Enhancing Access-Control with Risk-Metrics for Collaboration on Social Cloud-Platforms

Ahmed Bouchami, Elio Goettelmann, Olivier Perrin, Claude Godart

Enhancing Access-Control with Risk-Metrics For Collaboration On Social Cloud-Platforms

Ahmed Bouchami*, Elio Goettelmann*†, Olivier Perrin* and Claude Godart*

*LORIA - INRIA Grand Est
Université de Lorraine, Nancy, France
{*ahmed.bouchami, olivier.perrin, claude.godart*}*@loria.fr*

†Luxembourg Institute of Science and Technology
L-4362 Esch-sur-Alzette, Luxembourg
elio.goettelmann@list.lu

***Abstract*—Cloud computing promotes the exchange of information, resources and tasks between different organizations by facilitating the deployment and adoption of centralized collaboration platforms: Professional Social Networking (PSN). However, issues concerning security management are preventing their widespread use, as organizations still need to protect some of their sensitive data. Traditional access control policies, defined over the triplet (*User, Action, Resource*) are difficult to put in place in such highly dynamic environments. In this paper, we introduce risk metrics in existing access control systems to combine the fine-grained policies defined at the user level, with a global risk-policy defined at the organization's level. Experiments show the impact of our approach when deployed on traditional systems.**

I. Introduction

Cloud environments change the behaviors of companies, not only by transforming their use of IT through a different cost approach, but also through promoting new kinds of usages. Collaboration with third parties through dedicated platforms is one example for this: GitHub[1], Dropbox[2] or Agora[3]. By creating and promoting the exchange of information, resource or even tasks, the company becomes more flexible and can adapt rapidly to new requirements or situations. More specifically, these technologies are known as *Professional Social Networking (PSN)* [1], [2]. *A professional social network is a collaborative platform that offers several ways to professionals to collaborate by: exchanging information in an easy and practical way, working on common projects at the same time using shared data and/or tools (Platform as a Service), organizing meeting and sharing calendar (communication), etc*. In recent years, several professional social networks have emerged, e.g, Yammer [4], Tibco Software Tibbr [5] , OpenPaaS [6], eXo [7], the reader can find a good review in[8].

However, there are still remaining challenges concerning security management when companies decide to migrate their work environments to such solutions. Indeed, traditional access control systems may not be adapted for deployment on such platforms. On the one side, access control policies are too fine-grained for the company itself which has to apply it on all its users and resources. On the other side, the company cannot entirely rely on its users to define reliable and trusted policies. Malicious insiders or insufficient informed users can twist this system by defining erroneous policies. Moreover, the highly dynamic nature of such environments makes it difficult to apply static security policies on them. New users, new resources and new collaborations can be added and removed at a very high frequency, which is incoherent with the definition of fine-grained access control policies at the organization's level. Existing systems and solutions have to be adapted to these new requirements.

In this sense, we propose to enhance existing access control systems with a security risk approach. By mixing risk metrics with the access control we help companies to deny access attempts when these are considered as not trusted enough: a risk level assessed for each incoming request compared to a given threshold to exclude too risky ones.

The paper is structured as follows. Section II begin by introducing the architecture of our framework. Then, it details the global problem, and by means of some motivating examples, justify the objectives behind the use of the risk concept. It further gives some background about access control in PSN and security risk management. Section III begins by explaining how we align the concepts of risk management with those of access control. Then, it formalizes the proposed approach. Afterward, section IV presents the experiments we conducted and discusses them. Finally, sections V and VI respectively position our contribution in the existing work and concludes the paper.

II. Architecture, motivations and background

In our approach we evaluate the risk of an access request and integrate it in the organization's global security policy. Our approach acts at real-time, which means that it prevents an access attempt if its risk value is too high.

A. Global architecture

Our vision of a professional social network consists in an intersection between several domains of different types of organizations, e.g: high school, enterprise, restaurant, etc. This intersection is defined by the *community* concept.Our definition of the community concept is quite similar to Wenger's definition of communities of practices CoPs [3]. *a community within a professional social network is a logical federation that encloses different entities, namely, users, resources and theirs*

[1]http://www.github.com
[2]http://www.dropbox.com
[3]http://www.agora-project.net
[4]https://www.yammer.com/
[5]http://www.tibbr.com
[6]https://research.linagora.com/display/openpaas/Open+PaaS+Overview
[7]http://www.exoplatform.com/company/en/resource-center
[8]http://www.capterra.com/social-networking-software/

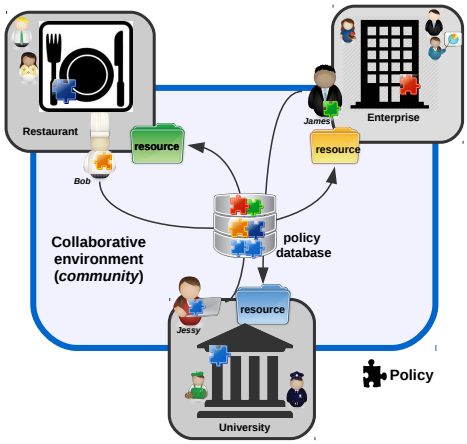

Fig. 1: Context overview

organizations. A community is created by any user having the required permissions (defined at the level of the social network). In this work, we assume that communities are fully independent for managing the access-control policies to the shared resources. In such a context, the authorization policies are valid just in the community in which they are defined. Inter-communities access-control policies mapping is out of the scope of this paper.

