Insider threat likelihood assessment for flexible access control

Sofiene Boulares, Kamel Adi, and Luigi Logrippo

Département d'informatique et d'ingénierie
Université du Québec en Outaouais
Gatineau, QC, Canada
{bous42,kamel.adi,luigi.logrippo}@uqo.ca

Abstract. Users who request to access protected objects must obtain the authorization of access control systems. Among the elements of decision for such systems should be the risk of authorizing accesses under various assumptions, and one of the notions of risk is threat likelihood. Access control systems deals essentially with insider threats coming from people within the organization, such as employees, business associates or contractors, who could violate access control policies. We present in this paper a new approach for insider threat likelihood assessment for secrecy and integrity properties by considering reading and writing operations within the context of access control systems. Access operations, the trustworthiness of subjects, the sensitivity of objects, and the applied security countermeasures are all considered in the assessment of the likelihood of this category of insider threats. Both qualitative and quantitative assessments are provided. Hence our approach makes it possible to compare and calculate the likelihoods of these insider threats, leading to more flexible and more informed access control decisions in various situations.

Keywords: Information Security, Access control, Insider threat, Threat likelihood assessment, Risk assessment.

1 Introduction

Access control to data is governed by means of policies determining whether a subject (or user) has the right to execute an action (read, write, etc.) on an object (file, database table, etc.). Conventional access control systems are rather rigid, since they consider only properties of subjects and objects to take decisions. Risk-based access control offers mechanisms to take access decisions by determining the security risks associated with access requests, thus achieving greater flexibility. In the following, we present two examples of situations where this type of access control could be useful. Consider:

Example 1: A situation where a workflow architect asks an IT security specialist to determine which combinations of operations are less risky for the tasks composing a workflow, given the subjects, objects and actions involved in each operation. Which combination should be chosen for each task and on what basis?

Example 2: An emergency situation where there is an urgent need to consult a patient's file which is classified *Top Secret*. However, of all the doctors present, none has the clearance to read the file. Which of the doctors present should be chosen to read the file and on what basis?
In both examples above, the decision could be based on the evaluation of access risks, by selecting the combination giving the lowest risk value in the first example and the doctor yielding the lowest risk value in the second example.
An access control system that can give employees risky accesses can cause insider security incidents. According to the US firm Forrester Research, insider incidents within organizations represent 46% of security breaches [21]. In addition, the survey Global Corporate IT Security Risks 2013 [10], conducted by Kaspersky Lab, shows that 85% of companies worldwide have experienced an insider computer security incident.

Bishop et al [2] distinguish two categories of insider threats:

1. violation of access control policy by using authorized access,
2. violation of security policy by obtaining unauthorized access.

The first category includes cases where an employee uses his legitimate access to perform an action that violates the access control policy: discloses sensitive data to a third party, releases information to untrusted environments, provides information to employees who don't have the right to know them, steals property or information for personal gain, etc. The second category includes cases where an employee exploits a vulnerability in the system such as a buffer overflow [14] to obtain an access which he does not have.

The approach for threat likelihood estimation of access requests that we present in this paper deals with the first category of insider threats. Indeed, our method can be seen as an approach to estimate the threat likelihood of the violation of an access control policy, caused by the authorization of other access requests.

This paper is an extended version of our short paper [3]. In this paper, motivational examples, explanations, figures (Figures 1 and 2), formulae (formulae 4 and 5) and new examples have been added. This paper proves the correctness of formula (2) and presents our write threat likelihood assessment approach when secrecy is intended and our threat likelihood assessment approach when integrity is intended. Furthermore, the literature review has been much enhanced.

The rest of the paper is organized as follows. Section 2 presents an overview of our work and the contribution of this paper. In Sections 3 and 4, we present respectively our threat likelihood assessment approach for secrecy and integrity. In Section 5, we compare our work with notable work of the literature and we present the limitations of our approach. We draw conclusions for this paper and outline opportunities for future work in Section 6.

2 Overview and contribution

The access control model we propose, authorizes accesses that would be refused by the traditional models characterized by predefined access decisions. Assessing

the threat likelihood of different types of events with their predicted impacts is a common way to assess IT risks. OWASP [18] defines the risk R as "the product of the likelihood L of a security incident occurring times the impact I that will be incurred by the organization due to the incident, that is: $R = L \times I$". In the following section, we will adapt this risk definition to develop a risk assessment function for access control systems.

Our approach to assess the risk of access requests is based on the Mehari approach that gives guidelines for security assessment [6]. This approach differentiates between the *intrinsic threat likelihood* which is the probability that the risk in question will occur in the absence of security countermeasures and *threat likelihood* which considers the reduction of risk by application of countermeasures [6]. The security countermeasures could be devices, procedures, or techniques that reduce the likelihood of threat on the security of information that is processed, stored or transmitted. Such reduction can be achieved by eliminating or preventing the threat. According to the Glossary presented by Information security Today [11], countermeasures consist in the deployment of security services to protect against a security threat. A synonym for security countermeasure is security control [19, 22]. Examples of such countermeasures are enabled access logs, data encryption, etc.

The steps of our approach are summarized in the following:

1. assessment of the *intrinsic threat likelihood* of the access request by considering the information flow which could result if the access was allowed.
2. assessment of the effect of the security countermeasures permitting the mitigation of the *threat likelihood* of an access request.
3. assessment of the *threat likelihood*. In this step we answer the following question: "How likely is the occurrence of the risk made possible by the access request?".
4. assessment of the impact. In this step we answer the following question: "If the request was allowed what would be the extent of the damage?".
5. assessment of the risk.
6. decision on whether the risk is acceptable and hence to allow or deny the request.

Note that steps 4, 5 and 6 are out of the scope of this paper. They will be the subject of future publications.

