To authorize or not authorize: helping users review access policies in organizations

Pooya Jaferian
University of British Columbia
Vancouver, Canada, V6T 1Z4
pooya@ece.ubc.ca

Hootan Rashtian
University of British Columbia
Vancouver, Canada, V6T 1Z4
rhootan@ece.ubc.ca

Konstantin Beznosov
University of British Columbia
Vancouver, Canada, V6T 1Z4
beznosov@ece.ubc.ca

ABSTRACT

This work addresses the problem of reviewing complex access policies in an organizational context using two studies. In the first study, we used semi-structured interviews to explore the access review activity and identify its challenges. The interviews revealed that access review involves challenges such as scale, technical complexity, the frequency of reviews, human errors, and exceptional cases. We also modeled access review in the activity theory framework. The model shows that access review requires an understanding of the activity context including information about the users, their job, their access rights, and the history of access policy. We then used activity theory guidelines to design a new user interface named AuthzMap. We conducted an exploratory user study with 340 participants to compare the use of AuthzMap with two existing commercial systems for access review. The results show that AuthzMap improved the efficiency of access review in 5 of the 7 tested scenarios, compared to the existing systems. AuthzMap also improved accuracy of actions in one of the 7 tasks, and only negatively affected accuracy in one of the tasks.

1. INTRODUCTION

Understanding and authoring access control policies has been known as a challenging problem [29, 33, 30]. But the focus of previous studies were on personal access control, where the data owner, policy maker, and policy implementer are the same person. This problem has not been extensively studied in organizational context. Bauer et al. [1] found that managing access control policies in organizations faces a unique set of challenges. In large organizations, those who make policies are different from those who implement these policies. Therefore, developing a shared understanding of policy between different stakeholders is challenging. In this paper, we explore and address this problem by proposing and evaluating AuthzMap, a new user interface for sense making and reviewing implemented access policies or, in short *access review*.

Access review is an important IT security activity in organizations, where the managers make the access policy and security administrators implement it. The managers are mandated by many security regulations (e.g., SOX [35], HIPAA [6]) to regularly review and validate the access privileges of users. However, Cser [10] suggests that access review for every 2,000 to 3,000 users consumes approximately one full-time-employee equivalent per year, and many organizations cannot even finish one access review process before a new campaign begins.

Recent security incidents that cost governments and organizations billions of dollars show the importance but yet lack the ability in reviewing users' access rights. For example, the US army soldier, Chelsea Manning, who leaked the US embassy cables was cleared to access classified resources when she was on training as an intelligence analyst. She then changed her job and location multiple times before going to Iraq. According to Swensen [34], if a superior reviewed Mannings' access and requested the revocation of unnecessary privileges, she would not have been able to leak the data.

The overarching goal of this paper is to investigate improvements technology support for access review. Towards this goal, we performed two studies. In the first study, we conducted 12 semi-structured interviews with security practitioners to understand how people make sense, and review access of users, and to identify the challenges in access review. We then designed a new interface, guided by activity theory guidelines by Kaptelinin and Nardi [19], to address the identified challenges. We named the proposed interface AuthzMap. We then conducted an online study with 340 participants to test if AuthzMap improves the usability over two of the existing interfaces.

Besides understanding access review activity and improving access review tools, this research has broader implications for the design of access management interfaces. Our results suggest that context plays a role in understanding the access privileges of an enterprise user. The context of a user-to-role assignment includes the user's current and past jobs, the history of the user-to-role assignment, other users' access privileges, and those who requested, approved, and implemented the access. Therefore, tools that manage user-to-role assignments should take into account the aforementioned information, and present them in a way that reflects the spatial layout and temporal organization of the context.

2. BACKGROUND

Organizations use many IT applications to run their business. Employees who use an application for their job are provided with a set of access privileges, and other employees should be prohibited from accessing the application. Therefore, applications provide a set of *permissions* that can be assigned to a *user* to control what the user is authorized to do. Sometimes, permissions are grouped into *roles* to simplify the provisioning process. As the number of users and applications grows, the management of users, roles, and permissions becomes challenging. Therefore, organizations are man-

Symposium on Usable Privacy and Security (SOUPS) 2014, July 9–11, 2014, Menlo Park, CA.

dated by many security regulations (e.g., [35]) to frequently perform access reviews to make sure that users have the least set of privileges required for their job.

Next, we describe how access review is performed through an example. In an organization, security administrator, John, sends a request to manager Bob to review the access privileges of fifty employees who work in Bob's department. Bob is provided with the list of employees and their roles. He reviews the list of users one user at a time, looks at their roles, and verifies if the user-to-role assignments are valid. For example, Bob sees that Alice is assigned to 20 roles (R1, R2, ... R20). Bob needs to understand the meaning of the roles, what they authorize Alice to do, and if the authorizations are required for Alice to do her job. If an authorization is required, Bob *certifies* the assignment of Alice to the role. Otherwise, he *revokes* the assignment. If Bob cannot understand the meaning of a role, he may communicate with John or other managers to ask what privileges are associated with the role. This example shows that access review requires analysis, communication, and collaboration with other stakeholders.

3. RELATED WORK

There have been few studies related to access management in organizational context. Bauer et al. [1] performed a field study of access control practices in organizations. They suggest that the implemented access policy and the record of changes should be understandable and visible. Our findings confirm this, and our proposed interface improves understandability and visibility of access policy.

As opposed to access review, the problem of policy authoring has been previously studied. Brodie et al. [4] designed a privacy policy management workbench called SPARCLE to create policies in natural language. Although SPARCLE was successful in facilitating policy definition and management, it was not used or evaluated for the access review. Inglesant et al. [15] studied personal access control in Grid computing context. They showed that resource owners have difficulty expressing policies in RBAC and they prefer the use of natural language. Reeder et al. [29] proposed a new UI named "expandable grid" for understanding effective access policy in case of conflicting access rules. Expandable grid improves the understanding of access policy by end-users of commodity OSs, and their main goal is to address the issue with conflicting access rules that happen regularly in the Windows file system. The data from our interviews show that in enterprise environments, standard role-based access control without negative authorization rules is used. We also adopt the idea of expandable grid for use in an organizational context and use it in the design of AuthzMap. Smetters and Good [33] studied the use of policy authoring for personal documents. They found that users rarely change access policies, and tend to specify complex and error-prone policies. Our findings suggest that unlike access control for documents, the users' accesses change frequently in organizations. Vaniea et al.[36, 37] examined the effect of proximity of access management interface and the resources. They show that users detect errors better if controls are positioned near resources. Their proposed method was implemented and evaluated in the context of managing photo album privacy policy. In an organizational context, this proposal might not be possible, as resources do not have direct graphical representation, and the number of resources and permissions could be large. Beckerle and Martucci [2] identified six guidelines for designing usable access control rule sets, and showed that implementing those guidelines will help understandability of access policies. Their proposed solution can be used before presenting access control rule set in AuthzMap to reduce the complexity of policy.

Table 1: Interview participants' demographics

Code	Job title	Organization
P1	Security Manager	Insurance
P2, P8	Security Analyst	Insurance
P3, P7	Security Manager	Software
P4, P5	Security Administrator	Software
P6	Compliance Manager	Software
P9	Consultant	Health care
P10, P11	Consultant	Financial
P12	Consultant	Software

4. STUDY 1: UNDERSTANDING ACCESS REVIEW ACTIVITY

The initial goal of the interview study was to understand how organizations perform identity and access management, and the challenges they face. After initial analysis of interviews, we turned our attention to answering the following research questions: (1) Why organizations perform access review? (2) Who are the involved stakeholders? (3) Why access review is challenging? (4) How better decisions can be made during access review?

4.1 Methodology

We conducted 12 semi-structured interviews with security practitioners responsible for access management in large organizations. The list of interviewed participants, their roles, and their organization sectors are shown in Table 1. The scope of the interviews was various activities related to identity and access management (see Appendix A for the interview guide). The interviews were conducted by one or two interviewers in the workplace of the participant (8 interviews) or over the phone (3 interviews). The length of the interviews was between one and three hours. The interviews were audio-recorded, and transcribed.

We analyzed the interview data using grounded theory methodology [7]. We imported the transcripts of the interviews to a qualitative analysis software (Qualrus v2.1), and then coded them with open-coding technique with the codes emerging from the data. We then performed axial-coding by combining conceptually similar codes and identifying various themes across the data. At this step, we found that identity and access management involves several activities including access review. We also found different themes related to each activity including the goal, actors, artifacts, division of work, rules, and challenges. We identified access review as one of the most challenging activities. Therefore, we chose it as the core concept, and performed a round of selective coding to answer the research questions. We reached theoretical saturation [7] and stopped recruitment after recruiting 12 participants.

4.2 Results

In this section, we first provide a detailed description of access review activity using the activity theory framework, and then discuss the identified challenges.

We use the triangle model of activity proposed by Engeström [12] to lay out our description of access review (Figure 1). We will later refer to this formulation when we justify our design decisions.

The goal of the activity: Access review is an activity with the goal of verifying users' access rights to minimize the risk of unauthorized and unmanaged access and comply with regulatory legislations.

Subject: "Reviewer" is the main actor in the activity who performs access review. Our participants indicated that the following stakeholders act as reviewers:

Managers: Most of the participants indicated that managers review employees under their authority. P1 further described the role

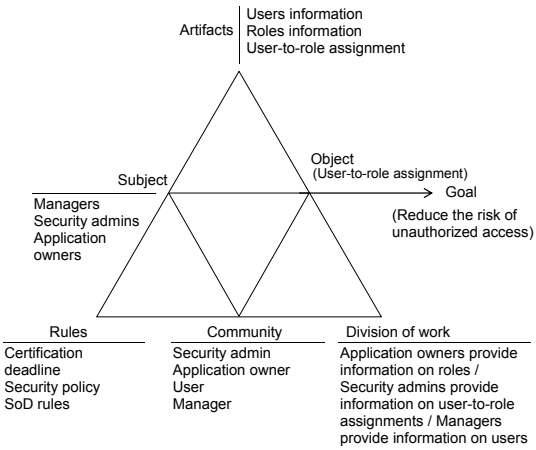

Figure 1: Overview of Access Review Activity

of the manager in access review: *"[Manager asks:] what access does Jim have? I'd like to review Jim's access because he's changing roles within my department, there's no official job posting but I'm doing a realignment and I would like to review Jim's access. So you need to do a specific report on Jim, which is to say here is the access profile that Jim has."*

Application owners: Two of our participants indicated that an application owner reviews the users who have access to the application, and certifies or revokes the users access privileges: *"Our team wrote some [scripts]. It goes out and it collects from these 80 or so applications, what the access lists are, what the rights are, it creates a report, we put it in a service desk ticket. Then it goes out to the [application owners] and they review it."* (P3)

Security administrators: P6 explained that his team is responsible for security compliance of a large enterprise application, and therefore he performs access reviews: *"We send a request to the manager that says Bob has changed from position A to position B. They are requesting position B roles. We are going to remove his position A roles. Do you agree with that?"*

Object: The object towards which the activity is performed is a user-to-role assignment. When managers or security admins perform access reviews, they review a set of roles assigned to a user (user access review). When application owners perform reviews, they review a set of users assigned to a role (application access review). We limit the scope of the AuthzMap to user access review. The same design techniques can be applied for building an interface for application access review.

Community and division of work: Access review involves security team members, employees, managers, and application owners. Involved stakeholders divide the work as follows: A member of the security team requests review of users' access rights. The reviewer (a manager in most cases) receives the request. He goes through the list of users, selects a user, and identifies the user's roles. For each user-to-role assignment, he chooses to certify or revoke the assignment. The reviewer might contact the application owners, the user, or the security team when he is unable to determine the correct action.

Rules and constraints: Different rules and constraints impact access review. (1) The security policy of the organization determines the validity of a user-to-role assignment. For example, P9 explained that in health care, they follow an optimistic security paradigm [28] and allow more access than usual so the physicians can access patients' files in emergency cases: *"So the whole access model in health care tends to be, you let people do what they need to do to get the job done."* (2) Static separation of duties (SoD) rules determine if a user can be assigned to two or more specific roles at the same time. (3) The review deadline set by security team constrains the time window of the review.