As depicted in Fig. 1, the user *James* which is an enterprise employee creates a community, then *James* invites other users to be part of his community (for a collaborative end): *Jessy* which is a university student and *Bob* which is a restaurant chef. Note that users could also send requests to belong to a given community. For instance, *Jessy* could send a request to join the community, and then the community creator *James* could accept or reject the demand.

Users from different organizations, can share resource(s) they own (through their organizations) within the community. When a user shares a given resource with another user, an access control rule is created and stored in the common community *policy database*. The rule is based on the triplet (*Subject, Object, Action*). In the architecture we propose, resources are still hosted by the organization. However, the access approval is made at the level of the central community decision point. Each user's policy is combined with his/here organization policy at the level of the community *policy database*. This allows organizations to still keep control over their resources. The policies definition and combination is based on our previous work [4].

To better understand our motivations, in the following, we introduce some examples. Then, we present some background to justify the mix of risk concept with the access control in our approach.

B. Motivations

In our opinion, the organization needs a high abstraction level for defining policies, due to the dynamic nature of the PSN in which users and/or resources tend to change frequently. Therefore, the main problematic can be summarized as follows: *how can we define a context-based global policy at the organization level, with a high abstraction level, and combinable with users level policies?*

In the context of an access control system, the risk can be generalized to the following statement: *a user gets unauthorized access to a resource of the system*. Indeed, even with an existing access control system, different reasons could lead to such an event, examples are:

- **Motivating Example 1**, *for this example, we show a normal case, where based on both, access control authorization and risk, a user will be granted.*
  e.g. The community *Com1* created by *James* uses the authentication mechanism *OAuth* [5]. Within *Com1* the student *Jessy* shares her *CV* and *marks* with the employee *James* for an internship application. *Jessy* gives to *James* the *Read* rights. *James* is a trusted person within *Com1*, and the shared resource *CV* is not confidential.
- **Motivating Example 2**, *the initial access control policies have been erroneously defined. In addition, imagine that an attacker impersonates a trustworthy user of the system or an existing user becomes malicious and tries to steal valuable information.*
  e.g. in *Com1*, *James* wants to share the daily lunch order for its enterprise with a restaurant chef. Then, *James* shares it by mistake with *Bob* which is not a chef. Basically, *Bob* in *Com1* is considered as a trusted person, however, over time his behavior becomes suspicious (possibly after an account hacking).
- **Motivating Example 3**, *The sensitivity of a resource changes over time, making the access control policy outdated.*
  e.g. *James* invites different software programmers to join *Com1* for working with *Jessy* for her internship. Among them, *Alice* who shares a *source code* document. Further, *Alice* notices that the project with *Jessy* becomes more and more confidential and thus has too many collaborators for working trustfully. Therefore, *Alice* removes some of the users from accessing the *source code* document.
- **Motivating Example 4**, *the authentication mechanism includes security flaws, so the user's identity cannot be guaranteed.*
  e.g. *James* creates another community *Com2* in which he invites *Carol* a travel agency employee. Then, *James* sends to *Carol* his desired trip period in the form of a professional traveling document, and *Carol* can modify this document by adding her proposed dates. Thus, *James* can choose one of the dates from this document which is considered as a more or less confidential resource. The authentication mechanism used for *Com2* is Login/Password which is considered as vulnerable.

As we said above, collaborative environments are quite dynamic, which means that we have to manage a high level of uncertainty regarding user's behaviors and context attributes changing states depending on real-time circumstances. With such issues, classical role based access control models [6],

[7], [8], [9] could not be efficient, because, additional attributes must be considered, like users behavior, context vulnerabilities and resources properties.

To deal with this challenges, we formalize this problem as a *risk problem*. Indeed, the *risk* definition allows us to consider the basic access control triplet (*Subject, Object, Action*) and the dynamicity and uncertainty of the collaborative context. The novelty and strength of our proposal is that we do not aim to replace the classical access control policies models (like *RBAC*, and others), but to adapt them to collaborative environments while keeping them unchanged. Moreover, the deactivation of the *risk* assessment module will not affect the decisions of the basic access-control model implemented in the environment in question. Based on our formalization of the *risk* assessment mechanism, and how we combine it with the access-control mechanism, settings for possible deactivation is a simple task.

C. Background

In this section we present some basic concepts related to the risk and access control management.

Access control in PSN

A professional social network is a *federated* [10] environment mainly composed by organizations. *An organization is essentially an umbrella for all types of business and social entities with multiple members in pursuit of a common goal*[9]. In a federated environment, the embodied organizations where they trust each other concerning information exchange, i.e, form a *circle of trust*. Nevertheless, the need of protecting the resources remains very important, due to the possible fraud of some untrustworthy partners, even if the organizations belong to a common *circle of trust*. This protection can be ensured by a strong authorization policy defined at the organization's level. However, in a dynamic and highly interactive environment, a serious issue about the definition of such policies is intuitively posed.