Let us assume the existence of the following entities: S a set of subjects, O a set of objects, A a set of actions, L_c a set of secrecy levels, L_i a set of integrity levels and SC a set of security criteria. We limit the set A to two actions, read and write, which will be collectively called *accesses* and abbreviated respectively r and w. We limit the set SC to secrecy and integrity abbreviated respectively c and i (confidentiality is a term sometimes used in the literature instead of secrecy.).

We adapt the risk definition presented in this section to define a risk assessment function for access control systems. Specifically, the risk of permitting a subject s to perform an action a on an object o for a security criterion sc

can be given by the following function (where $\times$ is not necessarily the usual multiplication operator):

$$Risk(s, a, o, sc) = Threat\_likelihood(s, a, o, sc) \times Impact(a, o, sc) \quad (1)$$

$Threat\_likelihood(s, a, o, sc)$ represents the likelihood of a threat that a subject s (threat source) may present towards an object o (threat target) when it executes an action a in the context of the security criterion sc. $Impact(a, o, sc)$ represents the adverse impact on the satisfaction of security objectives that can result from successfully performing action a on object o in the context of the security criterion sc.

2.1 Contribution

In order to assess the risk of *access requests*, we have focused our effort on developing a qualitative and quantitative approach for determining *threat likelihood* by considering the parameters of information flow, trustworthiness of subjects, sensitivity of objects and security countermeasures. To our knowledge, few works on assessing the risk of access requests have explicitly provided approaches to assess the *threat likelihood* that subjects may present towards data objects [1, 5, 16, 15]. However, in organizational contexts, such assessment is important to make the risk assessment method repeatable and accurate.

3 Assessment of threat likelihood when secrecy is intended

The assessment of threat likelihood can be done for secrecy or integrity. In this section, we propose our approach to assess threat likelihood on secrecy. We assume that we deal with systems where subjects and objects are classified by secrecy levels. Our approach considers the following factors:

- the intended security criteria (secrecy in this section),
- the requested action (read or write),
- the secrecy level of subjects requesting access,
- the secrecy level of objects to be accessed,
- the security countermeasures.

Information flow is the transfer of information between subjects and objects. Authorizing a read action creates an information flow from an object to a subject and authorizing a write action creates an information flow from a subject to an object. We assume that, when data secrecy is addressed, threat likelihood depends on the importance of information flow between objects and subjects, determined by the difference between their secrecy levels. In other words we can assume a correlation between the information flow that may result from permitted accesses and the threat likelihood.

In this paper, we present an information flow approach to assess the threat

likelihood that subjects may present towards objects. Such assessment is needed to evaluate the risk of access requests. According to [20, 24], secrecy is related to disclosure of information. In our approach, the likelihood of threat on secrecy increases when information flows down. Consider, for example, the information flow when a *Top Secret* subject writes in a *Public* object, such information flow is more important than the one when the same subject writes in a *Secret* object. In the first case, *Top Secret* information could be leaked to the public, in the second case this information would remain secret. It is reasonable to assume that the threat likelihood would be higher in the first case. The reasoning for integrity is dual.

We define a total order on L_c and for each secrecy level in L_c, we assign a numerical value in accordance with the defined order. For example, if L_c = {Unclassified, Restricted, Classified, Secret, TopSecret}, then the value Unclassified corresponds to the number 1, Restricted corresponds to the number 2 and so on. To simplify the notation, L_c will be considered to be understood and so it won't need to be mentioned: in each system, there is only one L_c which applies to subjects as well as objects. Throughout this paper, the following functions will be needed to develop our approach:
• $csl : S \rightarrow L_c$ formally represents the assignment of secrecy levels to subjects that reflects the trust bestowed upon each of them by the owner of the data.
• $col : O \rightarrow L_c$ formally represents the assignment of secrecy levels to objects that reflects the protection needs of the data.

3.1 Defining "threat likelihood"

As discussed earlier in Section 2 and precisely in relation to formula (1), threat likelihood metrics are a prerequisite for the computation of risk metrics. To assess likelihood of threat on secrecy, we use the intuition behind the *Bell LaPadula* model including mandatory rules preventing the flow of information from a high level of secrecy to a lower one [20]. This model has a binary view of threat likelihood [5]. In the case of a request from subject s to read object o, *threat likelihood* is equal to 0 if $col(o) \leq csl(s)$ and equal to 1 otherwise. The reverse is true in the case of write requests, threat likelihood is equal to 0 if $csl(s) \leq col(o)$ and equal to 1 otherwise.

Instead of adopting the binary vision of the *Bell LaPadula* model to assess the threat likelihood of read and write requests, we propose to consider the following principles, which replace the properties of the *Bell LaPadula* model: we consider that permitting a subject s to read an object o, such that $csl(s) < col(o)$ or permitting a subject s to write in an object o, such that $csl(s) > col(o)$, presents by itself a measurable threat likelihood, independently of what might happen to the information that is accessed.

In this section, we define the "threat likelihood" in the context of access control systems. In particular, the likelihood of threat on secrecy of accesses is defined as follows:
Case 1: we say that the likelihood of threat on secrecy is non null if a subject $s \in S$ is able to read an object $o \in O$, such that $csl(s) < col(o)$. But for any

attempt by a subject s to read an object o, such that $csl(s) \geq col(o)$ the threat likelihood is null. Any measure of read threat likelihood on secrecy in the first case is affected by the following two general principles:

- **Principle 1:** the likelihood of threat on secrecy increases (or decreases) as the object's secrecy level increases (respectively decreases).
- **Principle 2:** the likelihood of threat on secrecy increases (or decreases) as the subject's secrecy level decreases (respectively increases).

