: Number of security objectives

One can complete the computation using the above equations, and obtain then a more detailed equation of the belief structure of the security objective:

For any k, $1 \leq j \leq K$,

$m[O_k] = [+]_{j=1,M}\ \delta(j,k)m[F_j]$

$m[F_j] = [[+]_{i=1,N}\ m[S_i, q(i,j)]]$

$$\begin{aligned} m[O_k] &= [+]_{j=1,M}\ \delta(j,k)m[F_j] \\ &= [+]_{j=1,M}\ \delta(j,k)\ [+]_{i=1,N}\ m[S_i, q(i,j)] \\ &= [+]_{j=1,M}\ [+]_{i=1,N}\ \delta(j,k)m[S_i, q(i,j)] \\ &= [+]_{i=1,N}\ [+]_{j=1,M}\ \delta(j,k)m[S_i, q(i,j)] \end{aligned}$$

Then, for any k, $1 \leq j \leq K$,

$\Omega=\{v, u\}$; $m[O_k]: 2^\Omega \rightarrow [1, 2]$

For any, x in 2^Ω,

$m[O_k](x) = [[+]_{i=1,N}\ [+]_{j=1,M}\ \delta(j,k)m[S_i, q(i,j)]](x)$

$m[O_k](\Omega) = 1 - m[O_k](v) - m[O_k](u)$

Where:

v:'Favorable to corporate security'
u:'Favorable to corporate security'
O_k: Security objective k
F_j: Security objective j, $1 \leq j \leq M$
S_i: Security monitoring sensor i
N: Number of security policy monitoring sensors
M: Number of security objectives

Once we obtained the belief structures of the security objectives, we can still apply Dempster's Rule to fuse all evidence obtained through security monitoring on security objectives and produce the corporate security posture and its security risk. AT this point, we have all what we need to compute the security posture and current security risks, as follows:

$$\begin{aligned} m[P] &= [+]_{i=1,K}\ m[O_k] \\ &= [+]_{i=1,K}\ [+]_{j=1,M}\ [+]_{i=1,N}\ \delta(j,k)\ m[S_i, q(i,j)] \\ &= [+]_{i=1,N}\ [+]_{j=1,M}\ [+]_{i=1,K}\ \delta(j,k)\ m[S_i, q(i,j)] \end{aligned}$$

That is, we obtain the following belief structure of the security posture m[P]:

$\Omega=\{v, u\}$

$m[O_k]: 2^\Omega \rightarrow [1, 2]$

For any, x in 2^Ω,

$m[P](x) = [[+]_{i=1,N}\ [+]_{j=1,M}\ [+]_{i=1,K}\ \delta(j,k)\ m[S_i, q(i,j)]](x)$

$m[P](\Omega) = 1 - m[P](v) - m[P](u)$

Where:

v:'Favorable to corporate security'
u:'Favorable to corporate security'
P: Security posture
O_k: Security objective k

F_j: Security objective j, 1≤j≤M
S_i: Security monitoring sensor i
N: Number of security policy monitoring sensors
M: Number of security objectives

VI. Risk driven continual security

It is important to adopt a continual security approach as recommended by one of the accepted standards: ISO 27002, ISO 27001, NIST SP 800-18, NIST SP 800-39, and NIST SP 800-53 ([9], [10], [12], [13], [14]). This is achieved through continual enforcement and monitoring of the security policy and continual review of the policy and its enforcement and monitoring processes when security risks are estimated below risk policy tolerated risks.

We are now faced with two situations. As shown in Figure 4, the first situation is when current risks are higher than policy risks. The second situation is when current risks are lower than policy risks. In the first situation, we need to apply a risk-driven cost-effective security program capable of bringing security risks below policy risks as defined by owners. In the second favorable situation, we just continue monitoring the security features and objectives to make sure that security risks remain below policy risks.