One of the main advantages of social networking consists in promoting users autonomy, by giving them the possibility to share resources with other users within the social network. For securely sharing resources, the user defines basic policy rules, i.e, decides which *subject* can perform which *action* on which *resource*. Conventionally access control policies can be based on: *users*' attributes (ID, role, name,...), *resources*' attributes, and desired *actions* on resources (i.e, Subject/Object/Action). However, we believe that for defining a reliable access control policy, the context's attributes consideration [11], [12], [13], [14] is as important as the classical triplet S/O/A. Therefore, in order to keep and take advantage of the users' autonomy (offered by the social network), and take into account the context, we consider that we have a two level policy: the user level and the organization level. At the user level a simple S/O/A policy rule is defined, while at the organization level (collaboration context), additional access control conditions can be considered.

[9]http://smallbusiness.chron.com/organization-vs-enterprise-68475.html

Risk management

Generally, a *risk* is defined by the *probability* that an *event* occurs and by its *consequences* [15]. In the IT security context, where IT components (ex. hardware, network, etc.) support business assets (ex. information, processes, etc.), the *security risk* is defined in a more fine-grained fashion. The event is usually seen as a *threat* which uses one or more *vulnerabilities* of the IT environment in order to create a negative *impact* (ex. destruction, alteration, theft, etc.) on the business assets [16]. For instance, an attacker steals sensitive information (i.e., threat) through a compromised interface (i.e., vulnerability) which leads to the business reputation loss (i.e., impact).

In this sense, a *security risk assessment* consists usually in evaluating the following formula ([17], [15]): *risk* = *vulnerability* $\times$ *threat* $\times$ *impact* $\Longleftrightarrow f(V, T, I)$. The goal is to estimate security risks in a quantitative and/or qualitative manner, to select those that need to be reduced and to develop countermeasures. Developing countermeasures involves the implementation of *security controls* by constraining technical solutions and by reducing vulnerabilities on the business settings. Security controls are management, operational, and technical safeguards prescribed for an information system to protect the confidentiality, integrity, and availability of the system and its information. Examples are *firewalls* or *intrusion detection systems*.

III. Proposal

To evaluate the risk of each access request, we have to align the concepts of risk with those of access control systems.

A. Concept alignment

Basically, our risk value will help the system to evaluate if the existing access control policy is adapted for the incoming request. If the risk value is too high (i.e. it exceeds a given threshold), the request is rejected, even if the initial policy would have granted the access. According to the previous section, to assess this risk value we have to evaluate the *vulnerabilities*, the *threat* and the *impact* of each request.

The *impact* of an unauthorized access is heavily dependent on the resource being accessed. As some resources are more important or sensitive than others, they should be more protected. Moreover, some actions on a resource can also have more serious consequences than others. A very confidential information should not be easily read, however if it can be easily re-generated an unauthorized deletion would not be a big issue. In the same sense, a public report should neither be modified nor deleted by an unauthorized access, but reading it should be of no problem. Therefore, in our approach we propose to assess the *impact* of the risk by evaluating the importance of the *resource* and the consequences that the request's *action* could have on this resource.

The *vulnerabilities* leading to an unauthorized access are mainly due to an *authentication mechanism* that can be tricked. Indeed, if the identity of the user making the request cannot be guaranteed, there is a risk that the user is an usurper. Basically, some authentication mechanisms are more secure than others. A two steps identification[10] for example, is more difficult to

[10]https://www.google.com/landing/2step/

trick than a simple password login. But imposing a biometric authentication to all users does not really make sense. Another approach could be to evaluate the vulnerabilities of the collaborative platform. Some platforms natively generate unauthorized requests (and handle them with exceptions). Thus, in some environments it is more "common" to reject lots of requests.

Concerning the *threat*, it emanates directly from the *user*, since it is his/her action (the fulfillment of his/her request) that can generate the event and its consequences. Therefore, to evaluate the probability of an attack (but also of an unintentional adverse action), we propose to evaluate the trustworthiness of a user. For example, users who often try to access unauthorized resources, should not be as easily trusted than users who have an exemplary behavior. If the second one makes a suspicious action, it will be more lesser risky to accept it than if the first one does it.

The main challenge of this work, is to find an optimal way to define the policies at the level of organizations in an abstract manner. Because, we consider that the PSN is very dynamic in term of adding/suspending users/resources/communities, and organizations can not have a real-time knowledge about the content of all the possible communities its users are involved. Therefore, we propose to evaluate the risk of the incoming requests, and each organization defines its risk-based threshold for requests' acceptance rate. Therefore, organizations keep control over their resources even through an external (trusted) access control point.

B. Formal framework

Now, we will formalize the above mentioned concepts namely *vulnerability*, *threat* and *impact* and show how to evaluate the *risk* based on the attributes of users, resources and context.