Case 2: we also say that the likelihood of threat on secrecy is non null if a subject s is able to write in an object $o \in O$, such that $csl(s) > col(o)$. But for any attempt by a subject s to write in an object o, such that $csl(s) \leq col(o)$ the threat likelihood is null. Any measure of write threat likelihood on secrecy in the second case is affected by the following two general principles:

- **Principle 3:** the likelihood of threat on secrecy increases (or decreases) as the object's secrecy level decreases (respectively increases).
- **Principle 4:** the likelihood of threat on secrecy increases (or decreases) as the subject's secrecy level increases (respectively decreases).

We define a function $Threat\_likelihood : S \times A \times O \times SC \rightarrow [0, 1]$ that represents the threat likelihood value of a subject $s \in S$ requesting an action $a \in A$ on an object $o \in O$ when a security criterion $sc \in SC$ is intended (in this section, $sc = c$). We use relation $<_T$ to denote an ordering on *likelihoods of threats* of a set of subject-object accesses. In particular, we define $<_T$ in the following way: $(s, a, o, sc) <_T (s', a', o', sc)$ iff $Threat\_likelihood(s, a, o, sc) < Threat\_likelihood(s', a', o', sc)$. The relation $<_T$ allows threats likelihoods to be compared.

3.2 Read threat likelihood assessment for secrecy

In this section, we describe the settings of a scenario that will be used in the rest of the paper for motivating our approach. We assume the existence of the following subjects: s_1, s_2, s_3, s_4, s_5 and s_6. Table 1(a) illustrates the secrecy levels of these subjects. Let us also consider three objects o_1, o_2 and o_3. Table 1(b) shows the secrecy levels of these objects.

3.2.1 Read threat likelihood assessment for secrecy: qualitative approach

Assume that access for data objects has been requested by subjects who are employees of the business that owns the requested data objects (trusted and reliable to some degree by the system). In this case, data owners might be more concerned about the secrecy levels of objects than the secrecy levels of subjects. Hence, our approach for threat likelihood assessment in this paper is primarily based on the *secrecy levels of objects*.

Let us consider Table 2 which could be given by a workflow architect to an IT security specialist. The security specialist is asked to define a set of tasks

Subjects	Secrecy levels
s_1	4
s_2	4
s_3	3
s_4	2
s_5	1
s_6	1

(a)

Objects	Secrecy levels
o_1	4
o_2	3
o_3	2

(b)

Table 1. Secrecy levels for running examples.

composing a workflow by selecting the least likely threatening combinations of subjects, objects and actions for the secrecy of data. We see that task T_1 can be executed by s_6 reading from objects o_1 or o_2, task T_2 can be executed by either s_4 or s_6 reading from o_2 and task T_3 can be executed by either s_5 or s_6 reading from o_1.

Task	Subjects	Objects	Action
T_1	s_6	o_1, o_2	read
T_2	s_4, s_6	o_2	read
T_3	s_5, s_6	o_1	read

Table 2. Possible accesses by potential subjects to potential objects.

Example 1: According to **Principle 1** stated in Section 3.1, allowing s_6 to read object o_1 has a greater likelihood of threat on secrecy than allowing s_6 to read object o_2, i.e.:

$$(s_6, r, o_2, c) <_T (s_6, r, o_1, c)$$

This is because the secrecy level of object o_1 is higher than the secrecy level of object o_2. In the above example, we were able to determine which access has a greater threat likelihood by simply comparing the secrecy levels of the two objects. However, such a technique is no longer sufficient when object secrecy levels are the same as we can see in the following example.

Example 2: Determining the least threatening access for task T_2 by using **Principle 1** is not possible. This is because the subjects s_4 and s_6 request the access to the same object. In this case we use **Principle 2** stated in Section 3.1. According to this principle, allowing s_6 to read object o_2 has a greater likelihood of threat on secrecy than allowing s_4 to read object o_2. i.e.:

$$(s_4, r, o_2, c) <_T (s_6, r, o_2, c)$$

This is because s_6 has a secrecy level of 1 which is lower than s_4's secrecy level of 2.

To consider the effect of countermeasures in the reduction of threat likelihood, we define the additional following principle:

Principle 5: the likelihood of threat on data security increases as the effect of

security countermeasures reducing the threat likelihood decreases.
The following example shows the importance of the consideration of the countermeasures in our approach:
Example 3: Let us consider task T_3 where allowing s_6 to read object o_1 has the same likelihood of threat on secrecy as allowing s_5 to read object o_1. This is because s_5 and s_6 have the same secrecy level of 1. We also know that all subjects are aware of the terms of the security policy (existence of penalties, etc.) and that all accesses of s_5 are logged whereas access logs are not enabled for subject s_6. Then, according to **Principle 5**, allowing s_6 to read object o_1 has a greater likelihood of threat on secrecy than allowing s_5 to read object o_1. i.e.:

$$(s_5, r, o_1, c) <_T (s_6, r, o_1, c)$$

This is because enabling access logs represents a dissuasive countermeasure which aims at making it less likely that the subject s_5 will actually perform malicious actions if he is aware that this action can be attributed to him and can lead to severe penalties [6].

Examples 1, **2** and **3** suggest the following method:
Method 1: A read threat likelihood assessment technique that is primarily based on object secrecy levels should support the following:

1. always apply **Principle 1** (that is, read threat likelihood always increases as object secrecy level increases),
2. whenever object secrecy levels are the same, apply **Principle 2** (that is, read threat likelihood increases as subject secrecy level decreases),
3. apply **Principle 5** (that is, threat likelihood of accesses increases (or decreases) as the effect of security countermeasures reducing the threat likelihood decreases (respectively increases).

Based on the above comparisons, the least threatening combinations of subjects, objects and actions on secrecy according to **Method 1** are presented in Table 3.

Task	Subjects	Objects	Action
T_1	s_6	o_2	read
T_2	s_4	o_2	read
T_3	s_5	o_1	read

Table 3. The least threatening combinations according to **Method 1**.

