Belief-based Decision Making for Service Migration

Yanjun Zuo
University of North Dakota, Grand Forks, ND USA
yanjun.zuo@und.edu

Abstract

Service migration is an important strategy to improve a system’s ability to survive malicious attacks and to continually provide mission-critical services. By moving the critical services from their compromised platforms to other clean, healthy platforms in order for those services to continuously function on those new platforms, further loss can be avoided in case of a devastating attack. Given the increasing complexity of malicious attacks and limited resources to fully assess any damage caused by attacks within a short period of time, damage assessment as provided by each intrusion detection agent is often incomplete or uncertain. By integrating multiple sources of damage assessments from different intrusion detection agents, a more reliable and trustworthy damage assessment can be formed about a platform of concern. Such a combined damage assessment is important in determining whether a service migration is necessary. We present a transferable belief-based decision model to represent and combine individual damage assessment outputs from multiple intrusion detection agents and then construct a comprehensive, more reliable output. A final decision can be made to choose the most effective and cost efficient security action to take in an intrusion scenario.

1. Introduction

Service migration is an important survivability strategy in which critical services are relocated from their current compromised platforms to other clean, healthy platforms to avoid further loss and still make the services available to users. Service migration is crucial in case of a devastating attack, where some platforms have been severely compromised and limited resources are available to fully assess and recover those damaged platforms in a timely manner. To make sure that critical services can be continuously provided, the services executed on those compromised platforms are temporarily relocated to other platforms and resume where computation was left off on those new platforms.

As compared with other security and survivability techniques, service migration minimizes the requirement for potentially expensive system or component redundancy as often used in fault-tolerance and self-healing [1-3]. With an increasing complexity of malicious attacks and growing variety of fault types, it is often difficult or too costly to perform redundancy or system damage recovery [4]. Service migration is therefore an effective approach to move the critical services from their current compromised platforms and continue their functions on other healthy ones. Service migration also reduces the need for an organization to build “super-strong” systems for critical services [5]. As the cost of developing and maintaining high-security and high-integrity systems increases exponentially with users’ security requirements, developing such a highly required system is often cost prohibitive. In a service migration schema, services are strategically relocated to other platforms dynamically as needed, thus reducing the overall cost to maintain a highly trustworthy system. Service/task migration has been studied in various fields including system/service survivability [4, 6], secure cloud computing [7-8], fault tolerance and failure recovery [9-11], and adaptation and service balancing [12-13] (to cite a few).

The first and important step of a service migration process is to determine whether a service migration is the best course of action to take in a security incident. A decision for service migration must be made before any resources are allocated and any critical services are arranged to be relocated from their current platforms to others. Such a decision needs to make a balance between the cost of service migration itself (e.g., suspending current running processes, transferring the data and services to new platforms, and setting up the services on the new platforms) and the necessity of migrating services to avoid further loss (e.g., any direct and indirect cost resulted from the compromised platforms if a service migration is not performed plus any cost resulted from loss of services).

Fundamentally, a decision for service migration boils down to the question whether the platform under investigation has been severely damaged. Service migration is best used when the platform has been severely compromised to such a degree that any
services executed on the platform are in a great threat and thus must be migrated in order to continue their operations safely on other healthy platforms. We assume that the service software runs in a self-contained virtualization environment (which is often the case in various distributed platforms such as cloud and grid computing) and hence is relatively damage free even when the underlying platform may have been damaged. Service migration also allows the necessary time for the compromised platforms to be repaired while all the currently executing services have been relocated somewhere else.

Determining whether a platform has been damaged and to which degree it has been damaged is not a trivial issue. The effect of an attack may not be observable immediately and some damage may not be detected until certain conditions are met. In a security crisis scenario where the environment is challenging and a prompt action must be taken to respond to the attack, an accurate assessment of any damage on a platform is often difficult. Practically, an intrusion detection agent could only give the best estimation of a damage profile of the platform, e.g., the possibility that the platform is severely compromised, it is damaged with a moderate degree, it is only affected with a minor degree, or it is not damaged at all. Due to any uncertain and ill-known information about the system state and the attack, the damage assessments provided by multiple intrusion detection agents about the same platform are often redundant or even conflicting. A more precise and reliable damage assessment is highly required. This is also because any security action to be taken depends on the damage assessment on the platform. In terms of service migration, it is best used in case that the platform has been severely damaged and thus moving the services to other platforms is the most effective way to make those critical services continuously available to users.