At this point, we have compiled all information needed to devise a security policy enforcement program. Any security policy may be translated into auditable rules that can be automatically or semi-automatically monitored. Information security management defines a set of security monitoring sensors that are capable of detecting all types of security policy violations that affect the security features. We recommend following a simple template that consists of three simple phases as defined in Figures 5 and 6. This template provides a simple framework that information security management can adopt for enforcing their security policy.

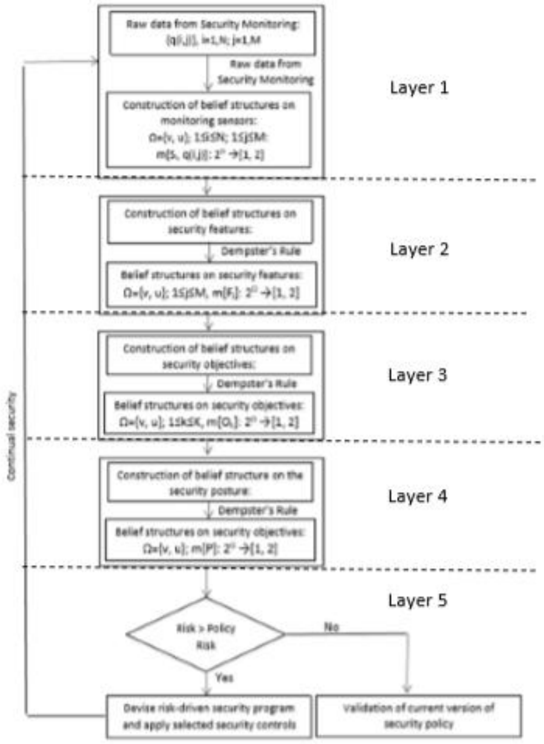
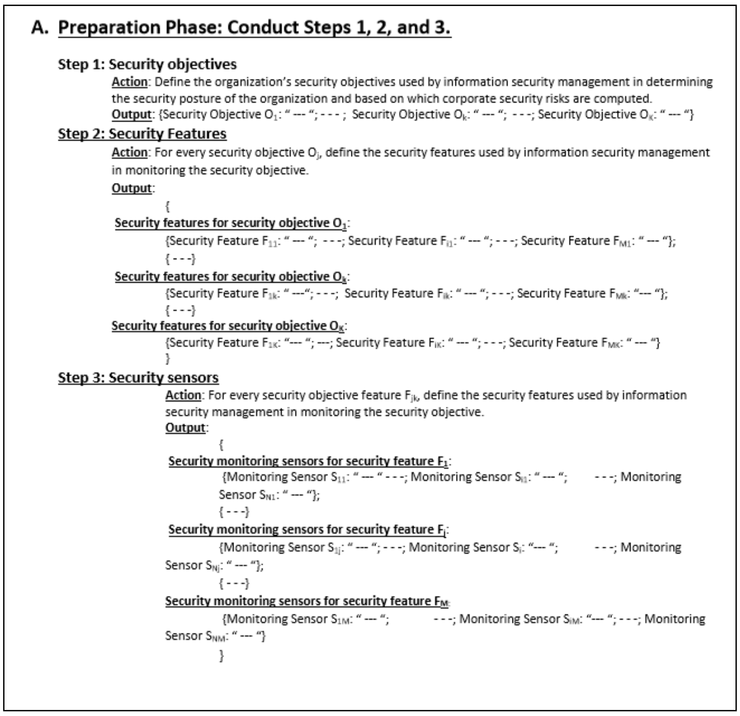

Raw data from Security Monitoring: {q(i,j)}, i=1,N; j=1,M
Raw data from Security Monitoring
Construction of belief structures on monitoring sensors: Ω={v, u}; 1≤i≤N; 1≤j≤M: m[S, q(i,j)]: 2^Ω →[1, 2]
Layer 1
Construction of belief structures on security features:
Dempster's Rule
Belief structures on security features: Ω={v, u}; 1≤j≤M, m[F_j]: 2^Ω →[1, 2]
Layer 2
Construction of belief structures on security objectives:
Dempster's Rule
Belief structures on security objectives: Ω={v, u}; 1≤k≤K, m[O_k]: 2^Ω →[1, 2]
Layer 3
Construction of belief structure on the security posture:
Dempster's Rule
Belief structures on security objectives: Ω={v, u}; m[P]: 2^Ω →[1, 2]
Layer 4
Layer 5
Risk > Policy Risk
No
Yes
Devise risk-driven security program and apply selected security controls
Validation of current version of security policy
Continual security