3.2.2 Read threat likelihood assessment for secrecy: quantitative approach

In this section, we present a quantitative approach for threat likelihood assessment and we show why and how it could be useful. To this end, we start with the following example:
Example 4: Table 4 shows that task T_4 can be executed by either s_3 or s_4

reading from o_1. The two subjects request access from two distant sites where s_3 is connected via an unencrypted public network and s_4 via VPN which is a countermeasure that reduces threat likelihood by preventing disclosure of information. Indeed, VPNs typically allow only authenticated remote access using tunnelling protocols and encryption techniques. According to **Principles 1** and

Task	Subjects	Objects	Action
T_4	s_3, s_4	o_1	read

Table 4. Possible combinations to define task T_4.

2, allowing s_4 to read object o_1 has a greater likelihood of threat on secrecy than allowing s_3 to read object o_1. However, **Principle 5** tells us that this may not be true in the presence of countermeasures, that can reduce the threat likelihood of s_4 reading o_1. Hence the need to quantify the countermeasures effect and the threat likelihood of access requests.

Priority orders only permit a threat likelihood comparison in sets of accesses. However, quantitative measures which correspond to this threat likelihood ordering may be useful, such as in the case where there are many requests that we want to compare. There can be many different formulas which respect the properties of our approach and can measure the threat likelihood of granting access. In this section, we propose a formula and describe its construction.

ISO / IEC 27001 [9] requires regular verification of computer security. In order to determine to which extent the countermeasures are producing the desired outcome to meet the security requirements, the security administrator measures the contribution of the implemented security countermeasures in the reduction of risks. In this work, we adopt the concepts of Mehari methodology [9] to consider the effect of security countermeasures in the calculation of threat likelihood. We introduce a set of rules in Table 5 to determine the countermeasures that reduce threat likelihood of access requests and their effects. Each rule determines a countermeasure and its effect corresponding to an access request identified by the subject's security level, the object's security level, the action requested and the security criteria intended.

The content of Table 5 could be determined by the security administrator. It shows a representation of all possible read accesses by subjects to objects when secrecy is intended. Note that for an attempt by a subject s to read an object o, such that $csl(s) \geq col(o)$ the threat likelihood is null. Hence, Table 5 doesn't show the countermeasures and their values along or below the diagonal of the table. Each table entry $[i, j]$ includes a set of couples (measure, value) that represents the countermeasures and their contribution in the reduction of threat likelihood of a subject s reading an object o, where $csl(s) = i$ and $col(o) = j$. The sum of all countermeasures values in each entry is bound between 0 and 1.

The rule corresponding to the entry [1, 4] shows that if a subject having a secrecy level 1 reads an object having a secrecy level 4, then countermeasure

m_3 can reduce the likelihood of threat on secrecy by 0.5. The rule of the entry [2, 4] shows that if a subject having a secrecy level 2 reads an object having a secrecy level 4, then the countermeasures m_3 and m_4 can respectively reduce the likelihood of threat on secrecy by 0.5 and 0.2.

$Counter(s, a, o, sc)$ denotes the sum of the effects of the different implemented countermeasures to reduce threat likelihood if s executes an action a on an object o when the security criteria sc is intended. For example, we can see from Table 5 that if a subject s having a secrecy level of 1 requests to read an object o having a secrecy level of 5 when secrecy is intended and all three countermeasures are applied, we have $Counter(s, r, o, c) = 0.5+0.2+0.2 = 0.9$. Note that we consider that the countermeasures are independent and perfectly implemented and we don't consider their partial implementation that could result in a lower level of reduction of the threat likelihood.

We define the following additional principles for the calculation of the threat likelihood of access requests:

- **Principle 6:** The threat likelihood of an access request is equal to zero, if the cumulative effect of the corresponding security countermeasures is equal to or greater than the value of the intrinsic threat likelihood.
- **Principle 7:** The threat likelihood of an access request increases (or decreases) when the intrinsic threat likelihood increases (respectively decreases).
- **Principle 8:** The value of the threat likelihood of an access request is bound between 0 and 1.

Subjects secrecy levels	Objects secrecy level 1	Objects secrecy level 2	Objects secrecy level 3	Objects secrecy level 4	Objects secrecy level 5
1		$(m_5, 0.5)$	$(m_5, 0.5)$	$(m_3, 0.5)$	$(m_1, 0.5)$ $(m_2, 0.2)$ $(m_4, 0.2)$
2			$(m_2, 0.2)$ $(m_4, 0.2)$	$(m_3, 0.5)$ $(m_4, 0.2)$	$(m_2, 0.2)$ $(m_4, 0.2)$
3				$(m_3, 0.5)$	$(m_4, 0.2)$
4					$(m_4, 0.2)$
5					

Table 5. The effect of countermeasures in the reduction of the read threat likelihood.

We now introduce the concept of threat likelihood indexing when secrecy is intended. We assign a numerical value from the set $\{0, \cdots, |L_c| - 1\}$ that represents the threat likelihood index of a secrecy level *clevel* in L_c. For example, in the case of read accesses when secrecy is intended, from the point of view of subjects, we expect the threat likelihood to increase as subject secrecy levels decrease. Hence, subject threat likelihood index values decrease with subject secrecy levels. We write $\widehat{clevel}$ to denote an entity (subject or object) threat likelihood index that decreases with the entity secrecy level. Formally, $\widehat{clevel} = |L_c| - clevel$. For example, $(\widehat{Secret}) = 5 - 4 = 1$. However, from the point of view of

objects, we expect threat likelihood to increase as object secrecy levels increase. Hence, object threat likelihood indexes increase with object secrecy levels. We write $\widehat{clevel}$ to denote an entity threat likelihood index that increases with entity secrecy levels. Formally, $\widehat{clevel} = clevel$ -1. For example, $\widehat{Secret} = 4 - 1 = 3$.

