How to Evaluate Trust Using MMT *

Khalifa Toumi[1], Wissam Mallouli[2],Edgardo Montes de Oca[2], César Andrés[1], Ana Cavalli[1]

[1] IT/ TELECOM & Management SudParis, EVRY, F-91011
{Khalifa.Toumi,César Andrés, Ana.Cavalli}@it-sudparis.eu
[2] Montimage, 39 rue Bobillot, 75013 Paris - France
{wissam.mallouli,edgardo.montesdeoca}@montimage.com

Abstract. Trust evaluation is becoming a more and more active and critical area mainly for guaranteeing secure interoperation between communicating systems. One of the basic parameters used to evaluate the trust in a remote entity (user or system) is the previous experience, i.e. the interactions already performed between the truster and the trustee. However the monitoring of the trustee behavior and the analysis of the collected data and events are not an easy task. First of all, we need to define relevant patterns that describe the desired behaviors to be monitored and check them using a dedicated tool.
Within this paper, we extended an open source tool (MMT) to monitor users' behavior and define behavior patterns using temporal properties. We also design some evaluation strategies and illustrate the whole approach by the application to a real case study related to a collaborative programming project.

Keywords: Trust, Multi-Organization Environment, User experience, Monitoring and events correlation.

1 Introduction

Collaboration between public/private organizations like companies, universities, banks and hospitals spread more and more rapidly and usually shapes according to different partnership strategies in a Multi-Organization Environement (MOE). This has many advantages such as: (1) the ability to use remote and professional resources, services and knowledge, (2) the reducing of intervention duration and (3) the gain of experts skills and experience.

MOE is a paradigm that contains at least two organizations: an O-grantor and/or an O-grantee. The O-grantor is the participant that offers resources. These resources are acceded by users of another organization called the O-grantee. The resource sharing task is based on some restriction rules that constitute an *interoperability security policy*, and it allows to control the access to these resources.

Many works in the literature [2,4,7] focus on the trust and security challenges in distributed environment. In order to define a *trust level*, there are several different criteria that should be analyzed and evaluated. One of the most important criteria is the previous experience of a trustee.

"Experience is the teacher of all things." (Julius Caesar)

* The research work presented in this paper is supported by the European project Inter-Trust.

Different approaches have been proposed to evaluate this criteria based on the assessment of the historical interactions with the trustee. However, different challenges are still open. For instance, some assumptions related to the trustee behavior monitoring are considered in several works. Besides, no work proposes a detailed discussion "*How to monitor the trustee behavior*?" [2,5,7,6]. They generally assume that the monitoring is possible and that the different parameters (related to the trustee experience) are available as input which is not the case in real case studies. In order to target this issue, we propose in this paper:

- *A methodology to assess different interactions between at least two entities.*
- *An extension of a monitoring tool called MMT for the evaluation of a behavior based on the trust needs.*

Our approach is an extension of our previous work [7,6] to implement a prototype solution for trust evaluation framework. To reach this objective, we have adapted the MMT monitoring tool that allows to a real-time visibility of network traffic. Indeed, the paper approach consists of the following steps: (1) a new plug-in called 'trust-plug' is developed to analyze the interaction traces and to extract the different attributes that are relevant or can have an impact on the trust; (2)The formalism allowing the specification of MMT properties (denoting trust patterns) is extended, (3)MMT tool is also extended by adding periodical trace checking and a trust level notification has been also developed.

The rest of the paper is structured as follows. In Section 2, we define the trust parameters in MOE. Section 3 shows how to evaluate the satisfactory function and the possible strategies to evaluate a trustee. Afterwards, a case study is detailed in Section 4 in order to demonstrate the efficiency of the proposed approach. Finally, Section 5 concludes the paper and presents ideas for future work.

2 Trust definition

Currently, there are several definitions of the concept of trust in the literature. We will use an adaptation of the one presented in [1]: Trust is not an objective property of an entity but a subjective degree of belief about an organization or a user. In this context, the trust is a relationship between a **truster** and a **trustee** related to a **situation** at a given **time**.