Figure 4: Risk-driven security program



A. Preparation Phase: Conduct Steps 1, 2, and 3.

Step 1: Security objectives

Action: Define the organization's security objectives used by information security management in determining the security posture of the organization and based on which corporate security risks are computed.

Output: {Security Objective O_1: " --- "; - - - ; Security Objective O_k: " --- "; - - -; Security Objective O_K: " --- "}

Step 2: Security Features

Action: For every security objective O_j, define the security features used by information security management in monitoring the security objective.

Output:

{

Security features for security objective O_1:

{Security Feature F_{11}: " --- "; - - -; Security Feature F_{i1}: " --- "; - - -; Security Feature F_{M1}: " --- "};

{ - - -}

Security features for security objective O_k:

{Security Feature F_{1k}: " ---"; - - -; Security Feature F_{ik}: " --- "; - - -; Security Feature F_{Mk}: "--- "};

{ - - -}

Security features for security objective O_K:

{Security Feature F_{1K}: "--- "; ---; Security Feature F_{iK}: " --- "; - - -; Security Feature F_{MK}: " --- "}

}

Step 3: Security sensors

Action: For every security objective feature F_{jk}, define the security features used by information security management in monitoring the security objective.

Output:

{

Security monitoring sensors for security feature F_1:

{Monitoring Sensor S_{11}: " --- " - - -; Monitoring Sensor S_{i1}: " --- "; - - -; Monitoring Sensor S_{N1}: " --- "};

{ - - -}

Security monitoring sensors for security feature F_j:

{Monitoring Sensor S_{1j}: " --- "; - - -; Monitoring Sensor S_{ij}: "--- "; - - -; Monitoring Sensor S_{Nj}: " --- "};

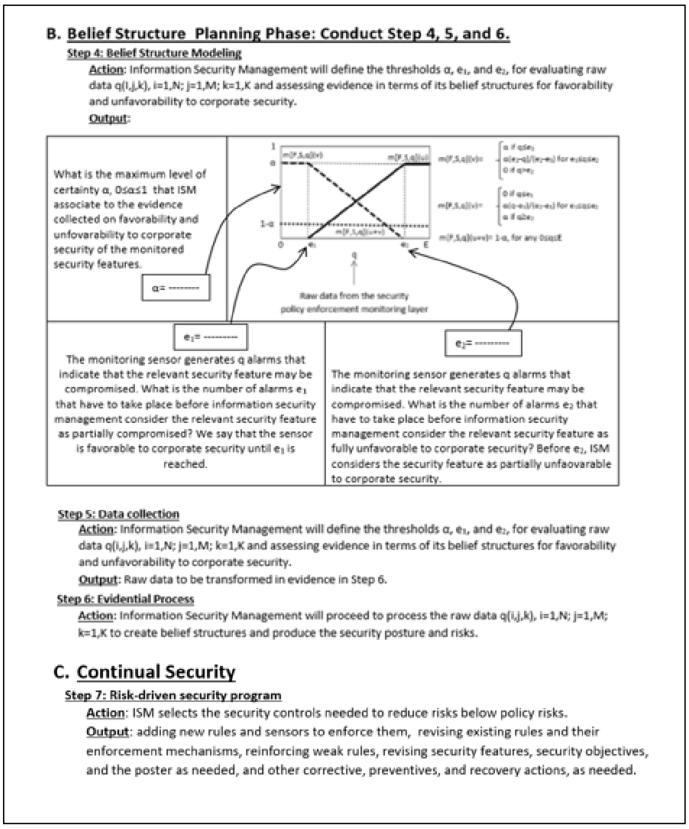
{ - - -}

Security monitoring sensors for security feature F_M:

{Monitoring Sensor S_{1M}: " --- "; - - -; Monitoring Sensor S_{iM}: "--- "; - - -; Monitoring Sensor S_{NM}: " --- "}

}

Figure 5: Policy enforcement template: Phase A



B. Belief Structure Planning Phase: Conduct Step 4, 5, and 6.

Step 4: Belief Structure Modeling

Action: Information Security Management will define the thresholds α, e_1, and e_2, for evaluating raw data q(I,j,k), i=1,N; j=1,M; k=1,K and assessing evidence in terms of its belief structures for favorability and unfavorability to corporate security.

Output:

What is the maximum level of certainty α, 0≤α≤1 that ISM associate to the evidence collected on favorability and unfovarability to corporate security of the monitored security features.

α= --------

1
α
1-α
0
e_1
e_2
E
q
m(F,S,q)(v)
m(F,S,q)(u)
m(F,S,q)(u+v)
m(F,S,q)(v)= α if q≤e_1; α(e_2-q)/(e_2-e_1) for e_1≤q≤e_2; 0 if q≥e_2
m(F,S,q)(u)= 0 if q≤e_1; α(q-e_1)/(e_2-e_1) for e_1≤q≤e_2; α if q≥e_2
m(F,S,q)(u+v)= 1-α, for any 0≤q≤E
Raw data from the security
policy enforcement monitoring layer

e_1= --------

The monitoring sensor generates q alarms that indicate that the relevant security feature may be compromised. What is the number of alarms e_1 that have to take place before information security management consider the relevant security feature as partially compromised? We say that the sensor is favorable to corporate security until e_1 is reached.

e_2= --------

The monitoring sensor generates q alarms that indicate that the relevant security feature may be compromised. What is the number of alarms e_2 that have to take place before information security management consider the relevant security feature as fully unfavorable to corporate security? Before e_2, ISM considers the security feature as partially unfaovarable to corporate security.

Step 5: Data collection

Action: Information Security Management will define the thresholds α, e_1, and e_2, for evaluating raw data q(i,j,k), i=1,N; j=1,M; k=1,K and assessing evidence in terms of its belief structures for favorability and unfavorability to corporate security.

Output: Raw data to be transformed in evidence in Step 6.

Step 6: Evidential Process

Action: Information Security Management will proceed to process the raw data q(i,j,k), i=1,N; j=1,M; k=1,K to create belief structures and produce the security posture and risks.

C. Continual Security

Step 7: Risk-driven security program

Action: ISM selects the security controls needed to reduce risks below policy risks.

Output: adding new rules and sensors to enforce them, revising existing rules and their enforcement mechanisms, reinforcing weak rules, revising security features, security objectives, and the poster as needed, and other corrective, preventives, and recovery actions, as needed.

Figure 6: Policy enforcement template: Phases B and C

VII. Numerical example

A security policy is often enforced as a set of auditable rules that can be evaluated in a quick manner. The literature provides a variety of methods for the generation of security policy rules []. In our example, we assume that we are monitoring our security policy enforcement effort using 5 security features {F1, …, F5}, and 3 security objectives {O1, O2, O3}, as follows:

Security Objectives:

O_1: Assure confidentiality
O_2: Assure integrity
O_3: Assure availability

Security Features:

F_1: Physical security
F_2: Network & system security
F_3: Application security
F_4: Privacy
F_5: Access control

Also assume that we only use security policy enforcement data base where we only detect 7 types of policy violations monitored using 7 sensors $\{S_1, …, S_7\}$ directly related to the security features above. Our security policy enforcement model with its evidence exchange is depicted in Figure 7.