In this paper, we assume that $|L_c| = 5$, hence there can be at most $5 \times 5 = 25$ combinations of subject-object accesses. We define a function $Intrinsic : S \times A \times O \times SC \rightarrow [0, 1]$ that represents the intrinsic threat likelihood value of a subject $s \in S$ requesting an action $a \in A$ on an object $o \in O$ when a security criterion $sc \in SC$ is intended. In this section, $sc = c$.

$$Intrinsic(s, r, o, c) = \begin{cases} \frac{(|L_c| \times \widehat{col(o)} + \widehat{csl(s)})}{(|L_c|)^2 - 1}, \textbf{if } csl(s) < col(o) \\ 0, \quad \textbf{Otherwise}. \end{cases} \tag{2}$$

A formula that respects the principles of **Method 1**, **Principles 6**, **7** and **8** for measuring the threat on secrecy likelihood of granting read access to a subject s for an object o, is given below:

$$Threat\_likelihood(s, r, o, c) = \begin{cases} Intrinsic(s, r, o, c) - Counter(s, r, o, c), \\ \textbf{if } csl(s) < col(o) \textit{ and} \\ Counter(s, r, o, c) < Intrinsic(s, r, o, c) \\ 0, \quad \textbf{Otherwise}. \end{cases} \tag{3}$$

The numerator of formula (2) is intuitive. Since we require that more importance be given to the threat likelihood index of objects, we multiply the object threat likelihood index by $|L_c|$ that equals the cardinality of the set of secrecy levels L_c. Then, we add the threat index of the subject. The numerator of the formula maps all possible read accesses by subjects to objects into an interval $[0 \cdots (|L_c|^2) - 1]$, where a higher value represents a greater threat likelihood. In order to have intrinsic likelihood threat values into an interval $[0, 1]$, we divide the value obtained from the numerator by $|L_c|^2$ - 1. In formula (3), we subtract the value representing the effect of the different implemented countermeasures corresponding to the request in question. The resultant value represents the object-based read threat likelihood value that respects the principles of **Method 1**, **Principles 6**, **7** and **8**.

If we apply formula (3) to **Example 1** stated in 3.2.1, we get the following: $Threat\_likelihood(s_6, r, o_1, c) = Intrinsic(s_6, r, o_1, c)$ - $Counter(s_6, r, o_1, c) =$ 0.79 - 0.5 = 0.29 (1) and $Threat\_likelihood(s_6, r, o_2, c) = Intrinsic(s_6, r, o_2, c)$ - $Counter(s_6, r, o_2, c) = 0.58$ - 0.5 = 0.08 (2). From (1) and (2), we have $Threat\_like\ lihood(s_6, r, o_1, c) > Threat\_likelihood(s_6, r, o_2, c)$.

The graph shown in **Figure 1** is obtained by formula (2). It illustrates that the required characteristics are retained. Indeed, for each subject s and object o where $csl(s) \geq col(o)$, the intrinsic threat likelihood is equal to zero. Furthermore, if a subject s attempts to read from an object o, such that $csl(s) < col(o)$ the intrinsic threat likelihood is not null. This satisfies the assumptions of **Case 1** stated in 3.1. The left most side of **Figure 1** shows that with the increase

in secrecy levels of objects, and the decrease of secrecy levels of subjects, the intrinsic threat likelihood increases. This satisfies **Principles 1** and **2**. The right most side of **Figure 1** shows that with the increase of the secrecy levels of subjects, and the decrease of secrecy levels of objects, the intrinsic threat likelihood decreases. This satisfies **Principles 1** and **2**. The values of **Figure 1** show that the threat likelihood values are bound between 0 and 1. This satisfies **Principle 8**.

The graph shown in **Figure 2** can be obtained by formula (2). It shows that

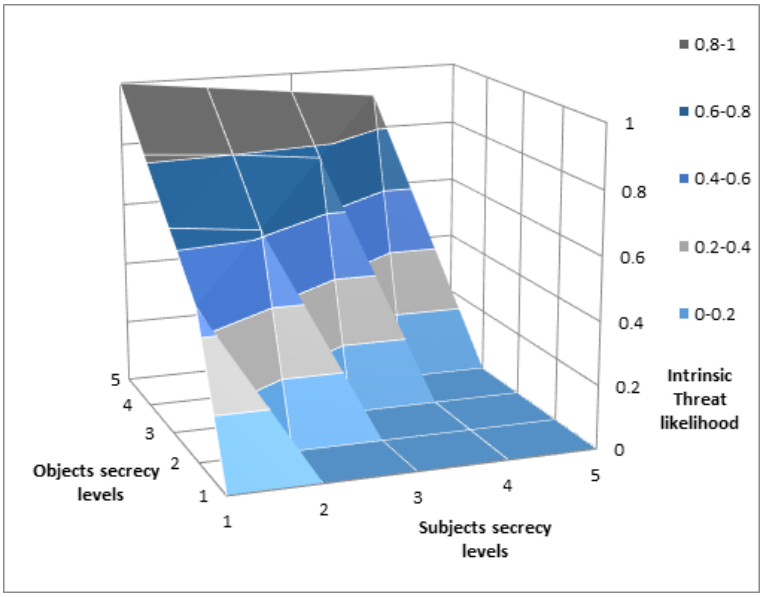

Fig. 1. Behavior of the values of the intrinsic threat likelihood in function of the secrecy levels of subjects and objects.