Artifacts: Reviewers use three artifacts during access review: (1) User's information, which include the identity related information, the job title, and other attributes like the phone number, email, department, etc. (2) User-to-role assignment information, which include who requested, who approved, and who implemented the assignment, when and why the user was assigned to the role, and who previously reviewed the assignment. (3) Roles' information, which include the role's name, description, the owner, and the permissions assigned to the role.

4.3 Challenges in access review

Our interviewees indicated that access review is a challenging activity. We classified these challenges into 5 categories:

Scale: Access review can involve large number of users, roles, and permissions. P6 explained that just one of the large applications in his organization has 16,000 users, up to 115 roles per user, and up to 407 permissions per role. He also indicated that reviewers have to review up to 200 users in a review activity. While these numbers vary from application to application, and from organization to organization, they show the magnitude of data that a reviewer needs to deal with.

Lack of knowledge: When managers act as reviewers, they do not have the expertise to understand the meaning of roles and permissions. P2 illustrated this problem in detail: *"we send these godawful long reports to the new manager hiring the employee is going into, saying "let us know which access this person needs to keep and what they need to remove." And a lot of it's, you know, cryptic RACF information and stuff they just have no idea what they're even reading so they either take their best guess and say, ok, then maybe this sounds kind of like something they might need. Or they just say they need it all."*

Frequency: While reviewing access is not the main job of managers, they are frequently asked to perform this activity. For example, P3 explained why they perform quarterly access reviews: *"... Once a quarter! We do quarterly access reviews. [...] Once a year is never good for any control because if you fail, you fail; at least twice a year you have a chance to remediate."* Additionally, P3 talked about ad-hoc access reviews: *"Every day, [access management software] looks at [every] person who has access and says has the person changed in any way. Did they move departments, did they move to geographical locations - if so it triggers an event which puts a ticket into the service desk system, sends a note to the Access Reviewers and says you need to review this ..."*

Human Errors: P3 described why human errors are common during reviews: *"So the policies of the company states that the business is responsible for the access. So the ultimate decision maker is the business. However they failed because it's a human process right? It's eyeballing [and] sometimes the lists are large."* Such errors would be costly for organizations, both in terms of leading to data breaches, and failing compliance reviews.

Exceptional Cases: In organizations, the validity of user-to-role assignments cannot be determined accurately only by knowing the user's job function. Users might need to fill in another employee's role for a period of time, or they might need temporarily access certain resources when they are on training. P6 explained a case where they thought they should remove existing access from a user because he asked for new access. They later realized the user is on training and still has his old job: *"The manager says no, he is training this person, as replacement, for three months."*

5. AUTHZMAP DESIGN GOALS

To design a new access review tool, we followed a design approach proposed by Kaptelinin and Nardi [19]. In this section, we present three main design goals. For each goal, we first present the theoretical support, and then we use the field study data to describe how we apply theory to the design of an interface for access review.

Flexible support for review actions: The goal of access review is verifying access privileges. This goal can be broken down to lower level subgoals, and actions to satisfy those subgoals. These actions can include: viewing list of users and identifying them, identifying users' job function, checking the list of users' roles, and certifying or de-certifying user-to-role assignments. To address the *Scale* challenge, a tool can help users perform the above actions more efficiently. This can be achieved by more flexible search and filtering mechanisms to view and identify users, and applying decisions in batch. Technology should also support alternative ways to attain an activity goal [20]. To achieve this, we present information at different levels of abstraction. The user can choose the right level of detail, based on his knowledge and understanding of the access policy. For example, a user with the knowledge of the access policy can use more abstract view, but a user who needs more information can use detail view. This approach can address the *lack of technical knowledge* challenge.

Visibility of context: Activity theory emphasizes that tools and artifacts used during an activity are part of the context, and the technology should facilitate access to those artifacts, integrate them with each other, and present them in a way that reflects the spatial layout and temporal organization of the context. The context of an access review activity includes users, roles, and user-to-role assignments. In addition, the following artifacts are part of the context and can be used for making access review decisions:

(1) Job changes: Our participants indicated that when users change their job or move between departments, their access changes. For example, P6 explains why job changes can be an important contextual information for access review: *"Now what happens is that we have a report that runs every single day and it tells me [if] people transfer [to another department] or change [their job]. [For example,] she gets a promotion. She went from warehouse manager to public relations manager. She will request something. I need a public relations manager role. My team goes automatically: 'why? That's not what you are. You are warehouse. No, I got a promotion, I'm this. Okay, we'll give you these three but you are losing those three."'* Providing job changes help reviewers better understand how and why the access privileges of users change, and therefore, address the *lack of knowledge* challenge.

(2) Other users' access: During access review, reviewers may need to review many users instead of one. These users have certain roles in common (e.g., basic access to the Internet, email, Sharepoint). For example, P1 explained that users who are doing the same job usually have similar access: *"...a manager who hired a new employee [and] who knew that you had the access that you needed to do the job for him or her would say, 'Oh, make this new employee's access just like yours.' And so then an employee would then inherit privileges based on the success of a previous employee in terms of doing that job."* Therefore, comparing access privileges of a user known to reviewer to that of an unknown users will facilitate sense making. This will address *lack of knowledge*, and *scale* challenges and reduces *human errors*.

(3) Previous reviews: The reviewer can employ the past review decisions and replicate them in his review. Replication is particularly useful if none of the user's attributes has been changed since the last review. Having access to and using past reviews can address *frequency*, and *scale* challenges, and reduces *human errors*.

(4) Other users involved in the activity: The process of provisioning users with access privileges is a collaborative activity between different stakeholders. Therefore, the interface should show who requested the access, who approved the request, and who executed the provisioning of access. (P12) explained that such information will help reviewer make an informed decision: *"So again, you think of the attestation process or even at any moment in time on a view user, we always talk about helping somebody make informed choice. So if I'm evaluating the correctness of an SAP account and I can look at when it was requested, who reviewed it, who approved it, when your last login time was, I can serve to make a pretty informed choice about why you have this or its level of appropriateness."* This can address *lack of knowledge* of why a user has certain access privileges.

(5) Policy violations: Our previous survey [16] shows that SoD violations are the most important violation to be detected during access review. Therefore, they should be highlighted on the interface. This can address *scale*, and *lack of knowledge*.

Make history visible: According to [19], analysis of the history of an activity can reveal the main factors influencing the development of the activity. Furthermore, Hollan et al. [14] studied experts working in complex environments, and suggested historical information can be incorporated in cognitively important processes. For access review activity, historical information can help reviewers understand how the policy has evolved over time, and therefore make better decisions in uncertain scenarios. This would address the challenges of *scale*, and *exceptional cases*.

To incorporate history in the interface, we first identified which of the three access review artifacts (users, roles, and user-to-role assignments) carry historical information. Interview data revealed that users, and user-to-role assignments (unlike roles) change over time, and therefore, have historical information. For example, P6 explained that employees frequently change their job, but roles should be designed in a way that are not impacted by such changes: *"[Employees' position] changes a lot when you start going through economic churns. So when you are laying-off 50 people at a time, 100 people another time, or department consolidations. I can tell you I've been in this role for two and a half years and I've seen five department consolidations in finance alone. So when you see all those changes happening, those composite roles hurt you. Because then you have to keep generating them over and over again."* Also when we asked P4 about how frequently they make changes to the roles, she responded: *"We don't. I wouldn't say never - very rarely. If we were to add a new region, which I don't think there are any left to be added at this point."* Therefore, AuthzMap visualizes the history of users' job changes, and the history of user-to-role assignments, and correlates them with each other. Showing the history can addresses *frequency* challenge by showing previous decisions, and help with understanding of *exceptional cases*.

Knowledge sharing: According to Kaptelinin and Nardi [19], technology should help in problem articulation and seeking help from colleagues. The interview participants indicated that reviewers hardly understand the meaning of the roles and access privileges. Therefore, our participants used the following strategies to mitigate the lack of knowledge:

(1) P6 talked about translation of technical terms to business related terms to help reviewers understand the meaning of roles: *"... and we get this huge profile - here's all the access the user has. We then have to translate that into more of an English format for the individual."*

(2) P7 described the use of communication channels to get help with certification decisions: *"The security coordinators take it to the [application owner] and explain what the risks are. They're the*

ones who do a kind of mini risk assessment say: OK, such and such business unit wants access to this data for such and such reason."

Therefore, one of the design goals in the proposed interface was to provide knowledge of each access privilege for the reviewers in the form of a description, and list of permissions (in case of using roles). Moreover, communication channels should be available in the interface to get help from other users with the knowledge of roles and permissions. Knowledge sharing would address the challenges of *lack of knowledge*, and can help with *exceptional cases*.

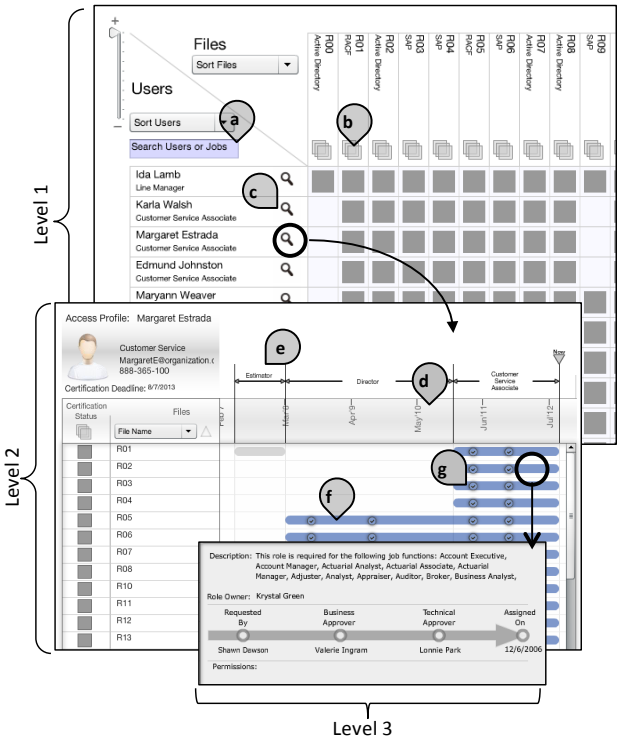

Figure 2: The three levels of the AuthzMap interface. The reviewer is presented with Level 1 of the interface. He can go into Levels 2 and 3 for making further sense of the accesses of the users.

5.1 AuthzMap Interface Design

To realize the goals discussed in the previous section, we designed a new interface and named it AuthzMap. We first built a low-fidelity prototype in Microsoft Visio, and improved it over multiple rounds of internal feedback. We then designed a medium-fidelity prototype in Adobe Flash, and refined it by getting feedback from external usable security researchers, as well as our industrial partner in this project. Finally, we built a high fidelity prototype in Adobe Flash. It loads access control related data through XML files and allows the user to perform access review tasks. We depict the AuthzMap in Figure 2, and with more details in Appendix B.

The AuthzMap uses three levels of abstraction to integrate different contextual artifacts discussed in the previous section. Level 1 shows users and roles in a grid that provides an overview of the overall review activity. The spatial layout of the interface was based on Lampson access matrix [22] model and inspired by the design of Expandable Grid [29]. Users are sorted from top to bottom, based on the number of privileges they have. This allows reviewers to quickly identify users who have large number of privileges. Reviewer can use the sorting and filtering functionality (Figure 2a) to group and compare users with similar job titles, or roles that serve access to similar applications. AuthzMap provides batch certify accelerators (Figure 2b) to certify a role for all users. Reviewer can obtain the detailed access profile of a user using the second level of the interface (Figure 2c).

In Level 2, we integrated contextual information related to the user-to-role assignments, the job changes of the user, and previous reviews. This level also uses a timeline metaphor (Figure 2d) to show the temporal relationship between the job changes (Figure 2e), roles (Figure 2f), and previous reviews (Figure 2g). The reviewer can re-arrange the roles based on the role name, active roles, and the time the role is assigned to the user.

If the reviewer needs more details on one particular user-to-role assignment, he can click on the role bar to go to Level 3 of the interface, which shows the description of the role, the role owner, the permissions assigned to the role, and the workflow through which the user obtained the role. Level 3 allows the reviewer to learn about the meaning of the role. If the reviewer cannot articulate the meaning or the impact of the role using this information, he can use communication channels to seek help by clicking on the name of each stakeholder (e.g., owner, requester, approver, and implementer).

