Heuristics for Evaluating IT Security Management Tools

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

Kirstie Hawkey
Dalhousie University
Halifax, Canada, B3H 1W5
hawkey@cs.dal.ca

Andreas Sotirakopoulos
University of British Columbia
Vancouver, Canada, V6T 1Z4
andreass@ece.ubc.ca

Maria Velez-Rojas
CA Technologies
San Jose, California
maria.velez-rojas@ca.com

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

ABSTRACT

The usability of IT security management (ITSM) tools is hard to evaluate by regular methods, making heuristic evaluation attractive. However, standard usability heuristics are hard to apply as IT security management occurs within a complex and collaborative context that involves diverse stakeholders. We propose a set of ITSM usability heuristics that are based on activity theory, are supported by prior research, and consider the complex and cooperative nature of security management. In a between-subjects study, we compared the employment of the ITSM and Nielsen's heuristics for evaluation of a commercial identity management system. Participants who used the ITSM set found more problems categorized as severe than those who used Nielsen's. As evaluators identified different types of problems with the two sets of heuristics, we recommend employing both the ITSM and Nielsen's heuristics during evaluation of ITSM tools.

Categories and Subject Descriptors

H.5.3 [**Group and Organization Interfaces**]: Evaluation / methodology

General Terms

Security, Human Factors

Keywords

Heuristic evaluation, IT security management, computer supported cooperative work, complex systems

1. INTRODUCTION

Information technology security management (ITSM) tools serve several purposes including protection (e.g., network, system, and data), detection (e.g., tools for threat and vulnerability management), and user management (e.g., tools for identity and access management) [4]. Recent research [6, 57, 14] has highlighted the need to understand how ITSM tools support collaboration and information sharing between IT security practitioners (SPs); IT administrators; and other stakeholders, such as managers and end-users. Werlinger et al. [57] identified nine security activities that require collaborative interactions and developed a model of the complexity of their interactions. This complexity arises from organizational attributes (e.g., distribution of IT management); the need for SPs to interact with multiple stakeholders with different perceptions of risk and levels of security training; and their need to engage in multiple security related activities. Each of these activities may require different tacit knowledge and kinds of information to be conveyed.

Evaluating the usability of specific ITSM tools is challenging. Laboratory experiments may have little validity due to the complexity of real-world security problems and the need to situate a specific tool within a larger context [33]. However, it is difficult to recruit SPs for simple interviews, let alone field observations [6, 26]. Direct observation of tool use can be time consuming as much security work is spontaneous (e.g., security incident response), or occurs over many months (e.g., deploying an identity management system). As ITSM tool use is intrinsically cooperative, its study inherits the difficulties of studying cooperation [33]. Therefore, heuristic evaluation of ITSM tools could be a viable component of tool usability evaluation.

The goal of our research is to develop and evaluate a new set of heuristics for evaluating ITSM tools. The focus of our heuristics is on finding problems that hinder the use of tools in those ITSM activities that are distributed over time and space, involve collaboration between different stakeholders, and require knowledge to deal with the complexity.

In this paper, we propose heuristics grounded in prior usability research for IT and ITSM tools, and supported by post-cognitivist theories [24]. We then report an empirical evaluation of our heuristics in which we compared their usage to Nielsen's. We conducted a between-subjects study with 28 participants and examined different aspects of evaluation when deploying the two sets of heuristics. Our results suggest that the number of major problems that are found using the ITSM heuristics is higher than the number of problems that are found using Nielsen's. Our results also show that evaluators using ITSM heuristics tend to find more severe problems compared to those using Nielsen's. Furthermore, our results show that the evaluation of the IdM system

Symposium on Usable Privacy and Security (SOUPS) 2011, July 20–22, 2011, Pittsburgh, PA USA

requires more evaluators compared to evaluations performed by Nielsen on simple user interfaces; we observed few overlaps between problems identified by individual evaluators using either Nielsen's or the ITSM heuristics. Due to the low degree of overlap between the two sets, we recommend that heuristic evaluations of ITSM tools employ both ITSM and Nielsen's.

The remainder of this paper is organized as follows. We provide some background information on heuristic evaluation method, and review the prior research on developing new heuristics in Section 2. In Section 3, we explain the process of developing the ITSM heuristics, describe the heuristics, and provide the theoretical background for them. We then describe the methodology for the comparative evaluation of our heuristics in Section 4, before laying out its results in Section 5. In Section 6, we discuss our observations in performing heuristic evaluation with Nielsen and ITSM heuristics, and then describe future work. We conclude with an overview of our results and recommendations.