Principles 5, 6, 7 and **8** are satisfied. The figure shows that when the effect of the security countermeasures increases (or decreases) the threat likelihood decreases (or increases respectively). This satisfies **Principle 5**. The right most side of the figure shows that the threat likelihood of an access request is null if the value of the corresponding security measures is greater than or equal to the value of the intrinsic threat likelihood. This satisfies **Principle 6**. The left side of the figure shows that with the increase in the intrinsic threat likelihood and the decrease of the effect of countermeasures values, the threat likelihood increases. This satisfies **Principle 7**. The right side of the figure shows that, with the decrease of the intrinsic threat likelihood and the increase of the effect of the corresponding countermeasures, the threat likelihood decreases. This

satisfies **Principle 7**. The values of threat likelihood are between 0 and 1. This satisfies **Principle 8**.

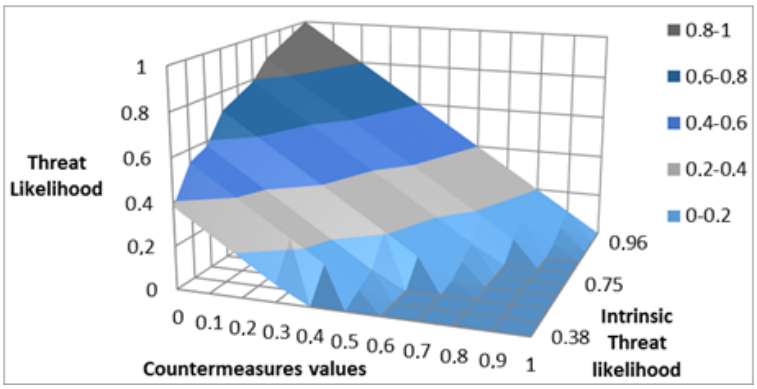

Fig. 2. Behavior of threat likelihood values in function of countermeasures and the intrinsic threat likelihood.

3.2.3 Proof of correctness

This section shows that formula (2) satisfies **Principles 1**, **2**, **5**, **6**, **7** and **8**. Suppose that $|L_c|$ is fixed, then we have the following:

- When $col(o)$ increases (or decreases), $\overbrace{col(o)} = col(o) - 1$ increases (or decreases respectively). Consequently, for any given subject, $Intrinsic(s, r, o, c)$ increases (or decreases) as $col(o)$ increases (or decreases respectively). Hence, $Threat\_likelihood(s, r, o, c)$ increases (or decreases) as $col(o)$ increases (or decreases respectively). We conculde that formula (3) satisfies **Principle 1**.
- When $csl(s)$ increases (or decreases), $\widehat{csl(s)} = |L_c| - csl(s)$ decreases (or increases respectively). Consequently, for any given object, $Intrinsic(s, r, o, c)$ decreases (or increases) as $csl(s)$ increases (or decreases respectively). Hence, $Threat\_likelihood(s, r, o, c)$ increases (or decreases) as $csl(s)$ decreases (or increases respectively). We conclude that formula (3) satisfies **Principle 2**.
- If $csl(s) < col(o)$ and $Counter(s, r, o, c) < Intrinsic(s, r, o, c)$, when $Counter(s, r, o, c)$ increases (or decreases), $Threat\_likelihood(s, r, o, c)$ decreases (or increases respectively). We conclude that formula (3) satisfies **Principle 5**.
- When $Counter(s, r, o, c) \geq Intrinsic(s, r, o, c)$ then $Threat\_likelihood(s, r, o, c) = 0$. We conclude that formula (3) satisfies **Principle 6**.
- When $Intrinsic(s, r, o, c)$ increases (or decreases), $Threat\_likelihood(s, r, o, c)$ increases (or decreases respectively). We conclude that formula (3) satisfies **Principle 7**.
- The maximum value that could be obtained by this formula is equal to the maximum value of $Intrinsic(s, r, o, c)$ which cannot be greater than 1. The

minimum value that could be obtained by this formula can not be less than 0. Hence, the threat likelihood values of an access request are bound between 0 and 1. We conclude that formula (3) satisfies **Principle 8**.

3.3 Write threat likelihood assessment when secrecy is intended

We can quantify threat likelihood of excessive write accesses, when subjects secrecy levels are higher than objects secrecy levels, by adhering to **Principles 3**, **4**, **5**, **6**, **7** and **8**. For this, we propose the following two formulae which give respectively values representing the intrinsic object_based likelihood of threat on secrecy when write access is requested and the object_based likelihood of threat on secrecy when write access is requested , note the symmetry with respect to formulae (2) and (3):

$$Intrinsic(s,w,o,c) = \begin{cases} \frac{(|L_c| \times \widehat{col(o)} + \widehat{csl(s)})}{(|L_c|)^2 - 1}, \textbf{if } csl(s) > col(o) \\ 0, \quad \textbf{Otherwise}. \end{cases} \tag{4}$$

$$Threat\_likelihood(s,w,o,c) = \begin{cases} Intrinsic(s,w,o,c) - Counter(s,w,o,c), \\ \textbf{if } csl(s) > col(o) \textit{ and} \\ Counter(s,w,o,c) < Intrinsic(s,w,o,c) \\ 0, \quad \textbf{Otherwise}. \end{cases} \tag{5}$$

4 Threat likelihood assessment when integrity is intended

In the previous sections, we have presented an approach for assessing the likelihood of the threat on secrecy. This approach was based on the idea that this likelihood increases when information flows down. In this section we will briefly show how to apply the same concepts to the security criteria of integrity. Our approach is based on the idea that the threat on integrity increases when information flows up. This idea is at the foundation of the well-known Biba model [20]. In the following, we present definitions and principles to be used for the assessment of the likelihoods of threats on integrity. The following functions are needed to develop our approach:

- $isl : S \rightarrow L_i$ formally represents the assignment of integrity levels to subjects that reflects the trust related to data integrity bestowed upon each of them by the organization that owns the data.
- $iol : O \rightarrow L_i$ formally represents the assignment of integrity levels to objects that reflects the data integrity protection needs for each of them.