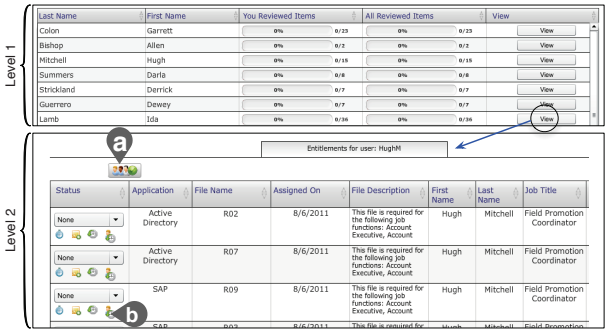

Figure 3: A screenshot of the List interface. Reviewer identifies the user and clicks on the View button. Reviewer is presented with the second level of the interface that includes the list of user's access privileges. The icon marked as (a) allows batch actions on privileges, and the four small icons (marked as b) do the following (from left to right): sets the access expiry time, writes notes for each privilege, shows history of actions on each privilege, and shows history of rejections for each privilege.

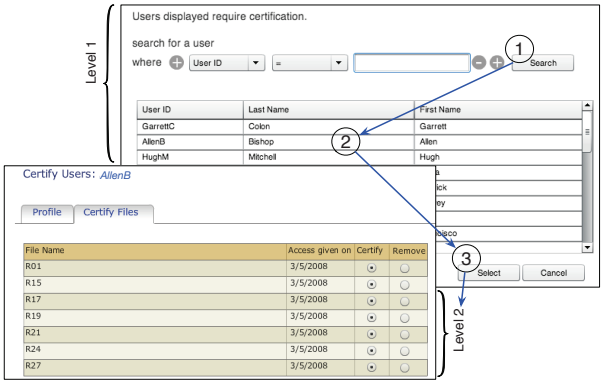

Figure 4: A screenshot of the Search interface. (1) Reviewer searches for a user. (2) Selects the users. (3) Clicks on the Select button and certifies or revokes access privileges in Level 2.

6. STUDY 2: EVALUATION OF AUTHZMAP

The goal of our evaluation was to test if AuthzMap is more usable than the two existing systems. Nielsen defines usability by five quality components [26]: (1) Learnability, (2) Efficiency, (3) Memorability, (4) Errors, and (5) Satisfaction. In this study, we identify efficiency and errors as two main usability goals of the interface, as they are directly related to the challenges described in Section 4.3. At the end of the study, we also collect data about subjective satisfaction of the participants.

6.1 Evaluation Methodology

To evaluate three interfaces, we designed a between-subjects study with 3 conditions (one condition per interface). We asked participants of each condition to perform seven tasks. For each task, the interface was the independent variable, and we measured the following dependent variables: (1) efficiency, by recording time to completion (TTC), and (2) accuracy, by recording correctness of the critical components of the task.

6.1.1 Evaluated Interfaces

We compared the AuthzMap interface to two other interfaces, named Search (Figure 4) and List (Figure 3). The detailed description of each interface is provided in Appendix B. The List interface is known as one of the two access review market leaders, and Search as one of the two the strong performers [10]. We choose not to reveal the actual names of Search and List interfaces as the purpose of the study is not to critique a particular commercial system, but rather compare three different approaches in the design of access review interfaces. The Search interface does not reveal the context at all. A reviewer can search for users, select users one-by-one, and review the user-to-role assignments. The List interface reveals certain contextual information such as the progress of reviewing individual users, the history of previous reviews, and information about the individual users (such as their job and department) and the roles (such as the date the user is assigned to a role, or the description of the role). But these contextual information are not correlated with each other or immediately accessible to the user. We chose to build a prototype of Search and List interfaces over using their full versions for two reasons. First, we wanted the three interfaces to be at the same level of granularity. Second, we did not have access to the installable version of the List interface. Third, prototyping allowed us to instrumentalize the interfaces for the user study.

6.1.2 Participants

We used Amazon Mechanical Turk (MTurk) for recruitment, and paid each participant $2. MTurk has been used as a user study platform for HCI [21] and usable security research [38]. Participants were asked to play the role of managers responsible for access review. Because we did not specifically recruit managers, we used an approach similar to the one by Convertino et al. [9], to provide participants with the beliefs and knowledge of managers. Using our interview data, we first determined managers' level of computer security, review tool, and organizational knowledge. Interviews showed that managers do not have an extensive computer security knowledge, but they understand the concept of access review, and they know the steps for performing it. In addition, managers are trained on using the access review tool (i.e., they are not the first time users of a novel tool). We also assume they are not daily users of the tool (they use it four to two times a year or on an ad-hoc basis). To help participants have similar level of knowledge, we trained them on the basics of access review, and the use of tool to perform reviews (see Section 6.1.3 for the details of our training procedure). We further allowed them to explore the tool and familiarize themselves with it.

6.1.3 Training Material

We designed training material to ensure participants understood the concept of access review, and could apply that understanding using the system. The participants were given a brief training on access control and access review. We followed the recommendations from previous research on designing training materials:

Brief, up to the tasks: Users will learn tools faster when the training focuses on performing the task rather than understanding the rationale behind the task [5]. We avoided training users on details of role-based access control, and concepts such as roles, and entitlements. Instead of using the notion of roles, entitlements, or access privileges, we used the notion of access to files. Previous research shows that participants can understand the meaning of file access control, and they are able to comprehend file access control policies [2, 29, 30].

Use of examples: We used examples throughout the training to explain the access review concepts. We also provided instances of how the interface can be used in interpretation of users' accesses.

Use of text-based material: Online participants can do better with short textual instructions, rather than videos, or demos [13] as it gives participants the opportunity to easily revisit the training during the study.

Use of multi-staged training To avoid overloading participants with training material, and to help them start working on tasks as soon as possible [5], we only taught them the basic access review concepts during the training. Task specific topics such as separation of duties (SoD) violations, privilege accumulation, etc. were taught as parts of the scenarios.

After the training, participants were asked to complete a test to check if they have the required knowledge to do the tasks. We tested the understanding of access control and access review using six multiple choice questions. Multiple choice questions are a reliable and objective way to assess the outcome of the learning, while the answers can be checked automatically [8]. We used standard techniques for designing multiple choice questions [8], and piloted them to ensure their effectiveness.

6.1.4 Study Material

Actual users of access review tools also posses the organizational and contextual knowledge that our participants lacked. For example, a manager may have an understanding of the consequences of having access to a resource, or awareness of the access privileges for doing certain job. Such knowledge is context dependent, that is, we cannot have a clear assumption that a manager always has or lacks such understanding. In the study tasks, we simulated both situations where reviewer has or does not have contextual knowledge and provided participants with documents and material as external knowledge sources (similar to [9]).

We presented participants with three documents: *file catalog*, *application catalog*, and *SoD catalog*. The file catalog showed the list of files that each job function was allowed to access. The interview participants talked about entitlement catalogs, which we changed to the file catalog for the purpose of this study. According to P7: *"One of the things we have been doing is also building a catalog of access requests that people can make [based on their job]."* The

application catalog listed all the applications and their files (entitlements). According to P3, they kept the track of this information in a knowledge base: *"our access procedures state that every application that has any level of criticality is supposed to have a published knowledge-based document in our service desk knowledge base that defines what the application is ..."* The SoD catalog showed pairs of files that caused SoD violations. P12 said they document these rules: *"And again whether they be SOD policies that say you can't have A if you have B or what we call 'restricted access' policies that say you can't have entitlement X if you are not cost center Y or division X or whatever the rule is, the ability to define that rule it lives with the entitlement in the resource catalog."* (P12)

Norman [27] describes that people can rely on *knowledge in the world*, *knowledge in the head* or a combination of both in their activities. To determine the validity of a user's access, reviewers may completely rely on the above documents (knowledge in the world), they may completely rely on their own knowledge (knowledge in the head), or they may use a combination of both. The lab study participants did not have the knowledge in the head of the hypothetical organization. Therefore, all the required knowledge for performing the tasks was included as knowledge in the world in the form of provided materials during tasks.

6.1.5 Study Tasks

After completing the training and the training test, participants were asked to perform seven tasks. We aimed to design tasks with three characteristics [24]: (1) Realistic; (2) Actionable; (3) Avoid Clues or Steps. In order to achieve realism, we designed the tasks based on interview data and a survey we previously performed [16]. Tasks #2 and #3 simulate conditions where the manager knows which access privileges are appropriate for users, and only needs to identify users, and certify or revoke the privileges. Tasks #4 to #6 simulate scenarios where the manager tries to detect access privileges with high risk. To further understand what type of access privileges are risky, we conducted a survey [16] and asked participants to rate the risk associated with certain types of access privileges. We chose to use the top three, which were SoD violations, accumulated privileges, and access privileges to critical applications, and used them to design tasks #4 to #6. The task #7 was a combination of previous scenarios to simulate a more uncertain and complex situation.

Training Task: The goal of this task was to familiarize participants with the interface. As we described in Section 6.1.2, managers will be familiar with their access review tool. This task gave participants an opportunity to perform a guided exploration of the interface, and understand how they can find pieces of information required in the upcoming study tasks. Participants were given the following scenario: "You are asked to identify the following information about "Clay Warren" : (1) his current job title, (2) list of files he has access to, (3) his previous job title, (4) the date of the last access review performed on the user." They were expected to select the correct answer to questions #1, #3, and #4 from seven possible options (including "I do not know"). They should also type the answer to question #2 in a text box.

Common Review: (P1) explained that a common access review scenario is when a manager reviews one user: *"[Manager says:] what access does Jim have? I'd like to review Jim's access because he's changing roles within my department, there's no official job posting but I'm doing a realignment and I would like to review Jim's access."* Therefore, participants were given the following task: "You are asked to review the files *Timothy Larson* has access to. Check the user's access to files, certify the access to those files the user requires to perform his job and revoke those he does not require. Feel free to use the *File Catalog* in the top menu to find the list of files required for performing each job." In this scenario we made an assumption that the manager can determine the correct set of access for users. This is simulated by providing participants with access to a *File Catalog* that shows what files are required for performing each job. Participants are expected to revoke access to two files that are not necessary for Timothy Larson's job.

User comparison: P2 explained that similarity between users with the same job is used to detect excessive and unncessary access: *"if you've got a group of 15 case managers and you bring them into the system, it'll say: ok, 12 out the 15 have 80% of access in common and these two people only have 20%. [...] oh this person has access they should not have, that has been carried over from somewhere else."* To simulate this scenario, participants were given the following task: "In this task you need to certify the access of three users. The certification is only limited to employees with *Loss Control Consultant/Specialist* job function. Identify such users, certify the files that users require to perform their job and revoke the access to the files they do not need. The catalog of jobs, and the required files to perform each job will be provided." In this scenario, we made an assumption that the manager can determine the set of access privileges required for the job and therefore provided participants with *file catalog*. In this task, there were three users with the "Loss Control Consultant/Specialist" job, and one of them had an unnecessary access to a file. Participants were expected to revoke the access to that file.

Privilege Accumulation: Many of our interview participants discussed the privilege accumulation problem in large companies. For example, P6 explained: *"I was warehouse worker, I became public relations. They would request the public relations roles, nobody would take away the other ones and you would wind up with somebody having 50 roles."* Therefore, this task evaluated the interface in finding and resolving accumulated privileges. We gave the participants the following scenario: "Assume you do not know the list of files required for performing each job. In this case, you need to evaluate each user's access to files based on the following rule: *If the user changes job, he should not keep any access from his previous job. Any access that is kept from a previous job should be revoked.* Please review the following users, and revoke invalid accesses according to the above rule: (1) Derrick Strickland, (2) Lynda Robertson." The two target users had two and one permission accumulated from their past job, and participants were expected to revoke those permissions.