Definition 1 (**Impact**). *Depends on the action being requested on a given resource. More the resource is important more the impact will be high. An important resource is a resource on which little access permissions are defined. Accordingly, for a given request req that implies user u, resource r and action a, we compute the average of the policy responses for requesting action a on the resource r regarding to all the users of the community. For this purpose, we check how much permission(s) are given to perform the action a on the resource r within the community:*

$$Impact(a, r) = 1 - \frac{\sum_{u \in User} Policy(u, a, r)}{Card(User)} \quad (1)$$

while

$User$ *is the set of users of the community,*
$a \in Action$, *the set of actions (i.e. {R, W, X}),*
$r \in Resource$, *the set of resources of the community,*
$Policy$, *the policy-decision (*$accept$*=1,* $reject$*=0),*
$Card(User)$, *the number of users of the community.*

Note that we take the opposite of the average, as the impact decreases when more users have the access right for the considered resource.

Definition 2 (**Vulnerability**). *Depends on the strength of the authentication mechanisms implemented on top of the community. In the context of this work we consider in an ascending strength order, the following authentication mechanisms:* $Auth$ = {*Guest, PIN, Login/password, OAuth, 2 Step Validation, Biometric*}. *Therefore we give to each of them a score that represents the strength level of the authentication mechanism. Thus for a given request* req

$$Vulnerability(req) = V(C) = Score(A_C) \quad (2)$$

while

C, *the community of the request* req,
A_C, *the authentication mechanism of* C,
$Score : Auth \rightarrow [0, 1]$, *the strength level of* A_C

Notice that, the scoring is subjective and depends on the community creator, therefore, more than one authentication mechanism can have the same score. Thus, the function from the authentication mechanism set to the grade set is Surjective. An example is given in Table I.

As an extension, other parameters can be involved to measure the vulnerability depending on the context. For instance if in the community the delegation between users is enabled, therefore, the delegation depth could be an additional parameter to measure the vulnerability of the community.

Definition 3 (**Threat**). *Its probability depends on the trustworthiness of the user making the request. More the user u is trusted, less the threat will be probable. As the trust values are belonging to the interval $]0, 1[$, we interpret this by the formula:*

$$Threat(u) = 1 - Trust(u) \quad (3)$$

The trust computation is out of scope of this paper. However, as our framework is generic, several trust computation models [18], [19] can be implemented in the professional social network at the level of communities. For each community, the creator can integrate the trust model she/he prefers. The integration of a trust mechanism can be made through a Web-service call. The availability of the description (informations) about the trust mechanism depends on the will of the community creator.

Definition 4 (**Risk**). *In our context we decide to take a linear approach for calculating the risk value. However, in some contexts, it may be interesting to focus more on one value than on another. Thus, we introduce weightings into the risk-formula:*

$$Risk(req) = \frac{k_v \times V(C) + k_t \times T(u) + k_i \times I(a, r)}{k_v + k_t + k_i} \quad (4)$$

while

V, *the Vulnerability, and* k_v *its weighting,*
C, *the community of the request* req,
T, *the Threat, and* k_t *its weighting,*
u, *the user of the request* req,
I, *the Impact, and* k_i *its weighting,*
a, *the action of the request* req,

Basically the weightings can all be set to 1 to take equally into account all values (thus the risk corresponds to the average). However, some organization could put their focus more on the impact. The result of such a weighting is a plane, given in Fig. 2, where the weighting is equal to $\{3,1,1\}$ for respectively the *impact*, the *vulnerability* and the *threat*. We fixed the vulnerability value for this graph to 1 (the maximum level).

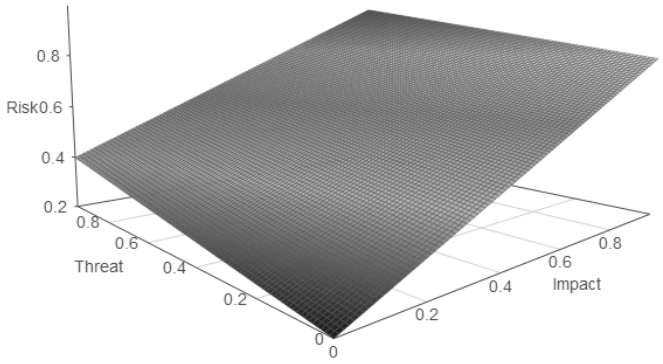

Fig. 2: Risk value for a vulnerability of 1 and a weighting of $\{3,1,1\}$

Threshold

Organizations still keep the control of the access to their resources (shared by users) by determining the maximum risk level they accept. Each organization is free to set (manually) a threshold value beyond which the incoming requests will be rejected. To define this threshold, the organization can either, opt for a cost mechanism like the one presented in [20], or, base it on the estimation of *damages* [21], [22].

In accordance to our objectives, our approach allows us to take into account additional information to evaluate the probability of:

- an impersonation as in *motivating-example 2*, through the *Threat* (the user's trust),
- the resource's importance as in *motivating-example 3*, through the *Impact*,
- a security flaw as in *motivating-example 4*, through the *Vulnerability*

IV. Experimentation

We conducted different experimentations to validate our approach. Due to a lack of data from real case scenarios, we simulated requests to cover a maximum range of possibilities. We calculated the risk value of each of these requests to see the influence of our approach on an existing system.

A. Implementation

First, we generate a random set of access control policies for fifty different users on fifty different resources. We selected the three commonly used actions *Read*, *Write* and *Execute* and distributed them respectively with following probabilities: $\{0.7, 0.3, 0.4\}$. Note that those probabilities do not impact the final result, but only allow to generate policies closer to a real case scenario. As we do not constrain our approach to any type of trust calculation, we assigned to each user a random trust level.