Layer 1 of the security policy monitoring sensors $\{S_1, …, S_7\}$ generate raw data q(i, j), i=1,7 and j=1,5 that is directed to relevant security features. A sensor S_i generates q(i, j) alarms to security feature F_j, j=1,5. This indicates the number of times a security policy violation took place that affected the security feature F_j. The number q(i, j) is then plugged into the basic belief assignment generator, depicted in Figure 8, to produce evidence on the state of the security feature F_j. The relevance of security policy violation alerts to security objectives is provided in Table 2. This table gives for every sensor S_i the number of alarms q(i,j) relevant to the security feature F_j. Evidence on the state of a security feature may be obtained from multiple sensors and then combined using Dempster rule of combination to produce a cumulative evidence on the state of the security feature.

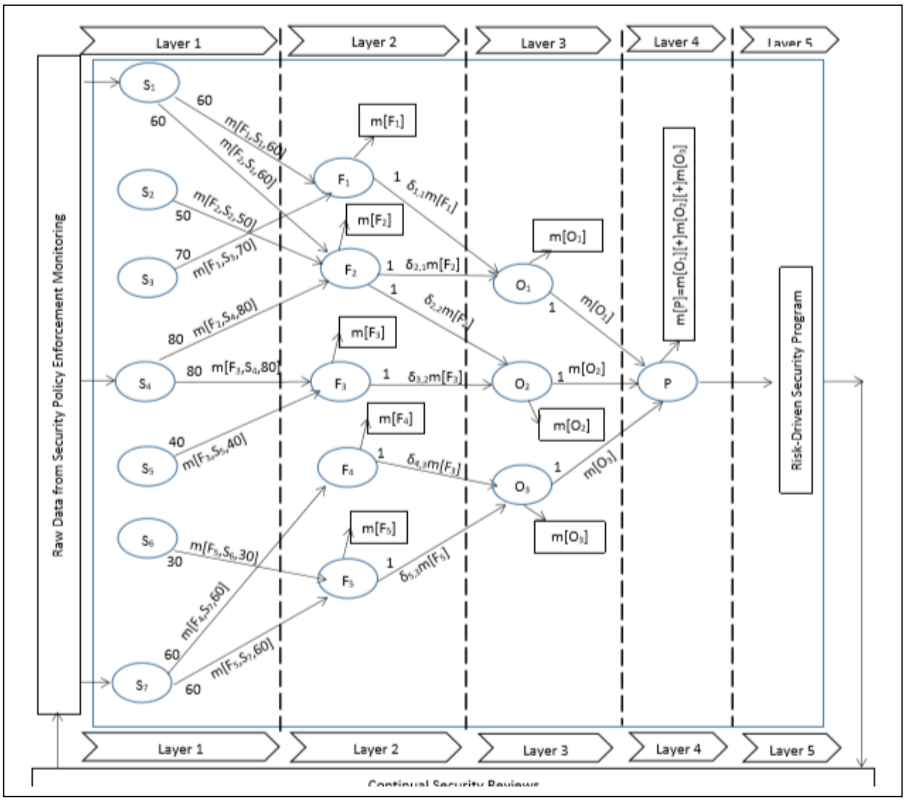


Figure 7: Exchange of evidence among layers for the numerical example

Let us start by evaluating the raw data obtained from the security policy enforcement monitoring sensors {S1, …, S7}. These numbers q(i,j), i=1,7; j=1,5 are provided in Table 1. These numbers are plugged into the membership function, in Figure 8, prescribed to produce the partial belief structures representing partial evidence on the states of security features.