SoD Violation Detection: P6 described SoD violations as one of the highest access related risks. He described a case that someone is moving from accounts receivable (AR) to accounts payable (AP), and access to AR and AP systems causes SoD violations: *"So you are going from - you are the AP person, you are going to AR and your AP person needs to be trained [by you] – your replacement. Then we don't like it and it becomes very problematic and we usually want lots and lots of controls if you want the person to have the access."* Therefore, the goal of this task was to evaluate the proposed interface in the detection of SoD violations. We gave the participants the following scenario: "Sometimes a user should not have access to two specific files at the same time. For example, a user can have access to either file A or B but not both, at the same time. This rule is called *Separation of Duties (SoD)*, and having access to those files at the same time is called an *SoD violation*. In this scenario, you are asked to review the files of two users, and detect and eliminate SoD violations. To do so, you should first identify the two files that cause SoD violations, and remove access to one of the files to eliminate the violation. Please check the following

users for SoD violations: (1) Ida Lamb, (2) Maryann Weaver." In this task, each of the users had access to two files that caused SoD violation, and participants were expected to revoke access to one of the files causing SoD violation.

Application Review: P3 noted that they sometimes prioritize the access review according to applications. Critical applications are reviewed first, and in some instances non-critical applications are excluded from the review: *"They run a process which goes out to a subset of all those applications - the ones that we call critical which is SOX applications plus other [...] It goes out and it collects from these 80 or so applications what the access lists are, what the right are, it creates a report, we put it in a service desk ticket. Then it goes out to the [reviewers] and they review it."* In this task, we evaluated interfaces for application specific reviews. We gave the participants the following scenario: "The company uses four applications for running the business: *Active Directory*, *Great Plains*, *RACF*, and *SAP*. Each of these applications uses a subset of the available files. You are asked to review the following users, and revoke access to the files related to the *Great Plains* application: Edmund Johnston, Nelson Murphy, Jane Hoffman, Olive Morris." The four users in the scenario had access to 27, 21, 15, 2 files respectively, out of which 7, 5, 3, and 0 files were related to "Great Plains". Participants were expected to revoke access related to the *Great Plains* application.

Comprehension Task: In the previous tasks, we evaluated interfaces for specific scenarios, and told participants to look for a specific situation. In reality, reviewers may deal with a combination of various scenarios and need to integrate various cues to make decisions. This task aimed to evaluate the interface for situations where reviewer needs to evaluate the risk of particular access in the presence or absence of various indicators of risk and safety. We gave the participants the following scenario: "You are provided with a list of users and their accesses, and you are asked to determine how risky access to each file is. Use the knowledge you gained during the previous tasks to determine the risk associated with each file: (1) Francisco Lee, Director, R06; (2) Marcella Owens, Claims Manager, R02; (3) Margaret Estrada, Customer Service Associate, R11; (4) Alyssa Jacobs, Customer Service Manager, R09"

For each of the four user/file pairs, participants were asked to rate the risk associated with the user having access to the file using a five point likert scale (1= Very Safe, 5 = Very Risky). The order of the four likert scale questions was randomized. Four user/file pairs had different levels of risk associated with them: (1) *Marcella Owens, R02*: Access to R02 caused a separation of duties violation with R44. We expected the participants to rate the risk at 5 (High risk). (2) *Francisco Lee, R06*: Access to R06 was given to the user during his previous job. Also there was no previous review of the user's access. On the other hand, there was another user with the "Director" job title who also had access to R06. We expected participants to rate the risk at 2, 3, or 4, as this access was associated with both indicators of risk and safety. (3) *Margaret Estrada, R11*: Access to R11 was given to the user as part of her current job, the access was certified twice during past reviews, and the two other users with the same job as Margaret had the same access. We expected participants to rate the risk at 1 (High safety). (4) *Alyssa Jacobs, R09:* Access to R09 was revoked from the user during a previous review, but the user gained access again after a while. Furthermore, other users with the same job did not have access to R09. We expect participants to rate the risk at 5 (High risk).

6.2 Analysis

The goal of our analysis is to compare the three tested interfaces in terms of efficiency, and accuracy.

Efficiency: We used time-to-completion (TTC) as a metric for efficiency. To capture TTC, we automatically logged the time users spent between starting and finishing each task. Then for each task, we tested the following null hypothesis: H_o: There is no difference between the median time to completion when using any of the three interfaces. H_1: There is a difference between time to completions. We used Kruskal-Wallis test, which is a non-parametric alternative to ANOVA, since we found that the time to completion was not normally distributed, and we could not normalize the distribution using transformation. Whenever we rejected the null hypothesis, we used pairwise Wilcox test with Bonferroni adjustment to test the following three null hypotheses: (A=L) There is no difference betweeen AuthzMap and List. (A=S) There is no difference between AuthzMap and Search. (L=S) There is no difference between List and Search. For each test, we report the p value and the effect size (r). We also discuss the practical significance of the difference between AuthzMap and the other interfaces by showing the percentage of improvement or declination of median TTC over the other interfaces.

Accuracy: We identified those critical components of each task in which participants can commit dangerous errors. An error is dangerous if it puts the system in insecure state (i.e., leaves user with excessive privileges). For each critical component, we calculated the total number of participants who did and did not commit the error. Then we tested the following null hypotheses: (1) (A=L) There is no difference between the correctness of answers of AuthzMap and List participants. (2) (A=S) There is no difference between the correctness of answers of AuthzMap and Search participants. We used two-sided Fisher's exact test with Bonferroni adjustment to test the above hypotheses.

6.3 Results

In this section, we present the results of our data analysis. First, we provide a summary of participants' demographics and experience. Then we present the findings of the study. In this section, we use abbreviated condition names when presenting the results (A = AuthzMap, L = List, S = Search). Table 2 shows the number of participants who consented to the study, attempted the study, finished the study (received a return code for compensation), and those who provided valid results. If participants clicked on the consent form, we counted them as a consented participant. If a participant at least started the background questionnaire, we counted them as an attempted participant. If a participant completed all of the stages of the study, we counted them as a finished participant. Some of the finished participants skimmed through the study (our system recorded their time to completion for certain tasks at 0 seconds), or intentionally or unintentionally bypassed our system in order to get to the finish page without completing all of the tasks. We eliminated these participants, and reduced the pool of participants to a set of valid participants. We made use of data from 430 valid participants in this section.

We also tested the following null hypothesis: H_o: The validity of participants is independent from the interface they were using. To test this hypothesis, we divided the attempted participants in each condition into two groups: those who were valid participants, and those who were not valid participants. Our chi-square test revealed that the validity of the participants depends on the interface ($\chi^2(2, N = 1030) = 20.424, p = 3.7e-05, Cramer'sV = 0.141$). However the effect size is small.

We show the total time needed to complete the entire study for the valid participants in Figure 5. We tested the following null hypothesis for the time to completion of the study: H_o: The choice of the interface does not impact the total time needed for comple-

Table 2: Classification of participants according to their progress in the study

	A	L	S	Total
Consented	355	355	354	1064
Started	341	341	350	1032
Finished	190	156	151	497
Valid	174	135	121	430

tion of the study. A Kruskal-Wallis test revealed a significant effect of interface on the time to completion of the study ($\chi^2(2) = 48.033, p = 3.7e-11$). A post-hoc test using Mann-Whitney tests with Bonferroni correction showed significant differences between AuthzMap and List ($p = 4.8e-10, r = 0.31$) and between AuthzMap and Search ($p = 6.4e-07, r = 0.25$).

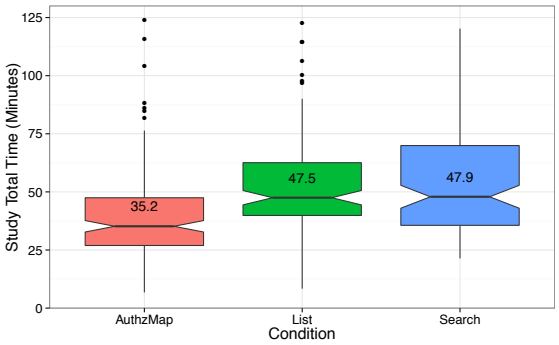

Figure 5: Total time needed to complete the study for participants in each condition

6.3.1 Participants Demographics

In the beginning of the study participants were asked to answer the background questionnaire. We show the overview of the participants' responses in Tables 3.

Table 3: Participants Demographics

		A	L	S	Total
Gender	Female	46.6%	52.6%	56.2%	51.2%
	Male	53.4%	47.4%	43.8%	48.8%
Education	Less than High School	2.3%	0.7%	0.8%	1.4%
	High School, diploma	8%	12.6%	10.7%	10.2%
	University/College Deg.	86.8%	85.9%	86.8%	86.5%
	Professional Deg.	2.9%	0.7%	1.7%	1.9%
Age	18-24 years old	30.5%	25.9%	42.1%	32.3%
	25-34 years old	43.7%	54.8%	39.7%	46.0%
	35-44 years old	15.5%	11.9%	10.7%	13.0%
	45-54 years old	6.3%	5.9%	5.8%	6.0%
	55-64 years old	3.4%	1.5%	1.7%	2.3%
	65-74 years old	0.6%	0%	0%	0.2%

6.3.2 Training

Participants were asked to complete the post-training test before proceeding to the study tasks. We summarized the number of attempts to complete the test in Figure 6. The results showed that nearly half of the participants in each condition could pass the test in the first attempt.

6.3.3 Per Task Results

In this section, we compare three conditions per task. Table 4 shows the median time to completion of individual tasks. The result of Kruskal-Wallis test for each task showed a statistically significant difference between three conditions. Therefore, we only show the result of three pairwise comparisons between conditions in Table 4.

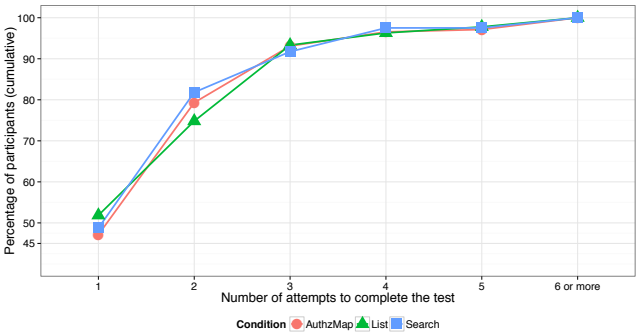

Figure 6: Number of attempts in completion of training test

Training Task: Table 4 shows that AuthzMap improved efficiency over the two other interfaces, although the effect size was medium. In terms of practical significance, AuthzMap reduced time to completion by about 20% compared to List, and by 25% compared to Search. Table 5 shows the results of the accuracy analysis. Fewer participants in AuthzMap condition committed errors in identifying the last job function of the user, compared to List condition, and in identifying the date of last access review, compared to Search condition.

Table 5: Comparing the correctness of participants' responses to the training task

	A	L	S	A=L	A=S
Job Title	97.1%	97%	98.3%	1	1
List of files	87.4%	80%	90.1%	0.443	1.000
Last Job	87.4%	54.1%	81%	<0.05	0.738
Last Review	75.9%	66.7%	35.5%	0.394	<0.05

Common Review Task: Table 4 indicates that for reviewing a single user, while the reviewer knows the files the user should have access to, Search is the fastest interface. Yet, looking at the effect size reveals that the size of the difference between AuthzMap and Search is small. In other words, AuthzMap reduces the median time-to-completion by approximately 17%, compared to list, but increases time to completion by approximately 25%, compared to Search. In this task participants could commit two dangerous errors (i.e., not revoking invalid access), and we show the proportion of participants who correctly revoked such access in Table 6. Table 6 shows that we rejected all four accuracy hypotheses, and shows that participants in AuthzMap condition had more errors than the two other conditions.

Table 6: Comparing the correctness of participants' choices in common review task.

	A	L	S	A=L	A=S
Revoked R19	70.7%	86.7%	88.4%	<0.01	<0.01
Revoked R10	70.1%	87.4%	87.6%	<0.01	<0.01

User Comparison Task: Table 4 shows that AuthzMap improves efficiency over the two other tasks. In terms of practical significance, AuthzMap decreased the time to completion by about 105%, compared to List, and by about 78%, compared to Search. The accuracy analysis (Table 7) did not reject any of the accuracy null hypotheses.

Table 7: Comparing the correctness of participants' choices in user comparison task.