Definition 1. The truster is any organization that offers an access to a specific resource. Any *O-grantor* in `MOE` will be a truster. □

Definition 2. The trustee can be an organization or a user that needs a service. □

Definition 3. A situation is composed by an **activity** and a **view**. An activity is an action to be performed and a view is a set of objects that may be accessed by the user. □

Definition 4. The time in `MOE` will be represented by intervals. □

Definition 5. The degree of belief, also known as *the trust value* is used to measure the belief between two entities (the **truster** and the **trustee**). Its value allows us to determine if we can trust or not the trustee (related to a **situation** and a **time**). □

In order to define the `belief` function in this framework we were based on the experience parameter.

Definition 6. Experience learning aims to establish wisdom on making decision. It is based on the evaluation of the previous interactions between the **trustee** and the **truster** related to a specific **situation** at a **period of time**. □

There are considered two types of experiences.

- The experience of the trustee organization that takes into consideration the previous *behaviors* of all users of this organization,
- and the direct experience where only the previous *behaviors* between this user and the truster are considered.

The evaluations of these parameters depend on a satisfactory function that is used in order to evaluate an interaction. We assume that any interaction can be valuated as a *satisfactory* or *unsatisfactory* behavior.

If the valuation is unsatisfactory then it is considered as a *bad behavior*, that is, it will decrease the experience evaluation of the trustee. On the contrary, if the valuation is satisfactory it will increase the experience evaluation. We note that the output of this function is a value in [-1,1] assigned to the interaction. In this paper we present an approach of how to evaluate this function denoted `sat`.

Evaluation of the user experience:

For any $u \in \texttt{Subjects}$, $s \in \texttt{Situations}$, $\hat{T}_i \in \mathcal{I}_{\mathbb{R}_+}$,

we define the experience evaluation function with respect to $\texttt{org}_A$ as:

$$\texttt{eX}_1(u, \texttt{org}_A, \hat{T}_n, s, l) = \frac{\sum_{i=0}^{n} \frac{\sum_{b \in l_i} sat(b)}{|l_i|}}{n}$$

where l_i contains the set of behaviors u that were performed before and during $\hat{T}_i$ related to the situation s and sat(b) is the function that will evaluate a behavior b. This function will be more detailed in Section 3.

Evaluation of the organization experience:

For any $\texttt{org}_B \in \texttt{Organizations}$, $s \in \texttt{Situations}$, $\hat{T}_i \in \mathcal{I}_{\mathbb{R}_+}$, and for any not empty log l, we define the experience evaluation function of an organization $\texttt{org}_\texttt{B}$ with respect to $\texttt{org}_\texttt{A}$ as:

$$\texttt{eX}_2(\texttt{org}_B, \texttt{org}_A, \hat{T}_n, s, l) = \frac{\sum_{u \in \texttt{employee}(\texttt{org}_B, \texttt{org}_A)} \texttt{eX}_1(u, \texttt{org}_A, \hat{T}_n, s, l)}{|\texttt{employee}(\texttt{org}_B, \texttt{org}_A)|}$$

where $\texttt{employee}(\texttt{org}_B, \texttt{org}_A)$ are the set of employees of $\texttt{org}_B$ that have collaborated with $\texttt{org}_A$.

3 Satisfactory evaluation

In this section, we propose a satisfactory evaluation method that aims to assess a behavior b in MOE. This assessment associates a value between [-1,1] to this behavior, it will be denoted $\texttt{Sat}(\texttt{b})$. If $\texttt{Sat}(\texttt{b}) \in [0, 1]$ then this means that the previous behavior does not respect some requirements and it is considered as a bad one that have to decrease the experience evaluation of the user. Otherwise b is considered as a good behavior that will increase the experience evaluation of the user.