Second, we define six different authentication methods and assign a *vulnerability* level to each of them as defined in Definition 2. The different methods and their value can be seen in Table I. We have considered that each incoming request can be made by using one of those authentication mechanisms. We agree that assigning a vulnerability level of 0 to a Biometric authentication is controversial[11], but we did this for an illustration purpose, only in the context of our experimentations. It should not be understood otherwise.

TABLE I: Authentication methods and their vulnerability level

Name	Description	Vulnerability level
None	Not authenticated user	1.0
PIN	Pin Code	0.8
L+P	Login and Password	0.6
OAuth	OAuth service	0.4
2F	Two factors	0.2
Bio	Biometric	0.0

Finally, we generate randomly 1500 different access requests. Each request corresponds to a 4-tuple $\{$*User*, *Resource*, *Action*, *Authentication method*$\}$. The distribution is homogeneous with the exception of the *action*. The requested actions were generated to influence the policy-based rejection rate, basically we defined 4 scenarios: 10% of policy-based rejects (Fig. 3), 15% (Fig. 4), 20% (Fig. 5) and 25% (Fig. 6). This allows us to compare the behavior of our system and its influence in different scenarios. For each of these scenarios we also made 2 different risk calculations: the first one with a weighting of $\{1,1,1\}$ (on the left), and the second one with a weighting of $\{3,1,1\}$ (on the right) for respectively the *impact*, the *vulnerability* and the *threat*. The second weighting helps to mitigate the effect of our random generation, as it puts a bigger focus on the impact value (which depends on the policies).

Thus, for each request we have a given threat value, a given vulnerability value and can calculate the corresponding impact (based on the generated policies). Each request is first evaluated by the access control policy to see if it should be accepted or not. A second decision point is the risk value which is compared to a threshold. In the following we describe the obtained results.

B. Results

We want to evaluate the influence of our system on an existing access control policy, so basically how many more requests will be rejected than without the risk threshold. For this purpose, we conducted the experimentations against different risk thresholds. Another interesting information is the *coherence* of the risk-rejections, by this we mean how many requests are rejected by both, the policy and the risk threshold.

The results are shown in stacked bars for different threshold values (from 0.4 to 0.9). The figures must be read in this way:

- **Lightgray** (bottom of each bar), the ratio of policy-only-based rejects. So, $\{Policy = Reject, Risk = Accept\}$.

[11] http://www.net-security.org/secworld.php?id=8922

- **Darkgray** (middle of each bar), the ratio of coherent rejects. So, $\{Policy = Reject, Risk = Reject\}$.
- **Gray** (top of each bar), the ratio of risk-only-based rejects. So, $\{Policy = Accept, Risk = Reject\}$.

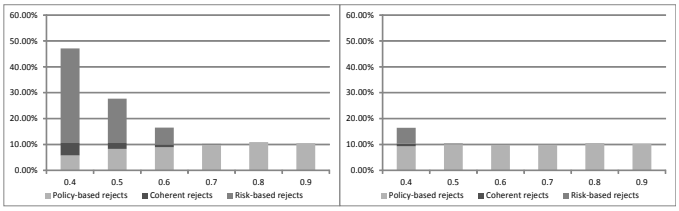

Fig. 3: Ratio of rejected requests in relation to a threshold, for a policy-rejection rate of 10%

In Fig. 3 (policy-rejection rate of 10%), the risk threshold has a significant influence on the global rejection rate only for a very low threshold (namely 40% rejections for a threshold of 0.4). With a greater threshold, the risk does not reject many requests. This observation is even emphasized with the second weighting scenario (focused on the impact value). The weighting reduces significantly the rejection rate in comparison to the non-weighted scenario (only 7% for a threshold of 0.4).

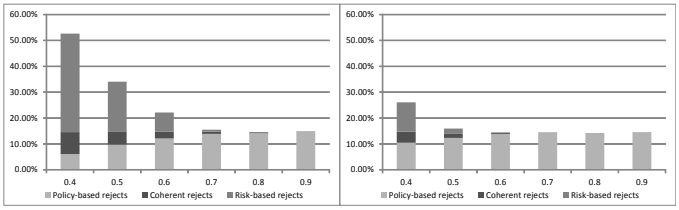

Fig. 4: Ratio of rejected requests in relation to a threshold, for a policy-rejection rate of 15%

In Fig. 4 (policy-rejection rate of 15%), the risk has a more important influence than in the previous scenario (around 45% risk-based rejections for a threshold of 0.4). In comparison, there are also more risk-based rejections for higher thresholds. Another point is the coherence, which is more important than previously. The second weighting, reduces the global ratio of rejected requests, but is more coherent than in Fig. 3. Also, the increase of rejected requests is more significant than for the non-weighted scenario (around 15% for a threshold of 0.4).