	A	L	S	A=L	A=S
Revoked R13	84.5%	88.9%	86.8%	0.632	1.000

Privilege Accumulation Task: Table 4 shows that AuthzMap improves efficiency over the two other tasks. In terms of practical significance, AuthzMap improved time to completion by about 186%, compared to List, and by about 112%, compared to Search. Table 8 shows the result of accuracy tests. We rejected three of the null hypothesis for comparing AuthzMap and List, but we did not reject any of the hypotheses for comparing AuthzMap and Search.

Table 8: Comparing the correctness of participants' choices in privilege accumulation task.

	A	L	S	A=L	A=S
R06, LyndaR	86.8%	68.9%	79.3%	<0.05	0.652
R03, DerrickS	88.5%	71.9%	80.2%	<0.05	0.4
R12, DerrickS	86.2%	71.1%	81.8%	<0.05	1

SoD Violation Detection Task: Table 4 shows that AuthzMap improves the efficiency of detecting SoD violations. In terms of practical significance, AuthzMap reduced the time to completion by about 218%, compared to List, and about 165%, compared to Search.

The result of the accuracy analysis (Table 9) rejected two of the null hypothesis for comparing AuthzMap and List, but did not reject any of the hypotheses for comparing AuthzMap and Search.

Table 9: Comparing the correctness of participants' choices in SoD violation detection task.

	A	L	S	A=L	A=S
SoD (R36, R11)	92.5%	83%	91.7%	<0.05	1
SoD (R14, R00)	94.8%	85.9%	89.3%	<0.05	0.451

Application Review Task: Table 4 shows that AuthzMap and List did similarly in terms of efficiency, while Search did worse. In terms of practical significance, AuthzMap reduced the time to completion by about 35%, compared to Search. The accuracy analysis (Table 10) rejected all the null hypotheses for comparing AuthzMap and List in favor of List, and rejected one of the 15 hypotheses for comparing AuthzMap and Search in favor of Search.

Comprehension Task: Our analysis (Table 4) suggests that AuthzMap does better in terms of efficiency than the two other interfaces. It also practically improves efficiency by about 72%, compared to List, and by 89%, compared to Search. This task involved the assessment of risk for users having specific access privileges.

Table 10: Comparing the correctness of participants' choices in application review task.

	A	L	S	A=L	A=S
EdmundJ, R10	82.8%	97%	78.5%	<0.05	1
EdmundJ, R15	81.6%	96.3%	86%	<0.05	1
EdmundJ, R23	81%	97%	76.9%	<0.05	1
EdmundJ, R11	83.3%	97%	80.2%	<0.05	1
EdmundJ, R22	83.3%	97%	78.5%	<0.05	1
EdmundJ, R30	70.7%	97%	86.8%	<0.05	<0.05
EdmundJ, R28	69%	97%	78.5%	<0.05	1
NelsonM, R10	82.2%	97%	78.5%	<0.05	1
NelsonM, R15	82.8%	97%	86%	<0.05	1
NelsonM, R23	79.9%	96.3%	78.5%	<0.05	1
NelsonM, R33	69.5%	96.3%	78.5%	<0.05	1
NelsonM, R35	70.7%	95.6%	85.1%	<0.05	0.149
JaneH, R10	83.3%	96.3%	81.8%	<0.05	1
JaneH, R23	82.8%	97%	81%	<0.05	1
JaneH, R35	71.3%	95.6%	86%	<0.05	0.0909

The summary of participants' responses to risk assessment questions is presented in Figure 7. We used pair-wise two-sided fisher's exact tests with Bonferroni correction, to test the following hypothesis for each of the risk assessment: (A=L) The choice of AuthzMap or List does not impact the accuracy of risk assessment. (A=S) The choice of AuthzMap or Search does not impact the accuracy of risk assessment. The result of the test rejected ($p < 0.05$) the all four (A=L) hypotheses, and rejected ($p < 0.05$) three of the (A=S) hypotheses (in risk assessment of R02, R09, and R11).

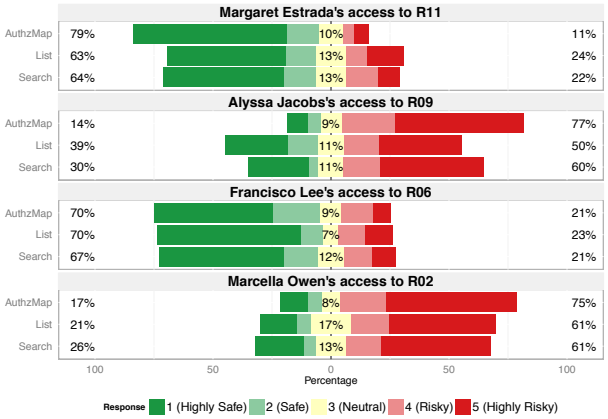

Figure 7: Summary of participants responses to comprehension questions.

Table 4: Median time to completion (TTC) for each of the tasks (in seconds), and pairwise comparison of TTCs. The highlighted cells show the cases where the null hypothesis was rejected and TTC for AuthzMap participants was lower than the other interface.

	Task	A	L	S	A = L	A = S	S=L
1	Training	192.5	243.0	259.0	$p < 0.01, r = 0.24$	$p < 0.01, r = 0.22$	-
2	Common Review	117.5	144.0	96.0	-	$p = 0.01, r = 0.10$	$p < 0.01, r = 0.14$
3	User Comparison	109.5	225.0	195.0	$p < 0.01, r = 0.45$	$p < 0.01, r = 0.37$	-
4	Privilege Accumulation	89.5	256.0	190.0	$p < 0.01, r = 0.50$	$p < 0.01, r = 0.42$	$p < 0.01, r = 0.15$
5	SoD Violation Detection	92.0	293.0	165.0	$p < 0.01, r = 0.57$	$p < 0.01, r = 0.35$	$p < 0.01, r = 0.39$
6	Application Review	181.0	185.0	280.0	-	$p < 0.01, r = 0.34$	$p < 0.01, r = 0.33$
7	Comprehension	247.5	426.0	469.0	$p < 0.01, r = 0.30$	$p < 0.01, r = 0.32$	-

7. DISCUSSION

In this section, we summarize, interpret, and discuss the findings of the user study. We first discuss the efficiency, and accuracy findings. Then, we discuss the limitations of the user study, including the use of non-expert participants, and a synthetic dataset. Finally, we discuss the larger implications of our findings.

7.1 User study findings

7.1.1 Efficiency

In Section 6.3, we show that participants in AuthzMap condition could finish the study faster than those in two other conditions. We also compared the use of three interfaces in various access review scenarios. We showed that AuthzMap improved the efficiency, compared to both of the other interfaces in five of the seven tasks, and compared to one of the interfaces in the two other remaining tasks. This finding require further discussion.

AuthzMap participants' performance in *Common Review* was not as efficient as Search, but was more efficient than List. We can provide three explanations for this: (1) The task involved reviewing only one user. The additional contextual information in the AuthzMap fisheye view could increase user's cognitive load, and hence reduce the performance. (2) The Search interface by default set the status of files to "certify". This helped the participants to change the status of two unauthorized files, and keep the rest of the files intact. Meanwhile, AuthzMap participants had to explicitly set the review status of each file. These two suspected issues provide an opportunity for further improvement. To address the first issue, we can use the focus plus context visualization [23] to highlight the user that the reviewer is currently working on, while still showing the contextual information in the background (e.g., by highlighting the current user and fading the rest of the users). The second possible issue was a design decision that we made in AuthzMap to prevent reviewers from using the default option, and rather make an explicit decision for each access privilege.

The *User Comparison* task was similar to *Common Review*, but it involved three users with identical jobs instead of one. AuthzMap participants did better than participants in two other conditions. We attribute the improvement to AuthzMap's ability to categorize, and filter users according to their job, and then use the contextual information (access of users with the same job) to quickly find the excessive access. Our analysis of study logs confirmed this, as participants used the *sort user* feature of the AuthzMap in this task significantly more than in other tasks. Furthermore, comparing the TTC of this task to the TTC of *Common Review* shows that increasing the number of participants did not impose an additional burden on AuthzMap participants, unlike Search and List participants.

In the *Privilege Accumulation*, *SoD Violation Detection*, and *Comprehension* tasks, AuthzMap performed better than the other two interfaces. We can attribute this to the visibility of context and history in the interface. For example, AuthzMap integrates the employment history and access privileges, and makes it accessible to users. The two other interfaces required participants to collect information from the HR and access review systems, and perform a mental process to formulate the relationship between access control and employment data. Additionally, AuthzMap integrated the SoD policy information with the existing access control data, and helped users quickly identify and resolve the SoD violations. Participants in two other conditions had to use an SoD catalog. Therefore, they needed to mentally associate the policy with the access control data.

In *Application Review* task, AuthzMap participants performed similar to List and better than Search. This is an expected result as both List and AuthzMap clearly integrate information about the application in the interface, but Search participants had to use the auxiliary application catalog to find the files related to a certain application.

7.1.2 Accuracy

In Section 6.3, we report that AuthzMap participants achieved more accurate results than the other two interfaces in only one task.

Accuracy results of *Common Review* task were unexpected. AuthzMap participants committed significantly more errors than participants in two other conditions. Examining the data closely shows many of these participants committed identical errors. After further investigation, we realized that AuthzMap's detail interface showed the user had accumulated privileges from a prior job. And a subset of participants who received the *Privilege Accumulation* task before *Common Review*, did not use the information in File Catalog, but rather did access review based on what they learned from *Privilege Accumulation* (about 15% of the participants in AuthzMap condition). This was our mistake in designing task data, and we should have controlled the privilege accumulation in the policy for this task. If we count the correct answers from the participants who looked at the task from privilege accumulation perspective as valid, there is no statistically significant difference between three conditions in accuracy.

Accuracy results for "User Comparison" task do not show a difference between three conditions. These results suggest the increase in efficiency did not impact the accuracy of participants in AuthzMap condition.

For two tasks that required decision making in uncertain conditions, *Privilege Accumulation* and *SoD Violation Detection*, AuthzMap positively affected accuracy, compared to List but not Search. These two tasks required contextual information that unlike AuthzMap was not integrated in List and Search interfaces. Search participants did surprisingly well in the collection and integration of the context with the information available in the interface but not the Lits participants. One explanation for this observation is that the List interface contains redundant information that could mentally overload users. On the other hand, Search is rather straightforward, and while it requires user to spend more time collecting and integrating information, it does not reduce the accuracy.

Accuracy results for "Application" task were rather surprising. AuthzMap and Search participants produced less accurate results than List. While we expected the List participants to do better than Search (List clearly showed the application associated with each access privileges, as one of the columns in the list of privileges), we expected AuthzMap to perform as good as List. Further looking at the participants errors, we did not find any patterns or evidence that participants committed mistakes rather than slips. There are three possible explanations of such slips: (1) The names of the applications were presented in a small text, and it was rotated 90 degrees. Prior research shows that text rotation can have a negative impact on human cognition, and requires mental rotation, before a human can recognize an object [18]. To address this, we can use slightly less rotated text (e.g., 45 degrees), as it is shown that rotation is positively correlated with cognitive load. (2) Complexity of the grid: to complete the task, users had to recognize the file related to an application (located in columns of the grid), and then check the target user for having access to the file. This process can be prone to errors due to the proximity of grid cells. To address this, we can utilize the focus plus context visualization [23], by allowing users click on the column to focus on a specific file.

Unlike other tasks, AuthzMap participants provided more accurate responses to three of the four questions in the *Comprehension* task. We expected this result, as participants in other conditions

should have used multiple information sources to complete the task, and they needed to build the correct model of the policy in their memory. Yet, AuthzMap participants could see the complete picture of the policy. The only question that we did not see a significant difference between AuthzMap and both conditions was the assessment of R06 risk, which we did not see a difference between AuthzMap and Search. R06 could be both safe or unsafe, therefore, we expect participants not to choose highly safe or highly risky. Further look at the graphs in Figure 7 shows that Search participants assessed R06 and R11 (which was highly safe) similarly. However, AuthzMap participants assessed R06 rather differently from how they assessed R11. This suggests that maybe Search participants naively chose unsure responses, but AuthzMap participants made a more informed choice.