3.1 Evaluation of the satisfactory function

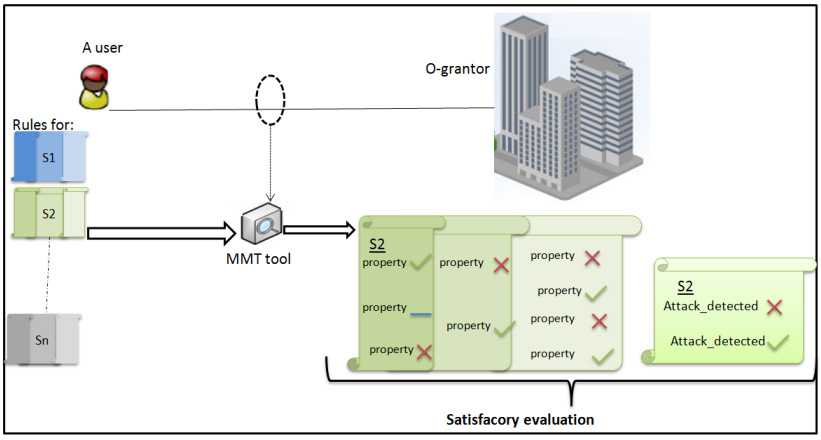

Fig. 1: Evaluation of an interaction.

Regarding the definition of satisfactory function, in this section we will present how to apply the definition and how to evaluate it. We will introduce it using the example presented in Figure 1.

As it is shown in Figure 1, for each *situation* we will have a list of *rules* that can be a security property to respect or an attack to detect. These *rules* will be written based on an extension of MMT language. Any *interaction* of a user will belong to one situation.

To evaluate it, we have:

1. To select the list of rules related to a fixed situation. In the running example, in Figure 1, we consider that the interaction of the user is related to S2.
2. To apply these rules as inputs of MMT tool. Then, it automatically computes which rules are respected, which rules are disrespected, and which are the different violations during the interaction.

In this proposal, **the influence of the different properties are not the same**. As it is shown in Figure 2, we provide three partitions of the different properties (high , medium and low) in order to differ between the list of properties. We will say that *in the case of a "security property", a high (resp. medium) security rule is more important than a medium (resp. low) rule.*

Based on the MMT tool and our new plug-in "Trust-plug", that analyzes the trace for trust proposals we are allowed to check that:

- If the rule is respected, then a value `+1` is assigned to it.
- If the rule is not respected, then a value `-1` is assigned to it.
- If we cannot have a decision about this rule during the interaction, then a value `0` is assigned to it.

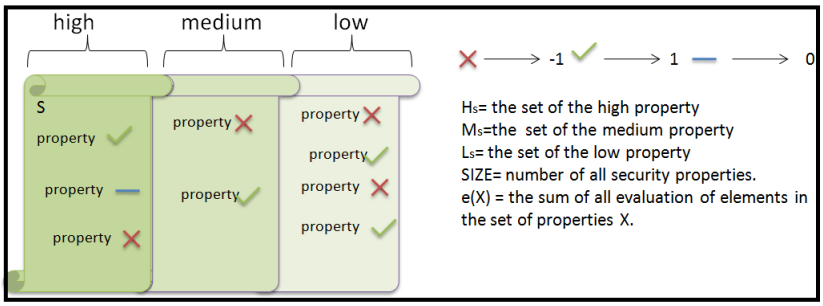

Fig. 2: Properties partition.

Evaluation1: Based on the assigned values, we define the satisfactory function for a situation s as:

$$\begin{cases} -1 & if\ an\ attack\ is\ detected \\ \frac{e(H_s)+e(M_s)/2+e(L_s)/4}{SIZE} & otherwise \end{cases}$$

where H_s, M_s and L_s are the set of high, medium and low security properties defined for a situation s, $SIZE$ is the total number of rules for this situation, and e is a function that takes as inputs a set of properties and give as output the sum of their verdict.

4 Case study

In this section we will present a case study, where we can show the usability of our solution.

4.1 Scenario

We will consider the following MOE scenario:

- Four organizations are participating in the same development of a project.
- The first organization $\texttt{org}_\texttt{A}$ has a server where several virtual machines are offered.
- These are the considered **activities**: configure, modify, execute, test, and manage.
- The organization org_A also offers the following **views**: source_code, application, testing_script, OS_System and resources.
- In this scenario four different **external** roles are defined: engineer, researcher, tester, and project manager.