2. BACKGROUND AND RELATED WORK

2.1 Activity Theory

One of the dominant theoretical foundations for HCI has been information processing psychology [24]. This theory focuses on human actions as the units of analysis. This approach to HCI has been criticized by many researchers as it doesn't take into account the context in which users' actions are situated. Activity theory has been proposed as an alternative theory. Kuutti [27] discusses four key aspects of activity theory. First, activity theory moves the unit of analysis beyond user actions, and proposes "Human Activity" as the unit of analysis, which includes the context in which user's actions are situated. Second, every activity has a history of its own. This history is often embedded in the activity, and historical analysis is required to understand the activity. Third, relationships between components of the activity are mediated by artifacts. Fourth, activities are realized as individual or cooperative actions. Engeström [10] proposed the formulation of activity theory shown in Figure 1. He suggests every activity has a subject that performs the activity, and an object toward which the activity is directed. Based on this model, a subject manipulates an object using a set of artifacts (e.g., tools, documents, procedures). Furthermore, an activity involves other stakeholders and a set of rules and norms, which governs the activity. As prior research shows that social and organizational factors impact ITSM activities, activity theory may be useful when describing the ITSM context.

2.2 Heuristic Evaluation

Heuristic evaluation is a type of informal or discount usability evaluation method [36]. As opposed to empirical usability evaluation, informal usability evaluation does not involve real users, and requires less time and a smaller budget. A survey of 103 user-centered design practitioners [55] shows HE has the highest impact among informal usability evaluation techniques, though it is the second most frequently used. Jeffries et al. [22] found that HE identifies more serious problems than usability testing, guidelines, and cognitive walkthroughs. Heuristic evaluation is performed by inspecting an interface and identifying usability problems. The evaluation should be conducted according to certain

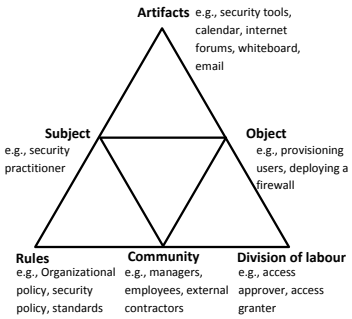


Figure 1: Activity triangle proposed by Engeström. Examples of each component are provided in the context of ITSM.

rules that guide the evaluation process. These rules can be chosen at different levels of granularity. According to Shneiderman [47], guidance to designers can emerge in three forms: "(1) high level theories and models, (2) middle-level principles, and (3) specific and practical guidelines." Te'eni et al. [50] describe principles as "representing the theory with an eye to what we should practice and the guidelines taking the principles one step further toward their application." For heuristic evaluation, the use of principles is recommended rather than theories or guidelines [38]. The most widely accepted heuristics are Nielsen's [38], which are theoretically grounded and extensively tested. They are based on Norman's theory of action [39], and focus on the dialogue between a single user and the physical world. Nielsen's heuristics can be modified or extended to address the characteristics of a specific domain.

2.3 Domain Specific Heuristics

Our review of literature on HE shows that there are two dominant approaches in proposing usability heuristics for a specific software domain. On the one hand, researchers extend or adapt Nielsen's usability heuristics for a specific domain (e.g., ambient displays [29], video games [42], virtual reality [49], medical devices [59], intelligent tutoring systems [30], and intrusion detection systems [60]). On the other hand, researchers develop new heuristics based on a specific theory that takes into account the characteristics of the target domain (e.g., heuristics based on the locales framework [12] for evaluation of groupware [15], heuristics based on the mechanics of collaboration [16] for evaluation of shared visual work surfaces for distance-separated groups [1, 2]). While the heuristics proposed by Baker et. al [1] and Greenberg et. al [15] can be used to evaluate certain collaborative aspects of ITSM tools (e.g., communication channels used during collaboration; and facilities for planning, initiating and, managing collaboration), they do not examine certain characteristics of IT security, such as the need for pattern recognition, and inferential analysis to address previously unknown conditions [6]; the use of historical data and logs in understanding the current context [6, 13]; the involvement of stakeholders with diverse backgrounds and knowledge [57]; the reliance on transactive memory to use other stakeholders' knowledge [5]; and the need for verification while performing actions on complex systems that involve uncertain immediate results [52].

We have found only one instance of applying HE to an

ITSM tool. Zhou et al. [60] developed a list of six heuristics, based on Nielsen's heuristics, for the usability evaluation of intrusion detection systems. They noted that they developed this list based on surveys and interviews with SPs, but they did not provide any details of their evaluation methodology. From their list of six heuristics, four are identical to Nielsen's heuristics, and the other two are extensions.

3. PROPOSED ITSM HEURISTICS

We chose to develop a new set of heuristics for ITSM tool evaluation rather than extending Nielsen's heuristics. Engeström [10] explains that "[action theories] have difficulties in accounting for the socially distributed or collective aspects, as well as the artifact-mediated or cultural aspects of purposeful human behavior." Therefore, while applying Nielsen's heuristics to ITSM tools may improve the usability of tools by helping users form and work toward immediate goals more effectively, it may not improve the usability of the tool by addressing socio-cultural and collaborative issues at the level of activity. On the other hand, heuristics for usability evaluation of groupware [15, 1, 2] do not account for certain characteristics like transactive memory, creativity, and informal social networks of the IT security domain. According to Nardi et al. [32], the bulk of CSCW research considers work in teams that have clearly defined and stable roles; this is not the case in ITSM. In ITSM, SPs need to coordinate ad-hoc teams and involve different stakeholders to perform security activities, such as incident response or policy development [57].