7.1.3 User Study Limitations

Ideally we would have evaluated AuthzMap by asking managers to use AuthzMap to review access of actual users in their company. But our experience from this study, and our past field studies [3] suggests that conducting a field experiments in real organizations is extremely difficult. In this study, we faced challenges similar to those discussed in [31]. First, AuthzMap is a prototype, and integrating it with real access management systems in organizations is a software engineering challenge. Second, asking managers to budget time for evaluating AuthzMap is challenging, particularly because access review is not their day-to-day task. Third, AuthzMap requires identity and access control data, which are commonly considered extremely sensitive. Our experience shows that even getting permission to conduct an interview requires approval from the legal department of a large company, as well as multiple managers, let alone conducting experiments using the sensitive data.

Due to the above challenges, we adopted an approach similar to [31], and conducted a set of during-design, exploratory studies before committing to a costly field study. First, we received feedback on AuthzMap from a large domain expert audience (employees of our industry partner). We also had two small group discussions with the engineering team, and usability team of our industry partner. Second, we conducted 12 heuristic evaluation sessions (using Nielsen [25] and ITSM [17] heuristics) with independent usability experts to identify usability issues with AuthzMap, and further improved the interface. Third, we conducted a lab study (Section 6) with non-domain experts to further evaluate the interface and compare it to existing systems. Sedlmair et al. [31] showed that conducting during-design experiments could be very helpful and lead to tools with higher usability, and eventually become a major reason for the tool being deployed in the field. Therefore, we conducted an exploratory study with MTurk participants to be confident that the tool does not have obvious usability problems, and fares well against existing systems.

The next step in evaluation would be to conduct an in-depth long-term case study [32] in an organization, by integrating AuthzMap with existing access management systems, asking managers to use AuthzMap, and then get qualitative feedback on the impact of AuthzMap. Such a field study can show if the tool will be adopted by managers, and could increase the effectiveness of conducting access reviews.

We used an automatically generated dataset. Using a real-world dataset was not feasible, as there are very few real-world enterprise access control data sets available to the research community. We examined five common datasets used regularly by access control community such as: americas_small, apj, healthcare, domino, firewall1 and firewall2 [11]. These datasets only contained lists of users, permissions, and user-to-permission assignments. Our study required contextual data, such as users' job, employment history, access history, and review history. Adding meaningful context to existing datasets was not possible, therefore, we elect to generate a dataset that best matched our interview study findings.

7.2 Implications Beyond Access Review

Our field study findings have larger implications than just understanding access review activity. Our findings suggests that while access control policies are usually composed of users, roles, and permissions, these three components are only parts of a larger context, and they evolve and change over time. Therefore, a snapshot of a user's access privileges does not provide a complete picture of access policy. We further determined the context of a users' access privileges, which includes other users' access, other policies that impact such access (such as SoD policies), user's job, and other stakeholders involved in the access control decisions, such as those who requested or approved the user's access. We also demonstrated that access control policies evolve over time, and identified users' job, access privileges, and previous reviews as important historical artifacts. Although our focus was on access control in large organizations, the concept of context for access control policies is still applicable to access control in other domains such as file systems, multimedia, etc. We should note that each domain should be studied separately, as the contextual information for enterprise domain (such as job or approval workflow) may not be applicable in other domains. For example, findings by Vaniea et al. [36] suggest that proximity of access control displays and photos helps users notice and correct access control errors. In this case, the photo (visual representation of the asset) is a part of the access control context.

The design of AuthzMap can serve as an example of how the contextual information can be integrated with access policy in a user interface, and our user study suggested that the design was successful. Furthermore, such integration will improve efficiency of accessing contextual information, and in complex decision making processes (such as *Comprehension* task in our study) can improve better understanding of policy, and therefore, facilitate making more accurate decisions. Our study results also suggest that showing context could increase the complexity of the interface and in few occasions could negatively impact the accuracy or efficiency. Therefore, we suggest improvements such as focus plus context visualization [23] to alleviate those conditions.

8. CONCLUSION

In this paper, we studied how access policies are reviewed in large organizations. We then identified a set of five challenges that organizations face during access review, and suggested four design goals to deal with those challenges. We then realized the design goals by building AuthzMap, a novel user interface for reviwing and making sense of access policies in organizations. We then conducted an exploratory user study with 340 MTurk participants to compare the use of AuthzMap to two of the existing access review systems. Our results show that AuthzMap improved efficiency of access review in five of the seven, and accuracy in one of the seven tasks. Our goal for designing AuthzMap was to address five challenges identified during the field study, and our results show that for those tasks that involve identified challenges, AuthzMap improved the efficiency, and in one task accuracy. The bigger HCI implications of this work are exploring the importance of context in access control, and proposing an effective approach for integrating contextual information in access control interfaces. As the next step, AuthzMap should be deployed in a real organizational setting, and its impact should be evaluated in a field study.

9. REFERENCES

[1] L. Bauer, L. F. Cranor, R. W. Reeder, M. K. Reiter, and K. Vaniea. Real life challenges in access-control management. In *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 899–908, New York, NY, USA, 2009. ACM.

[2] M. Beckerle and L. A. Martucci. Formal definitions for usable access control rule sets from goals to metrics. In *Proceedings of the Ninth Symposium on Usable Privacy and Security*, SOUPS '13, pages 2:1–2:11, New York, NY, USA, 2013. ACM.

[3] D. Botta, R. Werlinger, A. Gagné, K. Beznosov, L. Iverson, S. Fels, and B. Fisher. Towards understanding IT security professionals and their tools. In *Proc. of Symp. On Usable Privacy and Security (SOUPS)*, pages 100–111, Pittsburgh, PA, July 18-20 2007.

[4] C. Brodie, C.-M. Karat, J. Karat, and J. Feng. Usable security and privacy: a case study of developing privacy management tools. In *SOUPS '05: Proceedings of the 2005 symposium on Usable privacy and security*, pages 35–43, New York, NY, USA, 2005. ACM.

[5] J. M. Carroll, P. L. Smith-Kerker, J. R. Ford, and S. A. Mazur-Rimetz. The minimal manual. *Human–Computer Interaction*, 3(2):123–153, 1987.

[6] Centers for Medicare & Medicaid Services. The Health Insurance Portability and Accountability Act of 1996 (HIPAA). Online at http://www.cms.hhs.gov/hipaa/, 1996.

[7] K. Charmaz. *Constructing Grounded Theory*. SAGE publications, 2006.

[8] J. Considine, M. Botti, and S. Thomas. Design, format, validity and reliability of multiple choice questions for use in nursing research and education. *Collegian*, 12(1):19 – 24, 2005.

[9] G. Convertino, H. M. Mentis, A. Slavkovic, M. B. Rosson, and J. M. Carroll. Supporting common ground and awareness in emergency management planning: A design research project. *ACM Trans. Comput.-Hum. Interact.*, 18(4):22:1–22:34, Dec. 2011.

[10] A. Cser. The forrester wave: Role management and access recertification, q3 2011. Technical report, Forrester Research, inc., August 2011.

[11] A. Ene, W. Horne, N. Milosavljevic, P. Rao, R. Schreiber, and R. E. Tarjan. Fast exact and heuristic methods for role minimization problems. In *SACMAT '08: Proceedings of the 13th ACM symposium on Access control models and technologies*, pages 1–10, New York, NY, USA, 2008. ACM.

[12] Y. Engeström. Activity theory and individual and social transformation. *Perspectives on activity theory*, pages 19–38, 1999.

[13] A. Forget, S. Chiasson, and R. Biddle. Supporting learning of an unfamiliar authentication scheme. In *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, volume 2012, pages 1002–1011, 2012.

[14] J. Hollan, E. Hutchins, and D. Kirsh. Distributed cognition: toward a new foundation for human-computer interaction research. *ACM Trans. Comput.-Hum. Interact.*, 7(2):174–196, 2000.

[15] P. Inglesant, M. A. Sasse, D. Chadwick, and L. L. Shi. Expressions of expertness: the virtuous circle of natural language for access control policy specification. In *SOUPS '08: Proceedings of the 4th symposium on Usable privacy and security*, pages 77–88, New York, NY, USA, 2008. ACM.

[16] P. Jaferian and K. Beznosov. Access review survey report. Technical Report LERSSE-REPORT-2014-001, Laboratory for Education and Research in Secure Systems Engineering, University of British Columbia, May 2014.

[17] P. Jaferian, K. Hawkey, A. Sotirakopoulos, M. Velez-Rojas, and K. Beznosov. Heuristics for evaluating it security management tools. *Human–Computer Interaction*, 29(4):1–40, 2013.

[18] P. Jolicoeur. The time to name disoriented natural objects. *Memory & Cognition*, 13(4):289–303, 1985.

[19] V. Kaptelinin and B. Nardi. *Acting with technology: Activity theory and interaction design*. MIT Press, 2006.

[20] V. Kaptelinin, B. A. Nardi, and C. Macaulay. Methods & tools: The activity checklist: a tool for representing the space of context. *interactions*, 6(4):27–39, July 1999.

[21] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 453–456, New York, NY, USA, 2008. ACM.

[22] B. W. Lampson. Protection. In *5th Princeton Conference on Information Sciences and Systems*, page 437, New York, NY, USA, 1971. ACM Press.

[23] Y. K. Leung and M. D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Trans. Comput.-Hum. Interact.*, 1(2):126–160, June 1994.

[24] M. McCloskey. Turn user goals into task scenarios for usability testing, January 2014.

[25] J. Nielsen. Finding usability problems through heuristic evaluation. In *Proc. CHI '92*, pages 373–380, New York, NY, USA, 1992. ACM.

[26] J. Nielsen. Usability 101: Introduction to usability. http://www.nngroup.com/articles/usability-101-introduction-to-usability/, January 2012.

[27] D. A. Norman. *The Psychology of Everyday Things*. Basic Books, 1988.

[28] D. Povey. Optimistic security: A new access control paradigm. In *Proceedings of the 1999 Workshop on New Security Paradigms*, NSPW '99, pages 40–45, New York, NY, USA, 2000. ACM.

[29] R. W. Reeder, L. Bauer, L. F. Cranor, M. K. Reiter, K. Bacon, K. How, and H. Strong. Expandable grids for visualizing and authoring computer security policies. In *Proc. CHI '08*, pages 1473–1482, New York, NY, USA, 2008. ACM.

[30] R. W. Reeder, L. Bauer, L. F. Cranor, M. K. Reiter, and K. Vaniea. More than skin deep: measuring effects of the underlying model on access-control system usability. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, CHI '11, pages 2065–2074, New York, NY, USA, 2011. ACM.

[31] M. Sedlmair, P. Isenberg, D. Baur, and A. Butz. Information visualization evaluation in large companies: Challenges, experiences and recommendations. *Information Visualization*, 10(3):248–266, July 2011.

[32] B. Shneiderman and C. Plaisant. Strategies for evaluating information visualization tools: Multi-dimensional in-depth long-term case studies. In *Proceedings of the 2006 AVI Workshop on BEyond Time and Errors: Novel Evaluation Methods for Information Visualization*, BELIV '06, pages 1–7, New York, NY, USA, 2006. ACM.

[33] D. K. Smetters and N. Good. How users use access control. In *SOUPS '09: Proceedings of the 5th Symposium on Usable Privacy and Security*, pages 1–12, New York, NY, USA, 2009. ACM.

[34] T. Swensen. Wikileaked! wikileaked! what we can all learn from the bradley manning debacle. Novell connections magazine, April 2011.

[35] Unknown. Sarbanes-Oxley Act of 2002. Online Document, July 2002.

[36] K. Vaniea, L. Bauer, L. F. Cranor, and M. K. Reiter. Out of sight, out of mind: Effects of displaying access-control information near the item it controls. In *Proceedings of the 2012 Tenth Annual International Conference on Privacy, Security and Trust (PST)*, PST '12, pages 128–136, Washington, DC, USA, 2012. IEEE Computer Society.

[37] K. Vaniea, L. Bauer, L. F. Cranor, and M. K. Reiter. Studying access-control usability in the lab: lessons learned from four studies. In *Proceedings of the 2012 Workshop on Learning from Authoritative Security Experiment Results*, LASER '12, pages 31–40, New York, NY, USA, 2012. ACM.