4.2 Specification of the Interoperability Security Policy

The first phase is the specification and the deployment of an interoperability security policy. This policy is the result of a negotiation process between the O-grantor and the O-grantee. An example of how to do it is detailed in [3]. We show on the following a part of the $\texttt{org}_\texttt{A}$ interoperability security policy.

- R1: An engineer is permitted to manage OS_System.
- R2: A researcher is prohibited to manage resources.

– R3: An engineer is permitted to modify a source_code
– R4: An engineer is permitted to execute an application.
– This rules are only applied for the **external** engineers and researchers that does not belong to $\texttt{org}_A$.

Related to this rule we have this trust policy:

– R1 is activated only if the trust evaluation of the user is more than 0.4 **and** the trust evaluation of the organization is more than 0.
– R2 is activated only if the trust evaluation of the user is less than 0 **or** (if the trust evaluation of the organization is between -0.3 and 0.2 **and** the trust evaluation of the user is between 0 and 0.4)
– R3 is activated only if the trust evaluation of the user is more than 0.5 **and** the trust evaluation of the organization is more than 0.7.
– R4 is activated only if the trust evaluation of the user is more than 0.5.

4.3 Trust properties definition

The second step is to define a list of properties and threats that permits to evaluate the different interaction with $\texttt{org}_A$. For each situation, we have to write a list of properties.

Example 1. Figure 3 shows the trust properties for the situation s_1 `manage▷OS_System`. The Figure 4 shows how to write the the property p1 in our new extension of the MMT

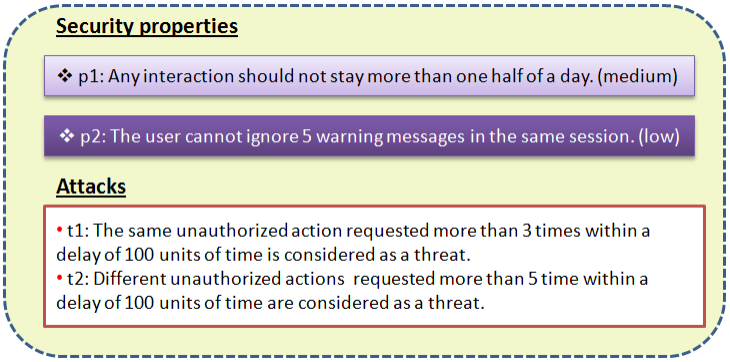

Fig. 3: Trust properties for `manage▷OS_System`

language.

We add firstly a new parameter 'partition' for the tag <property> in order to precise the importance of the rule. Moreover, as it is shown in Figure 4, a trust plug-in is developed that has to analyze the xml trace and to extract several elements as the **external** role of the user, its organization and the type of the request. □

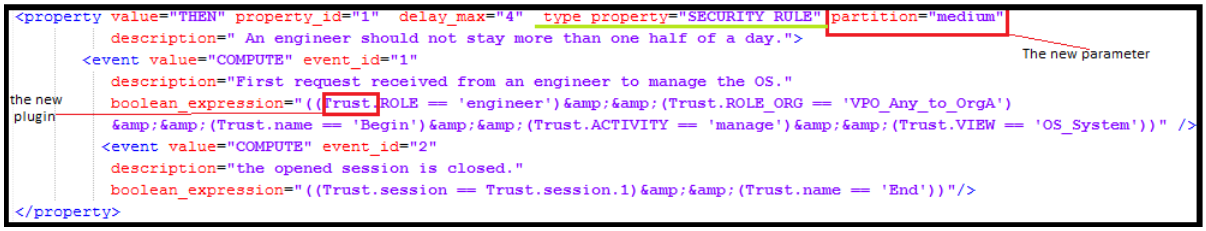

Fig. 4: A security property for the situation manage OS_System

Security rules summary results (period 4)

Id	Description	High	medium	low		
p1	SECURITY RULE: Any interaction should not stay more than one half of a day		medium		1	0
p2	SECURITY RULE:The user cannot ignore 5 warning messages in the same session		low		1	1

Attack summary results (period 4)

Id	Description		
t1	ATTACK: Unauthorized action requested more than 3 times within a delay of 100 units of time.	0	0
t2	ATTACK: Different unauthorized actions requested more than 5 time within a delay of 100 units of time .	0	0

Satisfactory evaluation (period 4)

Req_Id	user	Organization	Situation	TimeStamp	Sat
2117	a1	OrgA	manage OS_System	2013-06-04 16:34:12.000000	0.25

Fig. 5: Result file from MMT.