In this paper, we present a transferable belief [14-15] -based decision model to represent the damage assessment output from an intrusion detection agent and combine multiple sources of the assessment outputs from multiple agents. The combined result is a more accurate and reliable damage assessment about the platform of concern. In our design, each individual damage assessment provided by an intrusion detection agent is represented as a basic belief assignment (BBA), i.e., a belief mass function on the subsets of a belief domain (i.e., frame of discernment). A belief combination rule is applied to integrate multiple sources of beliefs to get a comprehensive, more reliable belief assignment which represents the final damage assessment on the platform. The combined belief assignment represents a probability distribution on all possible combinations of the damage states of the platform. Given the cost of performing different security actions on each damage state of the platform, our decision model determines whether a service migration is the most effective and cost efficient action to take in the security incident. In case that service migration is necessary, such a decision justifies the needs to allocate resources to move critical services from their current platforms to other platforms so that the services can be continuously provided on those new platforms.

Transferable belief model, an elaboration on the Dempster-Shafer theory of evidence, has been used for information fusion [16-17], state sequence analysis [18], and knowledge representation and combination [19]. Particular applications of the transferable belief model can be found in reliability analysis of systems [20], post address recognition [21], climate sensitivity analysis [22], and navigation system [23], just to cite a few. In our research, we use the transferable belief model to facilitate users in making a service migration-related security decision. To our best knowledge, our work is the first of its kind in applying a belief theory to service migration decision making.

In the rest of the paper, we first review the basics of the transferable belief model and the Bayesian network in Section 2. In Section 3 we present our service migration decision model with three components: constructing a BBA function to represent a damage assessment on a platform; combining multiple BBAs to form a more reliable and accurate belief assignment; and determining the most effective and cost efficient security action to take in a security incident. Finally, we conclude our paper in Section 4.

2. Preliminary

In this section, we briefly introduce the two theoretical foundations used in our research: the transferable belief model and Bayesian network.

2.1. Transferable belief model

2.1.1. The Basics. Belief theory is useful in representing and reasoning uncertain knowledge. In this section, we briefly introduce the basics of the transferable belief model (for details, see [14-15]).

Let \( \Omega \) represent a finite set of elementary events related to a given problem. \( \Omega \) is commonly called frame of discernment. User beliefs, i.e., a piece of evidence, on the different subsets of \( \Omega \) are represented by the basic belief assignment (BBA) [24], denoted as a function \( m : 2^\Omega \rightarrow [0, 1] \). A BBA \( m(.) \) satisfies the property: \( \sum_{A \subseteq \Omega} m(A) = 1 \). A basic belief mass \( m(A) \geq 0 \)
represents the part of belief exactly committed to the event \( A \subseteq \Omega \) given a piece of evidence. This quantity of belief can’t be apportioned to any strict subset of \( A \). An element \( A \subseteq \Omega \) with \( m(A) > 0 \) is called a focal element of \( \Omega \).

A BBA \( m(\cdot) \) is called a normalized basic belief assignment if \( m(\emptyset) = 0 \), where \( \emptyset \) represents the empty set. However, the notion of \( m(\emptyset) \) is relaxed in [14, 25] as the amount of conflict between the pieces of evidence or as the part of belief given to the fact that none of the hypotheses in \( \Omega \) is true.

As shown in [15], a basic belief assignment \( m(\cdot) \) can be transformed to a probability measure on each element of the frame of discernment, denoted as \( \text{Bet}\mathcal{P}(\cdot) \). The link between those two types of measures is achieved by the pignistic transformation:

\[
\forall A \in \Omega, \text{Bet}\mathcal{P}_m(A) = \sum_{B \subseteq A} \left( \frac{|A \cap B|}{|B|} \cdot m(B) \right) \frac{1}{1-m(\emptyset)} \quad (1)
\]

As we can see, a BBA \( m(\cdot) \) is defined on the domain of the subsets of \( \Omega \), i.e., \( 2^\Omega \) and \( \text{Bet}\mathcal{P}_m(\cdot) \) is defined on the domain \( \Omega \). As we will see in Section 3.3, this pignistic transformation is very useful in formulating decision formulas.

2.1.2. BBA combination. When information sources are considered reliable, multiple BBAs can be conjunctively combined. Consider a simple case with two BBAs \( m_1(\cdot) \) and \( m_2(\cdot) \). The combined basic belief assignment, denoted as \( m_{12}(\cdot) \), can be calculated using the conjunctive rule of combination [15]:

\[
m_{12}(A) = \sum_{B \subseteq C \subseteq A} m_1(B) \cdot m_2(C) \quad \forall A, B, C \subseteq \Omega \quad (2)
\]

There are also other conjunctive combination rules. For example, the Dempster’s rule [26], denoted as \( mc(\cdot) \), is defined as:

\[
mc(A) = \frac{1}{1-m(\emptyset)} m_c(A) \quad \forall A \subseteq \Omega \text{ and } A \neq \emptyset \quad (3)
\]

Since both of the above conjunctive combination rules are not idempotent, they can only be used when the sources of evidences are independent.