Clearly, the properties of integrity are dual with respect to the properties of secrecy. We leave it to the reader to modify the definitions and examples in section 3.1 in order to apply them to integrity. Similar formulas can be derived,

with the *sc* parameter set to i. Hence, in this section, we only introduce the concept of threat likelihood indexing when integrity is intended. We assign a numerical value from the set $\{0, \cdots, |L_i| - 1\}$ that represents the threat likelihood index of an integrity level *ilevel* in L_i. For example, in the case of write accesses when integrity is intended, from the point of view of subjects, we expect the threat likelihood to increase as subject integrity levels decrease. Hence, subject threat likelihood index values decrease with subject integrity levels. We write $\widehat{ilevel}$ to denote an entity (subject or object) threat likelihood index that decreases with the entity integrity level. Formally, $\widehat{ilevel} = |L_i| - ilevel$. However, from the point of view of objects, we expect threat likelihood to increase as object integrity levels increase. Hence, object threat likelihood indexes increase with object integrity levels. We write $\overbrace{ilevel}$ to denote an entity threat likelihood index that increases with entity integrity levels. Formally, $\overbrace{ilevel} = ilevel$ - 1.

The following two formulae give respectively values representing the intrinsic object_based likelihood of threat on integrity when read access is requested and the object_based likelihood of threat on integrity when read access is requested.

$$Intrinsic(s, r, o, i) = \begin{cases} \frac{(|L_i| \times \widehat{iol(o)} + \overbrace{isl(s)})}{(|L_i|)^2 - 1}, \textbf{if } isl(s) > icol(o) \\ 0, \quad \textbf{Otherwise}. \end{cases} \tag{6}$$

$$Threat\_likelihood(s, r, o, i) = \begin{cases} Intrinsic(s, r, o, i) - Counter(s, r, o, i), \\ \textbf{if } isl(s) > iol(o) \textit{ and} \\ Counter(s, r, o, i) < Intrinsic(s, r, o, i) \\ 0, \quad \textbf{Otherwise}. \end{cases} \tag{7}$$

We can derive two formulae giving respectively values representing the intrinsic object_based likelihood of threat on integrity when write access is requested and the object_based likelihood of threat on integrity when write access is requested. Note the symmetry with respect to formulae (6) and (7).

$$Intrinsic(s, w, o, i) = \begin{cases} \frac{(|L_i| \times \overbrace{iol(o)} + \widehat{isl(s)})}{(|L_i|)^2 - 1}, \textbf{if } isl(s) < iol(o) \\ 0, \quad \textbf{Otherwise}. \end{cases} \tag{8}$$

$$Threat\_likelihood(s, w, o, i) = \begin{cases} Intrinsic(s, w, o, i) - Counter(s, w, o, i), \\ \textbf{if } isl(s) < iol(o) \textit{ and} \\ Counter(s, w, o, i) < Intrinsic(s, w, o, i) \\ 0, \quad \textbf{Otherwise}. \end{cases} \tag{9}$$

5 Related work and limitations

In our previous work [16, 15], we present a framework for threat likelihood and risk assessment, which includes four different approaches. In this work, our threat

likelihood assessment approach is information flow based, makes a distinction between read and write accesses, takes into account security countermeasures and gives different estimates based on the intended security criteria (secrecy or integrity).

Fagade et al. explore the behavioural dimension of compliance to information security standards. Based on an established model of Information Security Governance Framework, they propose how information security may be embedded into organisation security culture [8]. Caputo et al. provide a prototype system for identifying insider threats. This research experimentally studies how malicious insiders behave and how they use information differently from a benign baseline group [4]. Greitzer et al. describe a predictive modeling framework that integrates a set of data sources from the cyber domain, as well as inferred psychological factors that may underlie malicious insider exploits. This threat assessment approach provides automated support for the detection of high-risk behaviors [12]. Hua et al. propose a game theoretical model to study the economic impact of insider threats on information systems security investments. They identify three factors influencing the optimal information systems security investment: breach function sensitivity, deterrence level, and advantage rate. They show that the optimal investment required to protect an information systems infrastructure from insiders is higher than for protecting against external threats [13]. Compared to our approach, none of these works present a qualitative or a quantitative method, to assess insider threat likelihood for access control systems.

Cheng et al. propose Fuzzy Multi-Level Security (Fuzzy MLS), which quantifies the risk of an access request in multi-level security systems as a product of the value of information and probability of unauthorized disclosure [5]. The Fuzzy MLS thesis considers that all subject-object accesses include a temptation to leak information and aims to quantify the risk of "unauthorized disclosure" of information by subjects. In comparison with Fuzzy MLS, the aim of our framework is to assess the threat likelihood posed by subjects towards objects by referring to object sensitivity and subject trustworthiness levels. Unlike Fuzzy MLS which is limited to the estimation of the threat likelihood of read accesses forbidden by Bell Lapadula, our approach estimates the threat likelihood of read and write accesses, is applicable when the objective of integrity is of interest (is not limited to secrecy) and considers security countermeasures mitigating the threat likelihood.

Bartsch proposes a policy override calculus for qualitative risk assessment in the context of role-based access control systems [20]. The risk is equal to the highest value from values estimated for each security objective (secrecy, integrity and availability). This work presents a qualitative estimation of threat likelihood. In comparison with the work of Bartsch, our approach is both qualitative and quantitative, developed in the context of generic access control systems by referring to the sensitivity of objects and trustworthiness of subjects and is not limited to RBAC.

Diep et al. describe an access control model with context-based decisions

that includes quantitative risk assessment [7]. However, they do not provide a method for estimating threat likelihood measures.