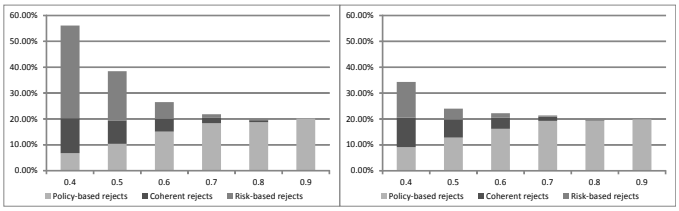

Fig. 5: Ratio of rejected requests in relation to a threshold, for a policy-rejection rate of 20%

In Fig. 5 (policy-rejection rate of 20%), the influence of the risk grows in comparison to the previous cases (around 50% risk-based rejections for a threshold of 0.4). Moreover, the effect "lasts longer" when increasing the threshold. Even for threshold of 0.7 there is still a non-negligible amount of risk-based rejected requests. Moreover, the coherence rate is higher than previously. This is even more true for the second weighting scenario, the risk decision becomes more coherent with the policy decision. Also, there are now around 25% of risk rejections.

Finally, Fig. 6 (policy-rejection rate of 25%) shows the most important risk-based rejection rate of the four scenarios (around 52%). Still, we can notice that the risk effect is getting more important, even for a higher threshold, and the coherence is globally increased. In opposition to the previous scenarios, Fig. 6 shows some risk-only-based rejections even for a threshold of 0.8. The statement made for Fig. 5 can be highlighted in this scenario, i.e, the coherence is more important in the case of the weighted risk calculation. Still, the influence of the weighted risk calculation is bigger than in the non-weighted scenario (around 35% of risk rejections).

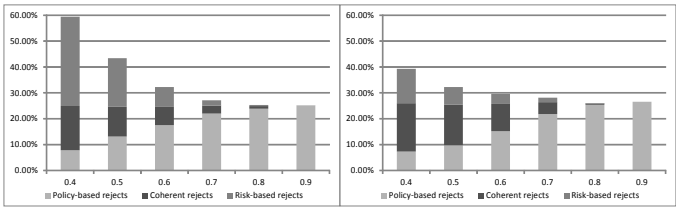

Fig. 6: Ratio of rejected requests in relation to a threshold, for a policy-rejection rate of 25%

C. Discussion

Globally, when analyzing the results for our experimentations, we can make two main observations:

- the less policy-rejections there are, the less important is the risk influence. So for systems with low rejection rates, the risk has not a big influence and does not reject many requests. On the contrary, for high policy-based rejection rates, the risk influence becomes more present: globally there are more requests rejected based on the risk, even for greater thresholds.
- the first observation is emphasized with the weighted scenario where the focus is put on the *impact* value. The increase of risk-based rejections for systems with higher rejection rates is more significant. But the weighting also improves the coherence of the risk decision with the policy decision. For the weighted risk calculation, the risk-rejections are globally more in accordance with the policy-decision than without the weighting.

These observations lead us to the conclusion that our system behaves as expected. The risk adapts to the systems behavior and the access control policies defined by the users of the community. Globally, when there are many rejects for a given system, this means that there are more reasons to be cautious. Thus, the risk, for the same threshold will reject more requests than for lower policy-based rejection rates.

Moreover, the global influence of the introduction of the risk metrics seems fair, as long as the thresholds remains reasonable. By that we mean that our system does not suddenly introduce a huge amount of additional rejected requests. Overall, the organization trusts its users but does only intervene in very few cases.

Finally, the experimentations with the weighted risk formula shows the coherence of the defined risk metrics with our objectives. The risk-decision is more coherent with the policy-decision when mitigating the influence of the randomly generated values for our experiments.

V. Related work

Dimmock [20] considers that the trust is inherently linked to risk, and both, must be considered for decision making. They offer a solution based on the probability density function and the trust to estimate the level of risk for each interaction. Access is granted when the risk is low enough (or the trust high enough). In opposition, in our work we also take into account the trust considerations, but additionally the vulnerabilities of the system and the impact, a very important information for evaluating the risk.

In [23] authors present CollAC, a role based access control framework designed for collaborative environments with an architecture quite similar to XACML [24]. Similarly to our approach, CollAC allows users to define access control policy on a resource they share and/or own. However, resources can be owned by multiple users, and for decision making a combination is used. A feed-back system is put in place for conflicting policies. As our access policy model, CollAC is formal (hybrid logic) [25]. However, the CollAC access control policies cannot be easily adopted at the hight level of PSN, as they are subjective and role based, which do not fit well with dynamic environments.

In [26] Y. Chen and B. Malin propose an unsupervised learning framework designed for the detection of Anomalous Insiders in Collaborative Environments via Relational Analysis of Access Logs and call it CADS (Community based Anomaly Detection System). CADS mines users relations for modeling behavior patterns and detects anomalies by determining if a particular user's behavior is different than expected. The context of this work is quite similar to ours as it considers the flexibility of the environment. In addition, the authors consider that collaborative information systems are inherently designed to support team-based environments. We agree with this point of view under the condition that each team is properly identified within the collaborative environments. Because, with our Attribute Based Access Control ABAC policy model [4] we deal with this challenge by allowing users to define a policy for a group of users, i.e, team, based on the group unique identifier. However, the objectives of our work are different. We aim to promote the fine-grained user policies by risk-based global policies designed for a higher abstraction level.