2.1.3. BBA discount and reinforcement. For a non-reliable source of evidence, its basic belief assignment is “discounted” before it can be combined with other belief assignments. On the other hand, for those sources which are considered very reliable, their basic belief assignments may be reinforced before the combination with other belief assignments. A BBA correction mechanism is defined in [27] for weakening or reinforcing a BBA \( m(\cdot) \): \n
\[
m^\prime(A) = w_1 m_0(A) + w_2 m(A) + w_3 m^\prime(A) \quad \forall A \subseteq \Omega \quad (4)
\]

where \( m^\prime(\cdot) \) represents the discounted or reinforced BBA; \( m_0(\cdot) \) represents the vacuous BBA, i.e., \( m_0(\Omega) = 1 \) and for all other \( \forall A \subseteq \Omega, m(A) = 0 \); \( m^\prime(\cdot) \) represents the totally enforced BBA corresponding to \( m(\cdot) \), which is defined as:

\[
(1) \quad m^\prime(A) = \frac{m(A)}{1-m(\emptyset)} \quad \forall A \subseteq \Omega, \text{ and } \quad (2) \quad m^\prime(\Omega) = 0 \quad \text{ and } w_1, w_2, w_3 \in [0, 1] \text{ represent the weights assigned to } m_0(\cdot), m(\cdot), \text{ and } m^\prime(\cdot) \text{ respectively, and } w_1 + w_2 + w_3 = 1.
\]

As we can see from Formula (4), \( w_1 = 1 \) represents a total discount of \( m(\cdot) \), i.e., \( m^\prime(\cdot) = m_0(\cdot) \), which essentially means that \( m^\prime(\cdot) \) represents a complete ignorance. On the other hand, \( w_3 \) represents a complete reinforcement of \( m(\cdot) \) by redistributing the belief mass of the empty set \( \emptyset \) to all other subsets of \( \Omega \). Therefore, by choosing the appropriate values for \( w_1, w_2 \) and \( w_3 \), a user can discount or reinforce an input BBA correspondingly.

2.2. Bayesian belief network

As will be seen, our proposed service migration decision model uses Bayesian belief network (or Bayesian network for short) to construct a basic belief assignment to represent a damage assessment provided by an intrusion detection agent about a platform under investigation. A Bayesian network is a powerful tool for modeling cause-effect relationships among a set of event variables. Basically, a Bayesian network is a joint probabilistic model for \( n \) random variables represented as a directed acyclic graph \( G = (V, E) \) where \( V \) is a set of nodes corresponding to the set of variables \( V = \{x_1, \ldots, x_n\} \) and \( E \subseteq V \times V \) contains directed edges connecting nodes. Instead of weights, the graph edges are described by conditional probabilities of nodes given their parents that are used to construct a joint distribution on the set of variables \( x_1, \ldots, x_n \).

A Conditional Probability Table (CPT) is associated with a node defining quantitatively how the node (called the child node) is influenced by its parent nodes. In a cause-effect relationship, each child node is conditionally dependent upon their parent nodes.

In a Bayesian network, the relationship between two nodes \( N_1 \) and \( N_2 \) is defined as a conditional probability, \( P(N_1 | N_2) \), which represents the probability of \( N_1 \) conditional on a given outcome of \( N_2 \). The conditional probability is calculated using Bayes’ Theorem [28]:

\[
P(N_1 | N_2) = \frac{P(N_1 \cap N_2) \cdot P(N_1)}{P(N_2)}
\]

where \( P(N_2 \mid N_1) \) is the conditional probability of \( N_2 \) given \( N_1 \), and \( P(N_2) \) and \( P(N_1) \) are the unconditional probabilities of \( N_2 \) and \( N_1 \), respectively. The probabilistic dependency is represented in the CPT, which is attached to the corresponding node. A CPT shows all possible outcomes of a given node and the
conditional probability corresponding to each outcome given all its parent nodes.

3. The service migration decision model

We assume that there are various intrusion detection agents deployed in a system. They operate based on different intrusion detection methodology and mechanisms, e.g., signature-based, anomaly detection-based and are deployed at different levels of the system, e.g., host-based, network-based. Those intrusion detection agents monitor and detect attacks from different perspectives and collectively provide intrusion awareness and damage assessments to system administrators in a security incident.