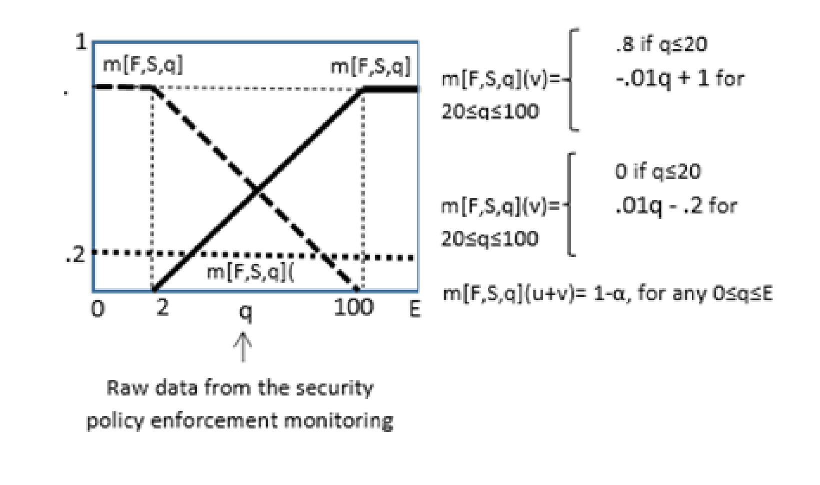

1
.
.2
m[F,S,q]
m[F,S,q]
m[F,S,q](
0
2
q
100
E
Raw data from the security policy enforcement monitoring
m[F,S,q](v)= .8 if q≤20; -.01q + 1 for 20≤q≤100
m[F,S,q](v)= 0 if q≤20; .01q - .2 for 20≤q≤100
m[F,S,q](u+v)= 1-α, for any 0≤q≤E

Figure 8: Belief structure construction for the numerical example

These initial belief structures are provided in Table 3. At this point, we have accumulated all initial evidence for each security feature, and we can now use Dempster rule to produce the cumulative evidence on the state of each security feature as shown in Table 4. Here are the belief structures obtained after applying Dempster rule to available evidence on the states of security features:

$$m[F_1]: \begin{cases} \Omega=\{v,u\} \\ 2^{\Omega} \rightarrow [0, 1] \\ m[F_1](v)= 0.382353 \\ m[F_1](u)= 0.558824 \\ m[F_1](u+v)= 0.058824 \end{cases} \qquad m[F_2]: \begin{cases} \Omega=\{v,u\} \\ 2^{\Omega} \rightarrow [0, 1] \\ m[F_2](v)= 0.4 \\ m[F_2](u)= 0.58 \\ m[F_2](u+v)= 0.02 \end{cases}$$

$$m[F_3]: \begin{cases} \Omega=\{v,u\} \\ 2^{\Omega} \rightarrow [0, 1] \\ m[F_3](v)= 0.466667 \\ m[F_3](u)= 0.466667 \\ m[F_3](u+v)= 0.066667 \end{cases} \qquad m[F_4]: \begin{cases} \Omega=\{v,u\} \\ 2^{\Omega} \rightarrow [0, 1] \\ m[F_4](v)= 0.2 \\ m[F_4](u)= 0.6 \\ m[F_4](u+v)= 0.2 \end{cases}$$

$$m[F_5]: \begin{cases} \Omega=\{v,u\} \\ 2^{\Omega} \rightarrow [0, 1] \\ m[F_5](v)= 0.823529 \\ m[F_5](u)= 0.117647 \\ m[F_5](u+v)= 0.058824 \end{cases}$$

As shown in Tables 1-5, we got a belief structure for the organization's security posture m[P] on Ω with a bba m[P]: 2Ω → [0, 1], such that m[P](v)= 0.710145; m[P](u)= 0.289855; m[P](u+v)= 5.11E-07. The organization's security risk is m[P](u)+m[P](u+v)= 0.289855.

Given that this risk is higher than the maximum security risk allowed by the security policy, then there is an urgent need to take the appropriate actions to review the security policy and its enforcement process, including the model used to monitor security and produce the organization's security risk. Examples of the possible actions the security administrator can take include adding new rules and sensors to enforce them, revising existing rules and their enforcement mechanisms, reinforcing weak rules, revising security features, security objectives, and the poster as needed, and other corrective, preventives, and recovery actions, as needed.