The context of the work presented in [27] is quite close to the one we look at in term of social networking and Collaborative Access Control aspects. In this work a new class of access control policies is proposed: *Collaborative Security Policies*, based on the multitude of collaborative users' for the ownership regarding to a given resource. The identification of the collaborative users is based on semantic web technologies. The main idea is to represent relations between users in the Online Social Networks (OSN) by a directed graph. Then, based on a minimum trust level of this collaborative relations and a maximum depth, access conditions are evaluated. These two parameters are determined by the resource owner. Note that, if several users are related (owner) to a given resource, a set of access rules corresponding to each owner will be defined and enforced based on the typology-based access control described in [28]. The major difference between this work and ours is that, for the access control enforcement, we do not implicate all the collaborative resource owners directly based on their rules. Instead, we consider them based on the resource's impact. Thus, we keep resource sharing simple for users by just indicating the triplet: subject(s), object and action.

In [29], [30], *Organization Based Access Control* is presented. OrBAC is an access control model in which the organization is the core entity of the authorization model. In OrBAC the main access control entities, namely: subject, object and action are respectively abstracted to: *role, view* and *activity*. A variant of this model designed for collaborative environment is the multi-OrBAC model [31]. The novelty in this policy model is that the basic entities' abstraction becomes respectively: *role in organization*, *activity in organization* and *view in organization*. Even if multi-OrBAC is suited for distributed environment and considers the context, it is still static. More precisely, a dynamic context change has no direct effect on the policies. In addition, it is strongly based on role, which can be considered as a drawback in the case of policy updates, especially in a collaborative environment.

The work proposed in [32] is based on the concepts of Risk Adaptable Access Control (RAdAC). This risk-based access control consists of five core characteristics for evaluating the risk value: *Operational Need*, *Security Risk*, *Situational Factors*, *Adaptable Access Control Policy* and *Heuristics*. The paper proposes the formalization of those concepts by extending the UCON Usage Control Model. However, this proposal does not provide any details about neither how to assess those different parameters, nor how they should be implemented. Moreover, in opposition to our work, this risk definition does not really respect any existing standard definition, even if the five given characteristics can be mapped to our concepts of *Impact*, *Vulnerability*, *Threat* and *Threshold*.

The authors of [33] propose a risk approach based on the economic theory of decision-making under uncertainty. It combines the *likelihood* (the trust beliefs) with the *impact* (outcomes) and implement it for a spam-detection system. Outcomes can also be of positive nature and thus are weighted through preference scaling functions. However, it does not take into account the vulnerability and is limited to an integration to Trust-Based Access Control systems.

In [21], the authors propose an access control system built with a risk approach. The risk calculation is based on fuzzy inferences (*if/then* rules) and uses an object sensitivity score system (similar to FICO) and a subject clearance score system. Thus, the system calculates a risk value (between 0 and 1) for each request of a subject on an object. In opposition to our work, the value does not prevent a too risky access, but allows a temporary access and builds *post-obligations* that have to be fulfilled after the access. Moreover, it does not take into account the vulnerability of the system and cannot be used to complement an existing access control system (such as RBAC rules).

Traditional RBAC systems are experienced, robust and already implemented in lots of systems. We argue that re-

placing existing systems with a completely inexperienced and probabilistic approach is not only non-optimal but can also be very counter-productive in some cases. This is why, we propose an approach to improve such existing systems with our risk-based approach. Both systems, role-based and risk-based can co-exist in the same environment and the decision of both approaches can be combined to get a more accurate result.

VI. Conclusion

In this paper, we proposed an enhancement for existing access control systems with risk metrics. In the context of collaboration environments, and more specifically in Professional Social Networks (PSN), this allows organizations to delegate the definition of fine-grained access control policies to their users but still having a control over their resources. Based on standard risk management methodologies, we define the risk for an incoming request through three values: the impact, the threat and the vulnerability. By defining a risk-threshold, the organization hosting the resources can deny the access request.

We have conducted evaluations to see how the proposed system would interfere with a traditional system without the risk metrics. Our evaluation has some limitations, as it is working on simulated data, and thus it is difficult to see the benefits and drawbacks on a real system. This will consist in the next step of our research work. Also, it has to be combined with a trust calculation approach, work which has been done in parallel to this paper.

References

[1] J. Rifkin, "Lâge de laccès," *La révolution de la nouvelle économie*, p. 177, 2000.

[2] R. D. Rowley, "Professional social networking," *Current psychiatry reports*, vol. 16, no. 12, pp. 1–6, 2014.

[3] C. M. Hoadley and P. G. Kilner, "Using technology to transform communities of practice into knowledge-building communities," *ACM SIGGROUP Bulletin*, vol. 25, no. 1, pp. 31–40, 2005.

[4] E. Zahoor, O. Perrin, and A. Bouchami, "Catt: A cloud based authorization framework with trust and temporal aspects," in *Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2014 International Conference on*. IEEE, 2014, pp. 285–294.

[5] D. Hardt, "The oauth 2.0 authorization framework," 2012.

[6] D. F. Ferraiolo and D. R. Kuhn, "Role-based access controls," *arXiv preprint arXiv:0903.2171*, 2009.

[7] D. F. Ferraiolo, R. Sandhu, S. Gavrila, D. R. Kuhn, and R. Chandramouli, "Proposed nist standard for role-based access control," *ACM Transactions on Information and System Security (TISSEC)*, vol. 4, no. 3, pp. 224–274, 2001.