The flowchart of our model is shown in Figure 1. As we can see, an intrusion detection agent’s damage assessment on a particular platform is represented as a basic belief assignment (BBA). As next section shows, we have proposed a Bayesian network-based approach to construct a BBA for each agent’s damage assessment. The damage assessment BBAs provided by multiple intrusion detection agents about the same platform are combined into a comprehensive damage assessment about that platform. As will be discussed in Section 3.2, the conjunctive combination rule is used to combine those multiple BBAs. Since the damage assessments from different intrusion detection agents have different reliability and importance, discounting and reinforcement are conducted to adjust each of those BBAs before they can be combined. Finally, a decision process is conducted based on the cost of performing each security action and the potential loss of the system in case that the action does not sufficiently address the damage caused by the attack. Thus, the decision model determines whether service migration is the most effective and cost efficient approach to take in the given security incident.

Figure 1. Flowchart of service migration decision model

3.1. BBA construction – representation of an agent’s damage assessment output

To define a basic belief assignment to represent an intrusion detection agent’s damage assessment on a platform, we first specify the frame of discernment \( \Omega \) – the set of elementary events to represent the possible damage statuses of the platform. We have defined \( \Omega \) to include the following elements:

\[ \begin{align*}
\text{Devastating blow} & - \text{the platform has been completely compromised;} \\
\text{Moderate damage} & - \text{the platform has been damaged with a moderate level of degree;} \\
\text{Microlesion} & - \text{the platform has been slightly affected;} \\
\text{Free of damage} & - \text{there is no damage on the platform.}
\end{align*} \]

Therefore, we have \( \Omega = \{ \text{Devastating blow, Moderate Damage, Microlesion, Free of damage} \} \). Since the first three elements of \( \Omega \) all indicate that the platform has been damaged to a degree, we define Impairment to represent the subset \( \{ \text{Devastating blow, Moderate damage, Microlesion} \} \). Given \( \Omega \), an intrusion detection agent outputs one of the following three types of results: (1) an element of \( \Omega \); (2) Impairment; and (3) the entire frame of discernment \( \Omega \).

The output Impairment indicates that the intrusion detection agent believes that the platform has been damaged but it is not sure to which degree it has been damaged. The output \( \Omega \) represents the total ignorance, i.e., the agent is completely uncertain whether the platform has been damaged or not. As we can see, any other combination of the elements of \( \Omega \) is deemed invalid since the composing elements must be mutually exclusive. For example, it does not make sense that an intrusion detection agent declares that a platform has been severely compromised (i.e., Devastating blow) and in the meantime claims that it has been damaged with a minor degree (i.e., Microlesion) or not damaged at all (i.e., Free of damage). In other words, either the subset \( \{ \text{Devastating blow, Microlesion} \} \) or \( \{ \text{Devastating blow, Free of damage} \} \) is treated as an invalid output, i.e., its belief mass should be zero.

We next present a Bayesian network-based approach to represent an intrusion detection agent’s damage assessment output about a platform in form of a basic belief assignment (BBA). The idea is that an agent determines whether a platform has been damaged and to which degree it has been damaged based on a set of influencing factors. The cause-effect relationships between those influencing factors and the damage profile of the platform are represented in a Bayesian network. From a security perspective, any damaged system and its components more or less demonstrate certain symptoms and the attack effects can be observed through monitoring a set of variables. For example, if a platform is severely damaged, then it must behave significantly abnormally and certain symptoms must be detected such as privilege elevation of user accounts, installation of hidden programs on critical system execution paths, establishment of network connections to external botnet masters or to malicious attacker sponsored web sites, and modification of key system files. Those symptoms are more often used by a host-based intrusion detection agent to detect system level intrusion and damage.
Other intrusion detection agents may use different principles and mechanisms to detect intrusion and assess damage. A network-based intrusion detection agent, for example, may detect intrusion based on abnormal network activities matching some known attacking profiles, hijacking of the platform by some malicious botnet masters, and close correction of this platform with some detected deadly attacks. As we can see, those observed symptoms and attack effects on the platform largely determine the damage status of the platform. A Bayesian network provides a flexible way to quantitatively represent those cause-effect relationships between the set of influencing factors and the damage status of the platform.