Table 1: Security sensor relevance to security features: Number of alarms generated by S_i that are relevant to feature F_j

		Security Features				
		F_1	F_2	F_3	F_4	F_5
Security Sensors	S_1	60	60	0	0	0
	S_2	0	50	0	0	0
	S_3	70	0	0	0	0
	S_4	0	80	80	0	0
	S_5	0	0	40	0	0
	S_6	0	0	0	0	30
	S_7	0	0	0	60	60

Table 2: Relevance of Security Objectives to Security Features

		Security Objectives		
		O_1	O_2	O_3
Security Features	F_1	1	0	0
	F_2	1	1	0
	F_3	0	1	0
	F_4	0	0	1
	F_5	0	0	1

Table 3: Computing Features

Features	Fused	m[F_j], j=1,5
F_1	{m[F_1,S1,5], m[F_1,S3,4]}	m[F_1,S1,5] [+] m[F_1,S3,4]
F_2	{m[F_2,S1,2], m[F_2,S2,6], m[F_2,S4,2]}	m[F_2,S1,2] [+] m[F_2,S2,6] [+] m[F_2,S4,2]
F_3	{m[F_3,S4,4], m[F_3,S5,4]}	m[F_3,S4,4] [+] m[F_3,S5,4]
F_4	{m[F_4,S7,2]}	m[F_4,S7,2]
F_5	{m[F_5,S7,5], m[F_5,S6,6]}	m[F_5,S7,5] [+] m[F_5,S6,6]

Table 4: Computation of bba's for security features

	1st operand in [+]			2nd operand in [+]			Bba's on security features		
	m(v)	m(u)	m(u+v)	m(v)	m(u)	m(u+v)	m(v)	m(u)	m(u+v)
F_1=m[F_1,S_1,60] [+] m[F_1,S_3,70]	0.4	0.4	0.2	0.3	0.5	0.2	0.382353	0.558824	0.058824
m[F_2,S_1,60] [+] m[F_2,S_2,50]	0.4	0.4	0.2	0.5	0.3	0.2	0.558824	0.382353	0.058824
F_2=m[F_2,S_1,60] [+] m[F_2,S_2,50] [+] m[F_2,S_4,80]	0.59	0.38	0.06	0.2	0.6	0.2	0.4	0.58	0.02
F_3=m[F_3,S_4,80] [+] m[F_3,S_5,40]	0.2	0.6	0.2	0.6	0.2	0.2	0.466667	0.466667	0.066667
F_4=m[F_4,S_4,80]	0.2	0.6	0.2	-	-	-	0.2	0.6	0.2
F_5=m[F_5,S_6,30] [+] m[F_5,S_7,60]	0.8	0	0.2	0.4	0.4	0.2	0.823529	0.117647	0.058824

Table 5: Computation of bba's for security objective

	1st operand in [+]			2nd operand in [+]			Bba's on security features		
	m(v)	m(u)	m(u+v)	m(v)	m(u)	m(u+v)	m(v)	m(u)	m(u+v)
m[O_1]=m[F_1] [+] m[F_2]	0.382353	0.558824	0.0588235	0.4	0.58	0.02	0.331919	0.66596	0.002121
m[O_2]= m[F_2] [+] m[F_3]	0.4	0.58	0.02	0.466667	0.466667	0.066667	0.410319	0.587224	0.002457
m[O_3]= m[F_4] [+] m[F_5]	0.5	0.3	0.2	0.823529	0.117647	0.058824	0.872881	0.110169	0.016949
m[O_1] [+] m[O_2]	0.331919	0.66596	0.0021209	0.410319	0.587224	0.002457	0.259252	0.740738	9.8E-06
m[O_1] [+] m[O_2] [+] m[O_3]	0.259252	0.740738	9.798E-06	0.872881	0.110169	0.016949	0.710145	0.289855	5.11E-07

VIII. Conclusion

We proposed an evidential model for a risk-driven security policy enforcement process using Dempster and Safer Theory. Information security management were assumed to have a corporate security policy made of auditable rules for which monitoring sensors are applied to detect violations of the security policy. We adopted security objectives and features on which the security policy is structured. Policy enforcement sensors were configured to generate the raw data needed to construct the belief structures needed to manage model uncertainty.

Belief structures were shaped using a membership function that accounts for two generic types of evidence, favorable to corporate security or unfavorable to it. The raw data were needed to be translated into belief structures on the security sensors, features, and objectives, that are later sequentially fused using Dempster rule to produce the corporate security posture. Information security management can then evaluate the company security risk and devise a cost-effective risk-driven security program capable to bring the current security risk below policy security risks.