Therefore, the satisfactory function of any interaction related to the situation `manage▷OS_System` will be:

$$\begin{cases} -1 & if\ t1\ or\ t2\ are\ detected \\ \frac{e(\varnothing)+\frac{e(\{p1\})}{2}+e(\{p2\})/4}{4} & otherwise \end{cases}$$

4.4 Executing MMT with the previous rules

After each period, MMT provides a result file as it is shown in Figure 5. This file contains three tables:

- The first one cites the different properties, how many times that are respected or disrespected, the partition of the property.
- The second table is about the detected attacks.

– Finally, the last one provides a table that show the interaction (request identity, the user, the organization, the situation and the timestamps) with its assigned satisfactory evaluation.

Based on these results, the trust level of the user and the organization $\texttt{org}_{\texttt{A}}$ will be updated in the configuration file. These results with the trust and the security policies will permit to give a response response for any request. For example, a request that will be received during the period 5 to manage OS_System will be accepted since:

– A permission rule (R1) is provided for this user (see subsection 4.2).
– R1 is activated since the trust evaluation of the user is equal to 0.5 more than 0.4 and the trust evaluation of the organization is equal to 0.1 more than 0.

This approach offers to an access control system to take into consideration the new interactions between the trustee and the truster. This permits to react by giving new permissions to unauthorized employee, to refuse an access for an authorized user in the previous period and to have a dynamic policy based on the analysis of the requester behaviors.

5 Conclusions and Future Work

In this paper, we present a methodology that permits to evaluate an interaction between a trustee and a truster. An extension of the monitoring tool MMT is proposed. Moreover, the basic function 'satisfactory evaluation' that permits to assess an interaction is well detailed. Finally, the different steps of how to do with a case study is presented.

As future work, we are planning to use our approach in other distributed system as the VANET networks and e-Voting system for the European project Inter-Trust and we aim also to integrate a new parameter 'reputation': its definition, evaluation and its spread between the different entities will be our interest on the future.

References

1. A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *33rd annual Hawaii International Conference on System Sciences, HICSS'00.*, volume 11, janaury 2000.
2. S. Chakraborty and I. Ray. TrustBAC: integrating trust relationships into the RBAC model for access control in open systems. In *ACM Symposium on Access Control Models And Technologies, SACMAT'06*, pages 49 – 58. ACM, 2006.
3. F. Cuppens, N. Cuppens-Boulahia, and C. Coma. O2O: Virtual private organizations to manage security policy interoperability. In *2nd Int. Conf. on Information Systems Security, ICISS'06, LNCS 4332*, pages 101–115. Springer, 2006.
4. D. Abi Haidar, N. Cuppens-Boulahia, F. Cuppens, and H. Debar. XeNA: an access negotiation framework using XACML. *Annals of telecommunications*, 64(1-2):155– 169, 2009.
5. Indrakshi Ray, Indrajit Ray, and S. Chakraborty. An interoperable context sensitive model of trust. *Journal of Intelligent Information Systems*, 32(1):75–104, 2009.
6. K. Toumi, Cesar Andres, and A. Cavalli. Trust-orbac: A trust access control model in multi-organization environments. In *International Conference on Information Systems Security, ICISS'12, LNCS*. Springer, 2012.
7. K. Toumi, Cesar Andres, A. Cavalli, and Mazen EL Maarabani. A vector based model approach for defining trust in multi-organization environments. In *International Conference on Risks and Security of Internet and Systems, CRISIS'12*. IEEE Computer Society Press, 2012.