To construct a Bayesian network, we need to determine the states (nodes) of the network, the cause-effect relationships among the nodes, and the conditional probability table of each state, which represents how the joint parent nodes impact the probability of each child node. The structure of a Bayesian network can be learned from training data and/or can be specified by incorporating security domain knowledge and applying heuristic rules. Heuristic rules allow the incorporation of user knowledge through the prior probabilistic and relative strength over the associations between the variables. In our example, for the host-based intrusion detection agent, system administrators may determine that the observed attack symptoms all indicate that the platform has been damaged to a certain degree. Therefore, there are strong cause-effect relationships between those factors and the damage status of the platform. Furthermore, the security and resilience ability of the platform also has a significant impact on the degree to which the platform has been damaged. Intuitively, a system with a strong security and resilience capability can effectively defend itself and therefore minimize the chance that the platform has been severely damaged. The encoding of the dependency between all those variables provides the basic structure of the Bayesian network.

We show an example of the Bayesian network constructed for a host-based intrusion detection agent to determine the damage status of a platform (see Figure 2). A node in a Bayesian network is represented graphically by a labeled rectangle. A node can take different values, called the states of the node. As we can see, our Bayesian network has a decision node “Damage status of the platform” and five nature nodes which all influence this decision node: “Privilege elevation of user accounts”, “System behaviors matching known attacking profiles”, “Key system files modified”, “Pre-attack probing activities detected”, and “System security and resilience”. An edge of a Bayesian network represents a causality relationship between two nodes, which is represented graphically by a directed arrow between the two nodes. The meaning of an edge drawn from node $N_1$ to node $N_2$ is that $N_1$ has a direct influence on $N_2$. In our model, the nodes representing the attack effects and damage symptoms and the system’s security and resilience ability jointly determine the possibility of the platform being damaged and to which degree it has been damaged. Therefore, in Figure 2, an edge is directed from each of the five nodes to the decision node “Damage status of the platform”.

To represent the conditional probabilities of parent nodes on a child node, every intermediate and leaf node has a conditional probability table (CPT). For every possible combination of the parent states, there is a row in the child node’s CPT that describes the possible state that the child node should be. The values in the CPT table are added based on expert knowledge and prior intrusion detection and damage assessment data. Those values can be modified as more information is available. In our example, the CPT table associated with the node “Damage status of the platform” provides an incident of causality relationship between the five parent nodes: “Privilege elevation of user accounts”, “System behaviors matching known attacking profiles”, “Key system files modified”, “Pre-attack probing activities detected”, “System security and resilience” and the child node “Damage status of the platform”. The table lists all the possible outcomes of the platform’s damage status given the five influencing factors as represented by the five parent nodes. In our example, the CPT table has 72 entries. Due to page limitation, we will not show the table here.

We used Netica [29] to solve the Bayesian network. Netica is a complete software package to solve problems using Bayesian networks and influence diagrams. The input to the Bayesian network is the detected damage evidence and the system’s security level. As shown in Figure 2, the input includes: (1) no privilege elevation has been detected on any user account; (2) the observed system behaviors and activities match pre-defined attacking profiles to a different degree with 80% being highly matched, 17% matched with a medium degree, and 3% not matched; (3) some key system files have been detected as modified with a confidence level of 90%. Given the challenge to accurately detect all the cases of a file modification and the difficulty to distinguish a normal file change (e.g., due to system update, software patch, or system reconfiguration) from a malicious modification of the file, a set of gradient values with different confidence levels are more appropriate to present different possibilities that some key system
files have been modified; (4) pre-attack probing activities have been detected; and (5) the system security and resilience is evaluated at a high, medium, and low level with a possibility of 60%, 35%, and 5%, respectively. This metric refers to a system’s basic, inherent ability to defend against malicious attacks. Intuitively, the higher level of a system’s security and resilience, the less degree (possibility) of damage to the platform in a security incident. However, it is practically prohibitive to build a “super-strong” system due to cost and other restrictions. Therefore, a system has a different level of security and resilience capability. Given those input, the Bayesian network outputs the possibilities that the platform has been damaged with a varying degree (including the possibility that it is free of damage).

Based on the Bayesian network in Figure 2, the belief-based damage assessment about the platform as provided by the host-based intrusion detection agent is represented by a basic belief assignment (BBA) $m_{\text{host}}(.)$ with the following focal elements and their corresponding belief masses:

- $m_{\text{host}}(\text{Devastating blow}) = 0.301$
- $m_{\text{host}}(\text{Moderate damage}) = 0.552$
- $m_{\text{host}}(\text{Microlesion}) = 0.0205$
- $m_{\text{host}}(\text{Impairment}) = 0.0162$
- $m_{\text{host}}(\text{Free of damage}) = 0.0088$
- $m_{\text{host}}(\text{Frame of discernment} \emptyset) = 0.102$