[38] Y. Wang, G. Norcie, S. Komanduri, A. Acquisti, P. G. Leon, and L. F. Cranor. 'i regretted the minute i pressed share': a qualitative study of regrets on facebook. In *Proceedings of the Seventh Symposium on Usable Privacy and Security*, SOUPS '11, pages 10:1–10:16, New York, NY, USA, 2011. ACM.

APPENDIX

A. INTERVIEW GUIDE

A.1 Organizational context

A.1.1 General Information about Interviewee and Organization

- What is your position?
- Background: What is your IT/Security education/path?
- Can you briefly describe your organization? (size, sector)
- Describe security management within your organization
  - Who is responsible for security within your organization?
  - What is the security management model (centralized, distributed, etc.)? (With little help to the person)
- Can you describe the security policies in your organization (also probe for participant's role)
  - What formal (official, written) security guidelines/ policies/ architectures/ models are in place?
  - What is done in practice? (To see if the policy is completely enforced)
  - What is the process for developing policies?
  - How are policies communicated?
- To whom are policies communicated?
  - How are security-related policies enforced?
- What security risks/challenges do you perceive to be important for your organization?
  - What are the security risks or challenges in your organization?
  - What security incidents has your organization experienced as a result of these risks/challenges?
  - To what extent these incidents relate to access and identity management?
  - Are there security incidents or risks that are least priority?

A.1.2 Activities

- What are your responsibilities within the organization? (get overall, lead into security specific activities)
  - Actual duties/ official duties (Let them talk, probe anything not on list to confirm that omissions are true negatives)
- Manage identities and accesses
- Perform and respond to security audits on the IT infrastructure?
- Develop security policies?
- Design and revise security services or projects?
- Implement security controls?
- Solve end user security issues?
- Educate and train?
- Respond to security incidents? (Skills, knowledge and strategies, resources (tools) used)
- Mitigate new security vulnerabilities?
- Prioritization (typical day)

A.2 Questions about Access and Identify Management (AIM) Process

A.2.1 AIM process (general)

- What do you consider to fall under the definition of access and identity management?
- What is the current process within your organization?
  - Activities? (policies, managing access, managing identities, audit, compliance, trouble shooting)
- Stakeholders? (management, HR, IT, security, employees, customers, external organizations...)
- What is your role?
- Knowledge required
- Importance?
- Frequency?
- Is it supported by tools?
- Can it be automated or supported better by the tool?
- How was this process before adopting an IdM solution ?

A.2.2 Compliance

- Is the organization required to comply with any standard? Which standard?
- What is the role of IDM solution in your compliance with the standard?

A.3 Probing specific activities (depends on their role)

A.3.1 Managing accesses and identities

- Can you describe the lifecycle for managing accesses and identities? (From creation to destruction of an identity)
- Which parts of this lifecycle is supported by your IdM system?
- How you manage changes in user status? (extending access for a user, changing access, discontinuing access)
- How frequently you face exceptions in setting up accesses and how you handle them? (For example: Employees should normally access X but not Y. But for a specific case you should temporarily provide access to an employee to Y.)
- How complex are the policies and how do you handle complexity?
  - Number of users? Number of resources? Number of roles? Number of access rules (E.g. Role X has access Y to resource Z)
- Are there any cases that you don't want system access to be controlled by your IDM solution?

A.3.2 Entitlements

- Can you give us a definition for entitlement ? Can you give us examples from your organization?
- How entitlements are managed in your organization ? Is there a process in place?
- What stakeholders are involved in determining the meaning of an entitlement and deciding about associating entitlements to users ?
- What is the process of checking if users are assigned to a correct set of entitlements?

A.3.3 Audit

- How can you make sure that the correct access rights are set for the intended person? (that the policy is implemented correctly)
- What is the process for identifying and removing the unused or discontinued identities and accesses?
- Do you have any formal audit procedure in place? If so, describe?
- Is there any legislation that require your organization to perform audit ?

A.3.4 Role Management

- How do you create roles in your organization? (define business responsibilities as roles and association of roles to entitlements?
- How frequently roles are changed or added?
- How do you perform "role engineering" in your system?
  - What is difficult/easy about it?
  - What approach do you use (top down, bottom up, hybrid)?
- What stakeholders are involved in the process of managing roles?
- What tools do you use for managing roles?

A.3.5 End-user experience

- What are the ways of accessing the system for users? Is there just one, or many (different usernames, different portals, etc)?
- Can you recall any end-user complaints relating to the IdM solution?
- Is it possible for users to manage access?
- How do the end users understand the configuration implemented by security practitioners? How can an end-user know which resources he has access to?
  - Does the tool give feedback?
  - Do you need to provide explicit knowledge? (For example about how they can find-out their access rights, changing their personal information (password, etc.))?
  - Do end-users need to be aware of their access rights or policy at all?
- Do you think the end-user experience has changed after adoption of IdM system ?

A.3.6 Troubleshooting

- How frequently you deal with problems that require troubleshooting?
- Can you give an example? (get details: collaboration?, blow-by-blow account)
- While performing troubleshooting, what is the magnitude of information that you work with? (means logs about accesses) Do you cut things or prioritize because of the volume of information?

A.3.7 Archiving

- What kind of activities/incidents/interactions/communications do you document and how?
- Is there a need for recording/archiving of communications? In what circumstances?

A.3.8 Reporting

- Describe the reports that you generate that are related to access and identity management.
- For whom do you generate these reports?
- How are your reports used?
- What tools do you use to help compose and send your IdM reports?
- Do you generate reports for different people? Who?
- If you compose different kinds of reports (different content, different level of granularity) for different people, is it easy for you to compose different kinds?
- What makes it easy or tedious?
- Do any of your report help you prioritize? What information helps? Where does it come from?

A.4 Questions about Access and Identify Management (AIM) Technologies

- What is your definition of an ideal IdM solution? (Solution that manage accesses, control digital identities, enable checking who did what and who granted the access, checking the compliance of the system)
- Do you currently have such solution?
- Which parts exist in your current infrastructure?
- What are the driving forces for adopting IdM technology in your organization ?

A.4.1 Purchasing/Evaluation

- What was the process for selecting the IdM tool in use?
  - What stakeholders are involved in the process?
  - How did you evaluate the competing tools?
- What features do you look for in a tool? Which features are available in your current tools?
- What properties to you wish for in your tools? (quality, user interface, performance, service, vendor reputation).

A.4.2 Tool deployment

- What are the pre-requisites for deploying an IdM solution? I mean should any specific business processes in place? Should any technological infrastructure be in place? Is there any training required? Is there any kind of knowledge required?
- Who are the people involved in the IdM deployment? I mean is there any relation for example with managers, end-users, or external organizations?
- What are the difficulties in deployment of the product?
- Do you need to customize out of the box identity and access management tools to meet your needs? If yes, can you describe the process for that?
- Do you need to integrate any of your existing systems (Databases, Terminals, Web Applications, etc.) with your IdM solution? Does the solution perform this automatically?
- Do you have any recommendations for improving deployment process?

A.4.3 Tool maintenance

- What maintenance tasks do you perform to keep the IdM solution running and who is responsible for them?
- How much technical knowledge and effort do they need to maintain the solution?
- What is the process of updating or changing your IdM solution?

A.4.4 Tool Use

- How do you use tool X and what do you like/dislike about it? (if possible, get them to show the interface and probe their view of the functionality/usability afforded by the tool. Try to take photos or draw sketches from what the show.)
- In addition to tools that are part of your general IdM infrastructure, are there any other tools used for the various IdM activities? (i.e., excel sheet for creating reports related to IdM)
- Are there any tools do you no longer use? (why?)
- What is the most error prone part of your identity management solution?
  - How do you find out that a tool has made an error?
  - What do you do to recover from errors?

A.5 Working/Dealing with other stakeholders

A.5.1 Collaboration

- With whom do you interact during IdM activities? What are the circumstances?
- Do you need to Co-ordinate your work with other people?
  - Do you need to delegate some part of an IdM task to other people? Do you need to work with other people in order to accomplish an IdM task?
- What is your relationship with other people who are responsible for identity management? How closely do you work with them?
- Do the people who manage accesses or identities have knowledge about computer security? Do they know whether or not risks are involved in what they do? Do they understand these risks?
  - Tools to facilitate awareness: Do you use any tools to support awareness of activities of others (workflows, shared calendars, shared to-do lists, whiteboards)
  - Does the IdM tool provide any support for activities which require collaboration?

A.5.2 Communication and Common ground (negotiating a shared understanding?

- What type of information do you need to share?
- Are there new issues that arise through your on-going experience with IdM which are necessary to communicate to others?
  - How are they communicated? (Can give example of Documents, Wikis, or SharePoint)
  - Is your IdM tool integrated with any of these communication channels?
  - Do you use specific terminology to communicate with other people involved in IdM activities?
- How do they know that the information and your communication is understood?
- How people understand each other while communicating and how they make sure and let each other know that they understood each other?
- Can you give us an example of misunderstanding during communication with other stakeholders about IdM?
- When is it necessary to interact with people outside of the organization?

B. DETAILED DESCRIPTION OF AUTHZMAP, LIST, AND SEARCH

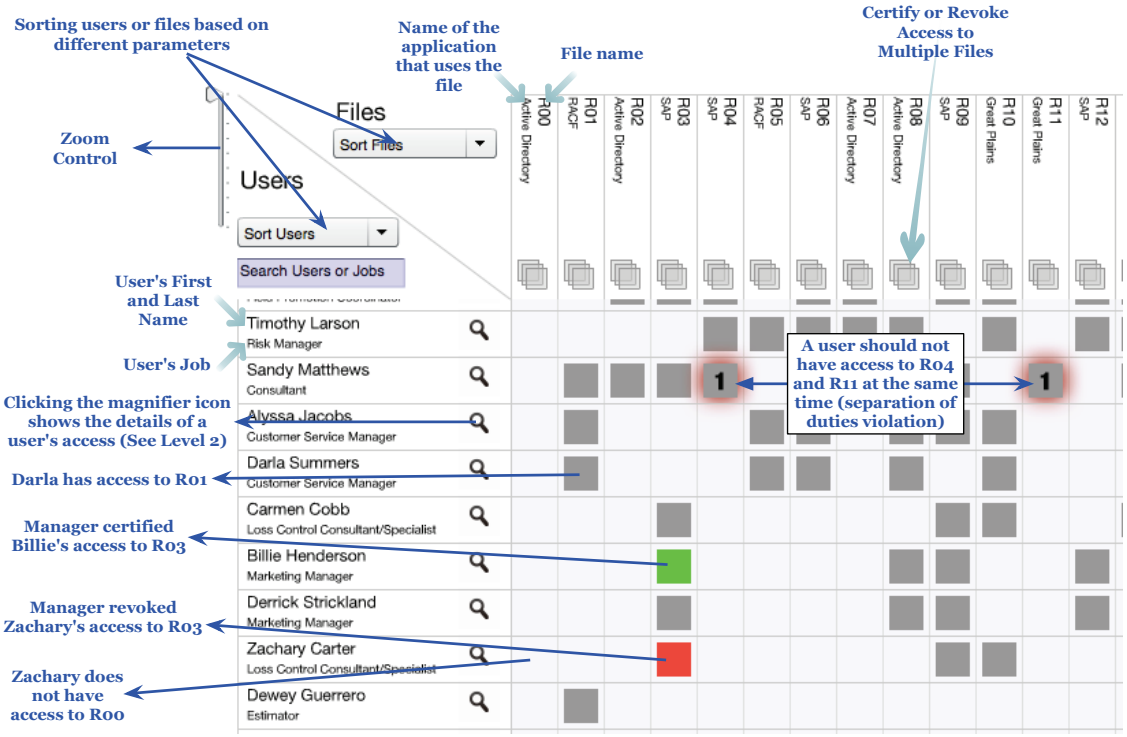

Figure 8: Level one of the AuthzMap interface. We used the notion of files in the user study, but eventually columns in the grid indicate roles, permissions, files, or any other type of entitlements.