[8] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[9] R. S. Sandhu, "Role-based access control," *Advances in computers*, vol. 46, pp. 237–286, 1998.

[10] S. S. Shim, G. Bhalla, and V. Pendyala, "Federated identity management," *Computer*, vol. 38, no. 12, pp. 120–122, 2005.

[11] G. Zhang and M. Parashar, "Context-aware dynamic access control for pervasive applications," in *Proceedings of the Communication Networks and Distributed Systems Modeling and Simulation Conference*, 2004, pp. 21–30.

[12] D. Kulkarni and A. Tripathi, "Context-aware role-based access control in pervasive computing systems," in *Proceedings of the 13th ACM symposium on Access control models and technologies*. ACM, 2008, pp. 113–122.

[13] Y.-G. Kim, C.-J. Mon, D. Jeong, J.-O. Lee, C.-Y. Song, and D.-K. Baik, "Context-aware access control mechanism for ubiquitous applications," in *Advances in Web Intelligence*. Springer, 2005, pp. 236–242.

[14] M. J. Covington, W. Long, S. Srinivasan, A. K. Dev, M. Ahamad, and G. D. Abowd, "Securing context-aware applications using environment roles," in *Proceedings of the sixth ACM symposium on Access control models and technologies*. ACM, 2001, pp. 10–20.

[15] National Institute of Standards and Technology, "Information Security - Guide for Conducting Risk Assessments," 2002.

[16] N. Mayer, "Model-based Management of Information System Security Risk," Ph.D. dissertation, University of Namur, Apr. 2009. [Online]. Available: http://tel.archives-ouvertes.fr/tel-00402996

[17] "AS/NZS 4360 SET Risk Management, Australian/New Zealand Standards," 2004.

[18] C. M. Johnson, "A survey of current research on online communities of practice," *The internet and higher education*, vol. 4, no. 1, pp. 45–60, 2001.

[19] G. Zacharia, "Collaborative reputation mechanisms for online communities," Ph.D. dissertation, Massachusetts Institute of Technology, 1999.

[20] V. Cahill, "Using trust for secure collaboration in uncertain environments," 2003.

[21] Q. Ni, E. Bertino, and J. Lobo, "Risk-based access control systems built on fuzzy inferences," in *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*. ACM, 2010, pp. 250–260.

[22] I. Molloy, L. Dickens, C. Morisset, P.-C. Cheng, J. Lobo, and A. Russo, "Risk-based security decisions under uncertainty," in *Proceedings of the second ACM conference on Data and Application Security and Privacy*. ACM, 2012, pp. 157–168.

[23] S. Damen, J. den Hartog, and N. Zannone, "Collac: Collaborative access control," in *Collaboration Technologies and Systems (CTS), 2014 International Conference on*. IEEE, 2014, pp. 142–149.

[24] E. Rissanen, "extensible access control markup language (xacml) version 3.0," 2013.

[25] G. Bruns, P. W. Fong, I. Siahaan, and M. Huth, "Relationship-based access control: its expression and enforcement through hybrid logic," in *Proceedings of the second ACM conference on Data and Application Security and Privacy*. ACM, 2012, pp. 117–124.

[26] Y. Chen and B. Malin, "Detection of anomalous insiders in collaborative environments via relational analysis of access logs," in *Proceedings of the first ACM conference on Data and application security and privacy*. ACM, 2011, pp. 63–74.

[27] B. Carminati and E. Ferrari, "Collaborative access control in on-line social networks," in *Collaborative computing: networking, applications and worksharing (CollaborateCom), 2011 7th international conference on*. IEEE, 2011, pp. 231–240.

[28] B. Carminati, E. Ferrari, and A. Perego, "Enforcing access control in web-based social networks," *ACM Trans. Inf. Syst. Secur.*, vol. 13, no. 1, 2009. [Online]. Available: http://doi.acm.org/10.1145/1609956.1609962

[29] A. A. E. Kalam, R. Baida, P. Balbiani, S. Benferhat, F. Cuppens, Y. Deswarte, A. Miege, C. Saurel, and G. Trouessin, "Organization based access control," in *Policies for Distributed Systems and Networks, 2003. Proceedings. POLICY 2003. IEEE 4th International Workshop on*. IEEE, 2003, pp. 120–131.

[30] F. Cuppens and A. Miège, "Administration model for or-bac," in *On The Move to Meaningful Internet Systems 2003: OTM 2003 Workshops*. Springer, 2003, pp. 754–768.

[31] A. A. El Kalam and Y. Deswarte, "Multi-orbac: A new access control model for distributed, heterogeneous and collaborative systems," in *8th IEEE International Symposium on Systems and Information Security*, 2006.

[32] S. Kandala, R. Sandhu, and V. Bhamidipati, "An attribute based framework for risk-adaptive access control models," in *Availability, Reliability and Security (ARES), 2011 Sixth International Conference on*. IEEE, 2011, pp. 236–241.

[33] N. Dimmock, J. Bacon, D. Ingram, and K. Moody, "Risk models for trust-based access control (tbac)," in *Trust Management*. Springer, 2005, pp. 364–371.