### 3.2. BBA combination – fusion of multiple agents’ damage assessment outputs

In this section we discuss the combination of multiple damage assessment BBAs into an integrated, more accurate assessment output. As we discussed earlier, each intrusion detection agent has a different level of reliability in terms of assessing a platform’s damage status. Aggregating multiple sources of damage evaluations can effectively reduce the false positive output resulted from some intrusion detection agents, thus ruling out those abnormal results deviated from the majority of other intrusion detection agents’ evaluations. Furthermore, due to its main objectives and design principles, an agent’s damage assessment result about a particular platform are considered with a different level of importance (to the users) in assessing the true damage status of the platform. Therefore, we apply either a discount or reinforcement operation to a BBA before it can be combined with other BBAs. This discounting or reinforcing process is to assign appropriate weights to adjust the agent’s damage assessment output. As shown in Figure 3, the output of the combination process is an integrated BBA $m(.)$, which represents the final more accurate damage assessment about the platform based on multiple sources of intrusion detection evidence. The conflict between those sources of BBAs is represented by the belief mass assigned to the empty set $\emptyset$ in the final combined BBA, i.e., $m(\emptyset)$.

To continue our example, consider the host-based intrusion detection damage assessment $m_{\text{host}}(.)$ as shown above and another network-based intrusion detection damage assessment $m_{\text{net}}(.)$. We assume that $m_{\text{net}}(.)$ have been constructed similarly by following the Bayesian network-based approach as shown earlier. The focal elements and their corresponding belief masses in $m_{\text{net}}(.)$ are shown below:

- $m_{\text{net}}(\text{Devastating blow}) = 0.412$
- $m_{\text{net}}(\text{Moderate damage}) = 0.33$
- $m_{\text{net}}(\text{Microlesion}) = 0.08$
- $m_{\text{net}}(\text{Impairment}) = 0.027$
- $m_{\text{net}}(\text{Free of damage}) = 0.01$
- $m_{\text{net}}(\text{Frame of discernment} \emptyset) = 0.141$

From an intrusion detection perspective, a host-based intrusion detection agent monitors the system
activities and the platform itself more closely as compared with a network-based intrusion detection agent which focuses on monitoring network activities. Therefore, we consider the damage assessment output from a host-based intrusion detection agent with a higher level of reliability (or more important) in a case of damage assessment on a platform. Consequently, an enforcement operation is applied to \( m_{\text{host}}(.) \) and a discounting operation is applied to \( m_{\text{net}}(.) \). We show the enforcement on \( m_{\text{host}}(.) \) first. By applying Formula (4) with \( w_2 = 0.5 \) we have the following enforced BBA \( m_{\text{host}}(.) \): 

\[
\begin{align*}
  m_{\text{host}}("\text{Devastating blow}") &= 0.318 \\
  m_{\text{host}}("\text{Moderate damage}") &= 0.583 \\
  m_{\text{host}}("\text{Impairment}") &= 0.0171 \\
  m_{\text{host}}("\text{Free of damage}") &= 0.0093 \\
  m_{\text{host}}("\text{Frame of discernment}") &= 0.051 
\end{align*}
\]

Similarly, by applying Formula (4) with \( w_1 = 0.3 \) and \( w_2 = 0.7 \) we have a discounted BBA \( m_{\text{net}}(.) \) corresponding to \( m_{\text{net}}(.) \): 

\[
\begin{align*}
  m_{\text{net}}("\text{Devastating blow}") &= 0.2884 \\
  m_{\text{net}}("\text{Moderate damage}") &= 0.231 \\
  m_{\text{net}}("\text{Microlesion}") &= 0.056 \\
  m_{\text{net}}("\text{Impairment}") &= 0.0189 \\
  m_{\text{net}}("\text{Free of damage}") &= 0.007 \\
  m_{\text{net}}("\text{Frame of discernment}") &= 0.3987 
\end{align*}
\]

As we will see in next section, \( m_{\text{f}}(.) \) is used as a basis to calculate the probability distribution of the true damage status of the platform for service migration decision making.

### 3.3. Service migration decision making

In Bayesian decision theory, a decision model includes three components [21]: a set of decisions, a betting frame, and a cost function given an action to take and a state of the system. We next discuss those three components.