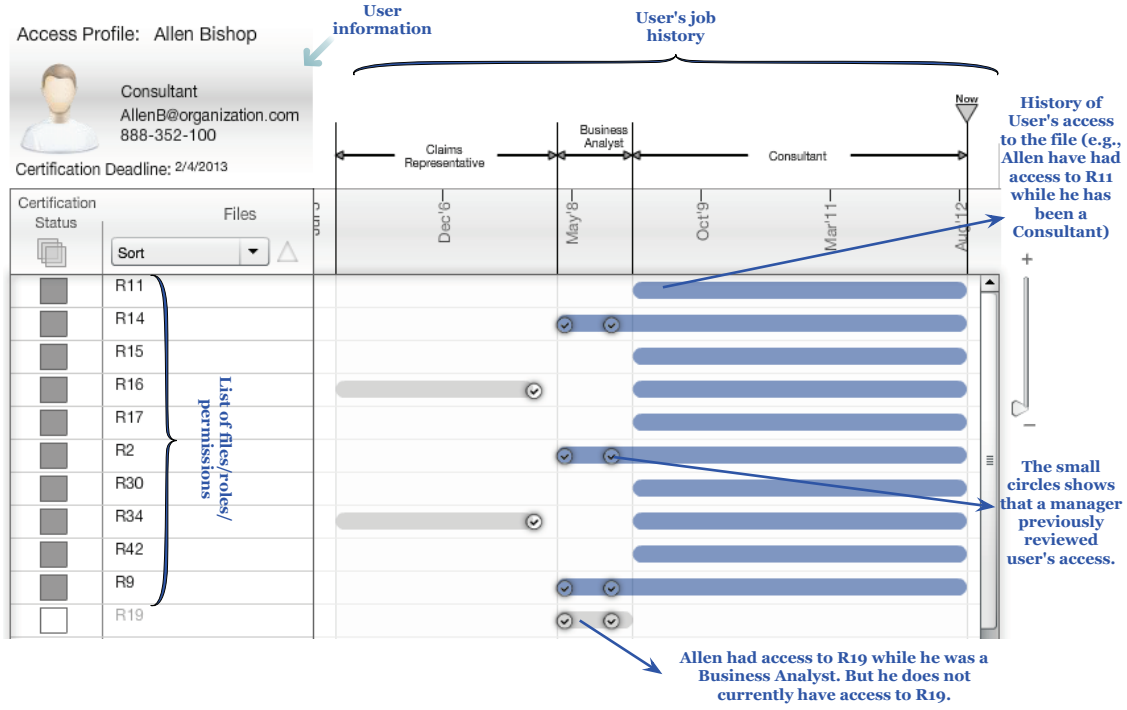

Figure 9: Level two of the AuthzMap interface. Reviewer can access this level by clicking on the magnifier icon in the level 1 of the interface.

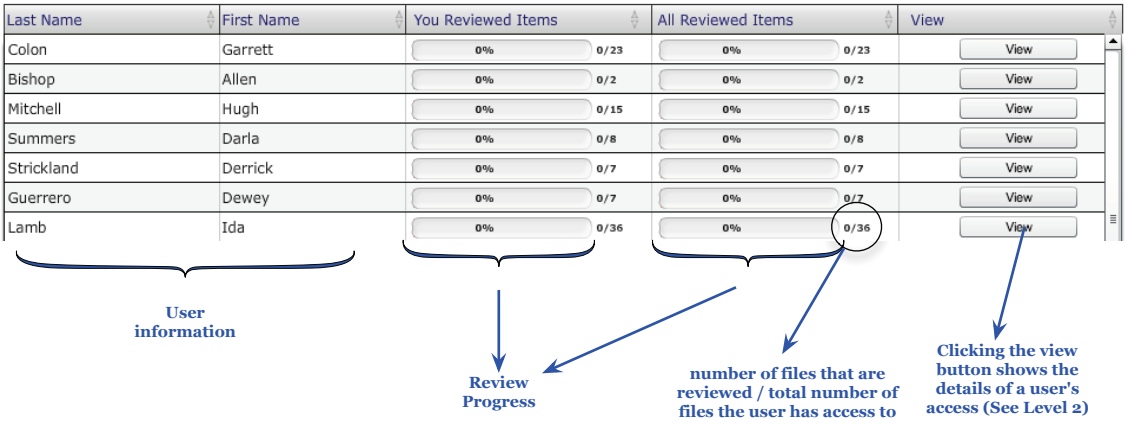

Figure 10: Level one of the List interface. The original interface used the notion of "entitlements", but we changed it to files for the purpose of the user study.

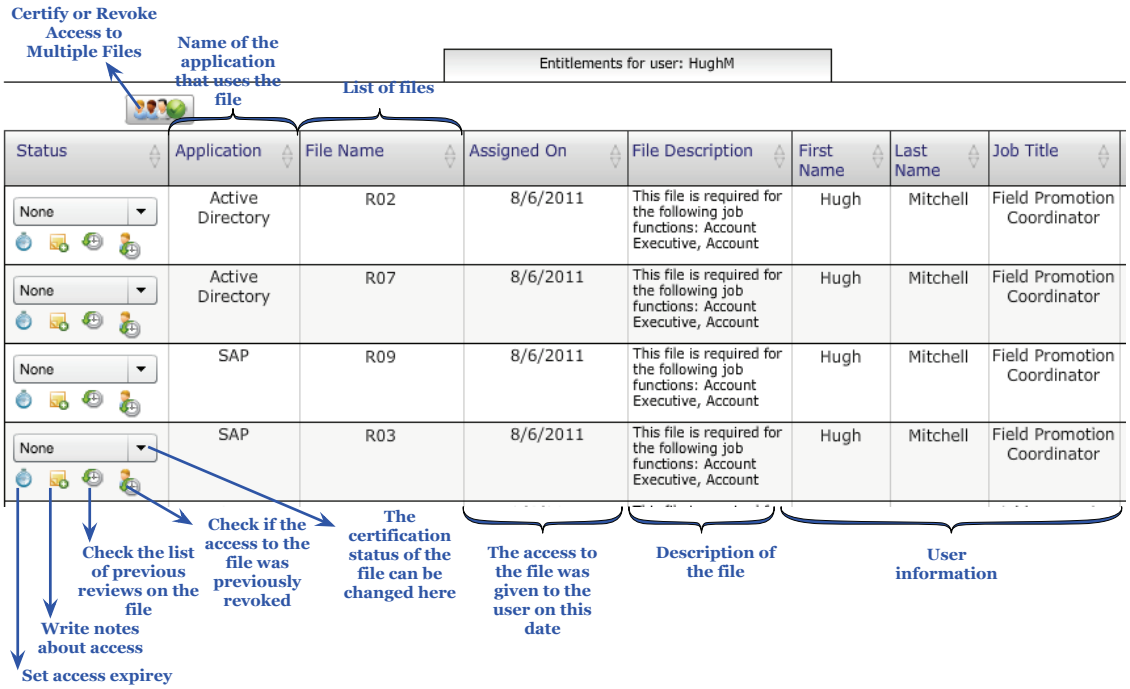

Figure 11: Level two of the List interface.

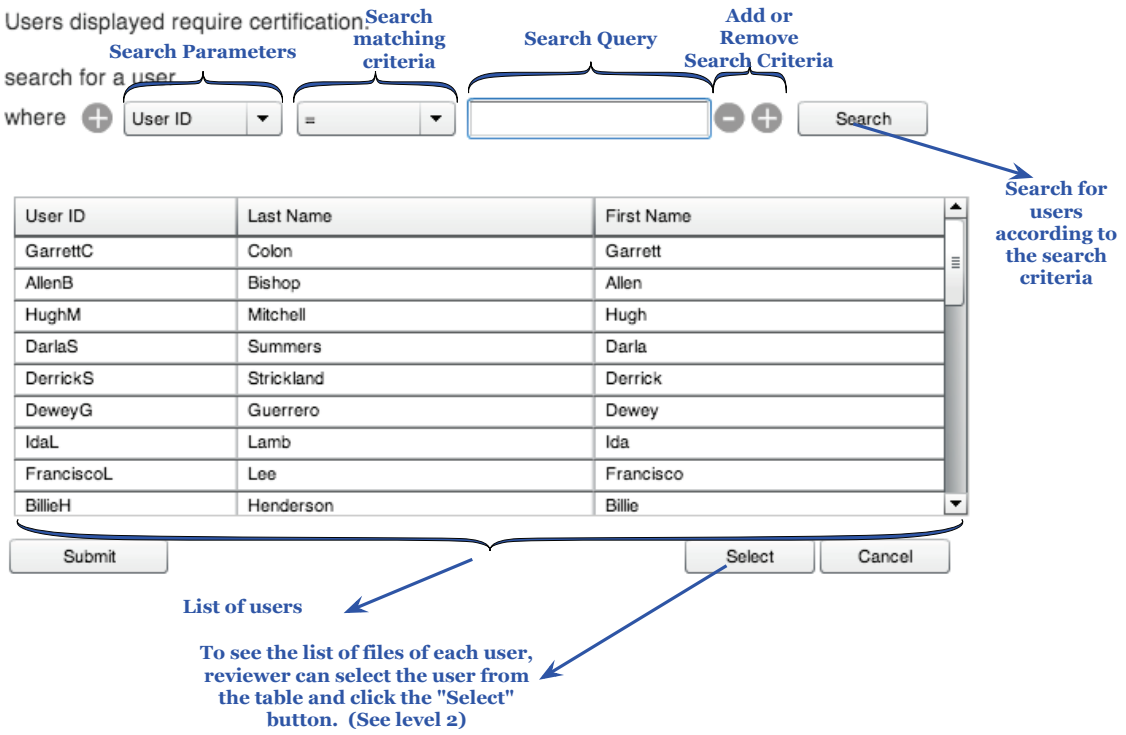

Figure 12: Level one of the Search interface. The original interface used the notion of "Roles", but we changed it to files for the purpose of the user study.

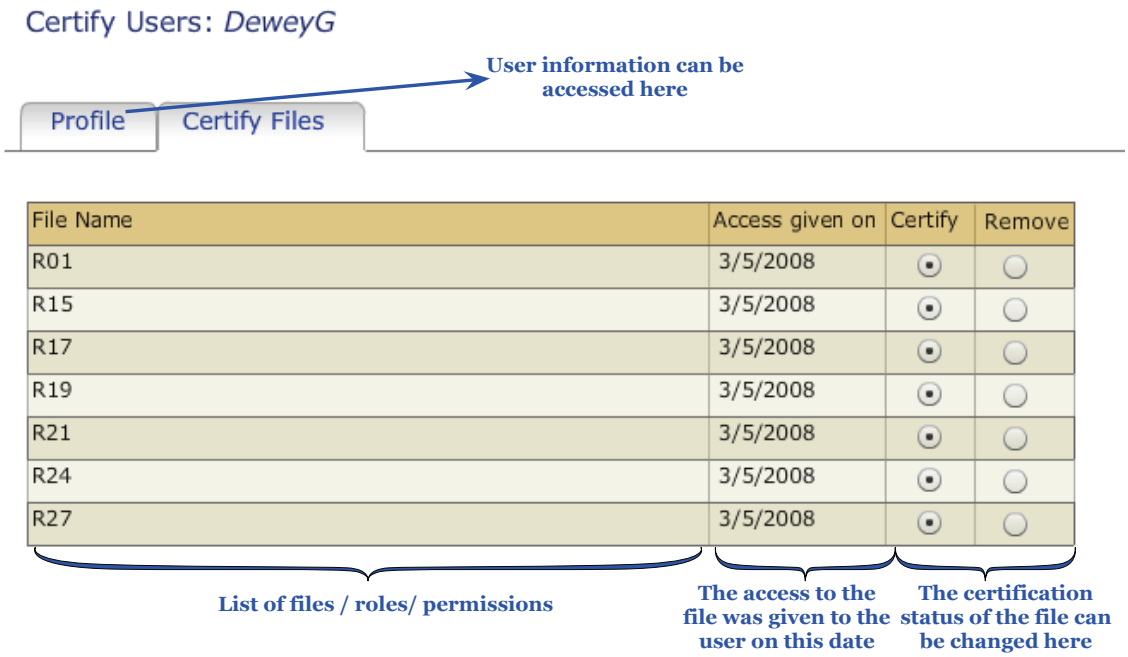

Figure 13: Level two of the Search interface.