Wang and Jin propose a method to quantify access risk by considering need-to-know requirements for privacy protection within the context of health information systems [23]. This work exploits the concept of entropy from information theory to compute risk scores of access requests. We believe that our framework could be extended to consider need-to-know requirements while assessing threats of subject-object accesses.

Kandala et al. develop a framework that captures various components and their interactions in order to develop "abstract models" for RAdAC [17]. However, this work does not consider concrete details of assessing threat or risk.

Threat likelihood assessment in our framework cannot cover unexpected threats such as those in which several other socio-technical parameters must be taken into consideration for reflecting the reality of internal threats such as users' access history, behavior, collusion with other users, etc. Hence, all these parameters are outside of the scope of this paper. Similarly, threats related to social engineering concerns and threats posed by Denial of service (Dos) attacks which might compromise the availability criterion by read and write operations, cannot be assessed by our approach.

6 Conclusion

The main contribution of this paper is a qualitative and quantitative approach for insider threat likelihood assessment in the context of access control systems. Our approach considers primarily the security levels of objects, hence giving more priority to the sensitivity of data. Our approach can easily accommodate other views, such as those presented in [16, 15]. In order to obtain realistic values of insider threat likelihood while being compliant with IT Risk standards and guidelines, our approach considers the effect of the security countermeasures mitigating the threat likelihood of access requests.

We believe that this framework is important because it can be used to assess the likelihood of threats posed by subjects towards objects that will subsequently affect the computation of risk metrics. Note that our framework could be extended to also consider need-to-know requirements while assessing threats of subject-object accesses by considering categories of data.

In this paper, we have presented a qualitative and a quantitative threat likelihood assessment approach, which is required for estimating access risks. Nonetheless, our objective is to develop an approach for estimating the risk of access requests by adopting the risk assessment formula (1). This requires us to extend the work reported in this paper by defining formulas for computing impact values in order to quantify the risk of access requests. Future papers will describe this extension.

6.1 Acknowledgements

This research was partially supported by the Natural Sciences and Engineering Research Council of Canada.

References

1. S. Bartsch. A calculus for the qualitative risk assessment of policy override authorization. In *Proceedings of the international conference on Security of information and networks*, pages 62–70, 2010.
2. M. Bishop and C. Gates. Defining the insider threat. In *Proceedings of the 4th annual workshop on Cyber security and information intelligence research: developing strategies to meet the cyber security and information intelligence challenges ahead*, page 15, 2008.
3. S. Boulares, K. Adi, and L. Logrippo. Insider Threat Likelihood Assessment for Access Control Systems: Quantitative Approach. In *International Symposium on Foundations and Practice of Security*, pages 135–142, 2016.
4. D. Caputo, M. Maloof, and G. Stephens. Detecting insider theft of trade secrets. *IEEE Security & Privacy*, 7(6):14–21, 2009.
5. P. Cheng, P. Rohatgi, C. Keser, P. Karger, G. Wagner, and A. Reninger. Fuzzy multilevel security: An experiment on quantified risk-adaptive access control. In *Security and Privacy*, pages 222–230, 2007.
6. Clusif. *MEHARI 2010 principes fondamentaux et spécifications fonctionnelles*. 2009.
7. N. Diep, L. Hung, Y. Zhung, S. Lee, Y. Lee, and H. Lee. Enforcing access control using risk assessment. In *Universal Multiservice Networks. Fourth European Conference on*, pages 419–424, 2007.
8. T. Fagade and T. Tryfonas. Security by Compliance? A Study of Insider Threat Implications for Nigerian Banks. In *International Conference on Human Aspects of Information Security, Privacy, and Trust*, pages 128–139, 2016.
9. International Organization for Standardization. *ISO/IEC 27001: Information technology – Security techniques – Information security management systems – Requirements*. 2013.
10. IT Global Corporate. Security risks (2013), 2013.
11. INFOSEC Glossary. National information systems security (infosec) glossary. 2000.
12. F. Greitzer and R. Hohimer. Modeling human behavior to anticipate insider attacks. *Journal of Strategic Security*, 4(2):25, 2011.
13. J. Hua and S. Bapna. Who can we trust?: The economic impact of insider threats. *Journal of Global Information Technology Management*, 16(4):47–67, 2013.
14. S. Kandala, R. Sandhu, and V. Bhamidipati. An attribute based framework for risk-adaptive access control models. In *Availability, Reliability and Security*, pages 236–241, 2011.
15. H. Khambhammettu, S. Boulares, K. Adi, and L. Logrippo. A framework for threat assessment in access control systems. In *Information Security and Privacy Research*, pages 187–198. 2012.
16. H. Khambhammettu, S. Boulares, K. Adi, and L. Logrippo. A framework for risk assessment in access control systems. *Computers & Security*, 39:86–103, 2013.
17. R. McGraw. Risk-adaptable access control (radac). In *Privilege (Access) Management Workshop. National Institute of Standards and Technology*, 2009.

18. M. Meucci and A. Muller. The OWASP testing guide 4.0. *Open Web Application Security Project*, page 30, 2014.
19. NIST. Risk management guide for information technology systems. 2002.
20. R. Sandhu. Lattice-based access control models. *Computer*, 26(11):9–19, 1993.
21. H. Shey, K. Mak, S. Balaouras, and B. Luu. Understand the state of data security and privacy: 2013 to 2014. *Forrester Research*, 10, 2013.
22. G. Stoneburner, A. Goguen, and A. Feringa. Risk management guide for information technology systems. 2002.
23. Q. Wang and H. Jin. Quantified risk-adaptive access control for patient privacy protection in health information systems. In *Proceedings of the ACM Symposium on Information, Computer and Communications Security*, pages 406–410, 2011.
24. C. Weissman. Security controls in the adept-50 time-sharing system. In *Proceedings of the fall joint computer conference*, pages 119–133, 1969.