A set of decisions, denoted as \( D \), represent the actions that can be taken in a particular application. In our model, the set of actions include the possible security actions to respond to a malicious attack. More specifically, the set of decisions includes: (1) migrating the critical services from their compromised platforms to other clean, healthy platforms so that those services can be continuously executed there; (2) conducting thorough diagnosis to fully repair and recover the damaged system and its components in a timely manner in order to minimize any interrupt of the services; (3) performing relatively lightweight security fixing and patching to restore the system and its components; and (4) accepting the risk and thus taking no action. Therefore, \( D \) is defined in our model as \( D = \{ \text{Service migration, System healing and restoration, System mending and refurbishment, Risk acceptance} \} \). As we can see, different security actions are best chosen based on different degrees of damage on the platform. For example, service migration is best used in case of a devastating attack when some platforms have been severely compromised and thus it is very difficult to fully recover the entire platform within a short period of time. In this situation, since the critical services must be continuously provided, the best strategy is to temporarily migrate those services to other clean, healthy platforms so that they can be executed on those new platforms. Once their original platforms have been fully recovered, those critical services can be relocated back and continue to finish their operations. System healing and restoration is a better choice in case of a moderate damage on a platform when a timely repair is completely possible given a well-defined diagnosis and repair plan. Such an action is more cost efficient since it does not require expensive activities to allocate resources and migrate services to other platforms. System mending and

\[
\begin{align*}
  m_{\text{f}}("\text{Moderate damage}") &= 0.27563 \\
  m_{\text{f}}("\text{Microlesion}") &= 0.0127 \\
  m_{\text{f}}("\text{Impairment}") &= 0.0081 \\
  m_{\text{f}}("\text{Free of Damage}") &= 0.00413 \\
  m_{\text{f}}("\text{Frame of discernment}") &= 0.020337 
\end{align*}
\]
refurbishment is the most appropriate action to take in case of only a minor damage on a platform since only a relatively lightweight repair work is required to recover the system. Finally, risk acceptance implies that no action will be taken. This is the case when the platform is assessed as damage free or the platforms is damaged but the cost to recover the system is high which does not justify the security actions to take.

A betting frame, denoted as \( \Gamma \), represents the states of nature in the particular application to make a decision. \( \Gamma \) is often equal to \( \Omega \). In our example, the betting frame represents the possible damage status of a platform. Therefore, we have \( \Gamma = \{ \text{Devastating blow, Moderate damage, Microlesion, Free of damage} \} \).

In a security incident, the true damage status of a platform is rarely known for sure. A probability function \( P: \Gamma \rightarrow [0, 1] \) is defined to represent the probability measure of each element in \( \Gamma \). This probability function is obtained from the combined basic belief assignment \( m_{\Gamma}(.) \). By applying Formula (1), we can get the probability measure of each element in \( \Omega \) from the given BBA \( m_{\Gamma}(.) \), i.e., \( P(A) = \text{BetP}_{\Gamma}(A) \) for \( \forall A \subseteq \Omega \). In our example, the probabilities of different damage statuses on the platform are calculated as shown below:

\[
\begin{align*}
P(\text{"Devastating blow"}) &= 0.4349 \\
P(\text{"Moderate damage"}) &= 0.51148 \\
P(\text{"Microlesion"}) &= 0.036967 \\
P(\text{"Free of damage"}) &= 0.0166 \\
\end{align*}
\]

The third and final component of our decision model is a cost function \( CT: D \times \Gamma \rightarrow R \), where each cost element \( CT(d, r) \in R \) is specified for a decision \( d \in D \) made in the true state of nature \( r \in \Gamma \). As we can see, in addition to the cost to carry a security action itself, this cost element also depends on the true state of the platform, i.e., whether the platform has been damaged and to which degree it has been damaged.

To better understand the value of each cost element \( CT(d, r) \), we have developed a reference table (see Table 1), which lists the cost of performing each security action \( d \) in every platform damage state \( r \). Such a cost element \( CT(d, r) \) is constructed in terms of the combined result of (1) the cost to carry the security action itself (e.g., service migration, system healing and restoration) and (2) the potential loss given the true platform damage state in case that the security action is not sufficient to cope with the attack and the resulted damage. Let's consider the cost element \( CT(\text{"Devastating blow"}, \text{"System mending and refurbishment"}) \). It is composed of the cost for performing system mending and refurbishment and any cost resulted from potential loss of the services in case of a severe damage of the platform due to the fact that the security action cannot sufficiently mask or handle the attack (since the damage is so severe and simply performing system mending and refurbishment is not enough to fully recover and restore the system). As another example, the cost element \( CT(\text{"Moderate damage"}, \text{"Service Migration"}) \) only includes the cost for performing a service migration but there is no any potential loss resulted from a moderate damage of the platform. This is the case because the security action (service migration) has sufficiently addressed the moderate damage caused by the attack – by migrating the services to other platforms in a timely manner, the moderately damaged platform can be repaired immediately and this damage will not affect the services which has been executed somewhere else. To make those concepts clear, we use the notion \( C(d) \) in Table 1 to represent the cost for performing a security action \( d \). Similarly, the notion \( C(r) \) is used to represent the loss when a platform is in a damage state \( r \) but the security action chosen is insufficient to respond to the attack and to mask the damage. Logically, those cost values should be set in such a way that the cost of service migration is higher than that of system healing and restoration, which is higher than the cost of system mending and refurbishment, i.e., \( C(\text{SM}) > C(\text{SHR}) > C(\text{SMR}) \). Similarly, any system loss resulted from a devastating blow is higher than the loss in case of a moderate damage, which is higher than the loss in case of only a microlesion damage, i.e., \( C(\text{DB}) > C(\text{MD}) > C(\text{M}) \).