Our model was intentionally designed in a generic manner for two reasons: 1) to allow for enough flexibility in defining the organization security features and objectives, the monitoring of its security policy rules, the computing of its security posture and risks, and the devising of a cost-effective risk-driven security program, and 2) to provide a standard template that information security management can adopt in the design of a security policy enforcement system capable of continual security. Our model can also serve as a mechanism for the validation of a security policy and as a risk-driven continual security process. We provided a numerical example to demonstrate the working of our evidential model and its claims.

Future directions to expand this work may be to take advantage of historical data on the security states of the computing environment and translate the behaviors of sensors into Poisson arrivals and Bernoulli trails, respectively for the occurrence of security incidents and for triggering policy risks. This probabilistic information may be very useful in predicting the occurrence of those incidents that elevate security risks above policy risks which will trigger information security management controls. Combining available probabilistic information and DST evidence in our model may produce a better security policy enforcement process.

References

[1]. Aguirre et al., (2013), Construction of Belief Functions From Statistical Data About Reliability Under Epistemic Uncertainty, IEEE Transactions on Reliability, 62(3).

[2]. Aickelin et al., (2007), Rule generalization in intrusion detection systems using SNORT, Int. J. Electronic Security and Digital Forensics, 1(1).

[3]. AlKalbani et al., (2017), Information Security Compliance in Organizations: An Institutional Perspective, Data and Information Management, 1(2), 104-114.

[4]. Cram et al., (2017), Organizational information security policies: a review and research framework, European Journal of Information Systems, 26(6), 605-641.

[5]. Dempster, A. P. (1969). Upper and lower probability inferences for families of hypotheses with monotone density ratios. Ann. Math. Statist. 40(3), 953–969.

[6]. Dempster, A. P. (2008). Dempster–Shafer calculus for statisticians. International Journal of Approximate Reasoning, 48, 265–277.

[7]. Denoeux, T. (2006). Constructing belief functions from sample data using multinomial confidence regions. International Journal of Approximate Reasoning, 42, 228–252.

[8]. Dutta S.K., (1998), Applications of Fuzzy Sets & The Theory of Evidence to Accounting II, Vol. 7, edited by P. Siegel, K. Omer, A. Korvin, and A. Zebda, published by Jai Press Inc., 1998, pp. 221-244

[9]. ISO, ISO 27002, https:/www.iso.org/

[10]. ISO, ISO 27001, https:/www.iso.org/

[11]. Moody et al., (2018), Toward a Unified Model of Information Security Policy Compliance, MIS Quarterly, 42(1), 285-311.

[12]. NIST, NIST SP 800-53, https://www.nist.gov/

[13]. NIST, NIST SP 800-39, https://www.nist.gov/

[14]. NIST, NIST SP 800-18, https://www.nist.gov/

[15]. Raggad, B., (2010), Information Security Management: Concepts and Practice, CRC Press, NY.

[16]. Shafer, G. (1976). A Mathematical Theory of Evidence, Princeton Univ. Press, Princeton, NJ.

[17]. Smets, P. and R. Kennes, (1994), The transferable belief model, Artificial Intelligence, 66. 191-234.

[18]. Smets P., (1990), The combination of evidence in the transferable belief model. IEEE-Pauern analysis andMachine Intelligence, 12, 447-458.

[19]. Tedeschi et al., (2019), A Design Approach to IoT Endpoint Security for Production Machinery Monitoring, Sensors 2019, 19(10).

[20]. Wasserman, L.A., (1990), Belief Functions and Statistical Inference, The Canadian Journal of Statistics, 18(3), 183-196.

[21]. Whitman, M.E. and H.J. Mattord, (2018), Management of Information Security,,Cengage Learning; 6th edition.

Bel G. Raggad. "Risk-driven Security Policy Enforcement: An Evidential Model." *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)* 15(6), (2020): 01-12.