Eventually, the service migration decision model is to determine which security action \( d^{*} \in D \) to take in a security incident so that it results in a minimum overall cost given the betting frame \( \Gamma \) and the cost function \( CT(.) \). Let \( Cost(d) \) denote the weighted cost to choose a security action \( d \in D \) given all the possible damage states of the platform. This cost is calculated as:

\[
Cost(d) = \sum_{r \in \Gamma} (CT(d, r) \cdot P(r))
\]
Therefore, the decision model is to identify a rational decision \(d^* \in D\) so that \(Cost(d^*) \rightarrow \min\).

Continue our example, we set the following cost values for performing different security actions and the potential loss of services in case that the security action is not sufficient to cope with the attack and recover from the damage:

\[
\begin{align*}
C(SM) &= 11, \quad C(SHR) = 8, \quad C(SMR) = 4, \quad C(RA) = 0 \\
C(DB) &= 21, \quad C(MD) = 9, \quad C(M) = 3, \quad C(FD) = 0
\end{align*}
\]

Given the above values, from Formula (5), we can calculate the cost \(Cost(d)\) for each decision \(d \in \{\text{Service migration, System mending and restoration, Risk acceptance}\}\) as shown below:

\[
\begin{align*}
\text{Cost("Service migration")} &= P("\text{Devastating blow}") \times C(SM) + P("\text{Moderate damage}") \times C(SM) \\
&\quad + P("\text{Microlesion}") \times C(SM) + P("\text{Free of damage}") \times C(SM) = 11 \\
\text{Cost("Service halting and restoration")} &= P("\text{Devastating blow}") \times (C(SHR) + C(DB)) \\
&\quad + P("\text{Moderate damage}") \times (C(SHR) + C(MD)) + P("\text{Microlesion}") \times C(SHR) + P("\text{Free of damage}") \times C(SHR) = 17.13 \\
\text{Cost("System mending and refurbishment")} &= P("\text{Devastating blow}") \times (C(SMR) + C(DB)) \\
&\quad + P("\text{Moderate damage}") \times (C(SMR) + C(MD)) + P("\text{Microlesion}") \times C(SMR) + P("\text{Free of damage}") \times C(SMR) = 23.87 \\
\text{Cost("Risk acceptance")} &= P("\text{Devastating blow}") \times C(DB) + P("\text{Moderate damage}") \times C(MD) \\
&\quad + P("\text{Microlesion}") \times C(M) + P("\text{Free of damage}") \times C(M) = 13.84
\end{align*}
\]

Then, it is clear that the decision “Service migration” is the most appropriate security action to take in this security incident since it results in the minimum cost of 11. Therefore, the rational decision is to migrate the services from their current platforms to other clean, healthy platforms and allow those services to be executed on those new platforms. In this way, the services can be continuously provided with little or no interruption.

4. Conclusion

Service migration is an important defensive security technique in responding to malicious attacks. Determining whether a service migration is necessary, however, is the first and most important decision to make before any resources can be allocated and any service migration activities can start. In this paper, we proposed a transferable belief-based decision model to represent a damage assessment about a platform provided by each intrusion detection agent and then combine multiple sources of the damage assessments into a more accurate final damage assessment about the platform. The basic belief assignment (BBA) to present an agent’s belief is constructed using a Bayesian network-based approach, where a set of influencing factors are identified which quantitatively determine the damage status of the platform. A service migration decision model is presented, which facilitate users in making the best decision regarding whether a service migration is necessary and, if so, why it is the most effective and efficient action to take given the current damage assessment on the platform.

Acknowledgment
This material is based upon work supported by the US Air Force Office of Scientific Research (AFOSR) under Award FA9550-12-1-0131. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the author and do not necessarily reflect the views of AFOSR.

References