Given these characteristics of ITSM, post-cognitivist theories are good candidates to provide a foundation for developing heuristics for ITSM. These theories extend their focus beyond information processing to include how the use of technology emerges in social, cultural and organizational contexts [24]. Therefore, we developed a set of new heuristics based on activity theory [23], which is considered the most widely used post-cognitivist theory in HCI [31, 23].

The process we used for building heuristics for ITSM began by understanding the characteristics of ITSM tools that help SPs perform activities more efficiently. We collected data from two sources: related work and interviews performed in the HOT-Admin project, which had the goal of understanding the human, organizational, and technical issues in IT security [19]. We first selected a set of primary publications to analyze, which included four HOT-Admin papers, as well as fourteen other papers about ITSM tools. Analyzing the literature as a qualitative data source, we identified 164 explicit guidelines for building ITSM tools, recommendations for improvement, design decisions in a particular tool that have positive impact on usability, and wish lists about tools. We categorized these using grounded theory [8]. First, we performed open coding using codes that emerged from the data, followed by axial coding to combine conceptually similar open codes. Meanwhile, we broadened our survey by reviewing the papers published in well-known conferences related to the topic, performing keyword searches, and mining the references from our original set of 18 papers. The result of this search was a list of 56 papers. We then reviewed the papers and found another 22 papers that could contribute to our guidelines. We also analyzed five semi-structured interviews with SPs to find support for our guidelines and illustrative examples. The interviews were part of the HOT-Admin corpus, but had not been analyzed when the HOT-Admin papers cited in our survey were written. This process resulted in 19 guidelines for ITSM tools (for a full description of the process and the generated guidelines, see [21]).

To develop the ITSM heuristics, we reviewed the guidelines and compared them to the theoretical constructs of activity theory. This helped us combine guidelines supported by the same theoretical construct into higher level principles. The theory allowed us to interpret the rationale behind each guideline, and consequently helped to consolidate and abstract the guidelines into heuristics, which are more general, yet cover a broader range of usability problems than just the guidelines. When principles are crafted as heuristics, they should be concise, easy to understand, and open to interpretation. We next present our heuristics and discuss the rationale behind them:

Heuristic 1 - Visibility of activity status: *"Provide users with awareness of the status of the activity distributed over time and space. The status may include the other users involved in the activity, their actions, and distribution of work between them; rules that govern the activity; tools, information, and materials used in the activity; and progress toward the activity objective. Provide communication channels for transferring the status of the activity. While providing awareness is crucial, limit the awareness to only what the user needs to know to complete his actions."*

Discussion: In IT security, the actions that form an activity are distributed across time and space. These actions are performed in an organizational context with certain norms and rules. Plans are created and modified by different stakeholders, and roles are assigned dynamically to address unknown conditions. Prior ITSM research points to the importance of providing awareness of organizational constraints [58], communication channels [57], methods for sending cues to different stakeholders to inform them about when and how to act [5], awareness of what other stakeholders perform in the system, sharing the system state between different SPs and grounding new participants in ITSM activities [17].

Looking at the problem through the lens of activity theory, tools can provide awareness about the components of activity shown in Figure 1. Carroll et al. [7] described three types of awareness: (1) social awareness, the understanding of current social context in an activity (e.g., rules, artifacts); (2) action awareness, the understanding of actions of collaborators on shared resources; and (3) activity awareness, the understanding of how shared plans are created and modified, how things are evaluated, and how roles are assigned. As IT security tools deal with sensitive information, a balance should be kept between visibility and privacy; in the words of Erickson and Kellogg [11], visibility should be in the form of social translucence rather than social transparency.

While this heuristic is similar to Nielsen's *"Visibility of System Status"* heuristic, there are differences between the two. Nielsen's visibility heuristic focuses on the immediate status of the system, which is needed by the user to decide which action to execute, as well as on the immediate status after execution, which is needed by the user to evaluate the outcome of the action. On the other hand, the ITSM heuristic includes aspects of system status that might not immediately and locally be available to the user. The user must then use different communication channels and cues to understand the status of the system.

Heuristic 2 - History of actions and changes on artifacts: *"Allow capturing the history of actions and changes on tools or other artefacts such as policies, logs, and communications between users. Provide a means for searching and analyzing historical information."*

Discussion: Accountability and reflecting on work are important aspects of ITSM [13, 52]. As ITSM involves creative work to address unknown conditions, providing usage histories supports creativity, learning, and quality improvement [48]. Audits, which aid in reflecting on work, are mandated in IT security contexts as a part of regulatory legislations such as the Sarbanes-Oxley Act [46]. Prior ITSM research [13] showed the need for SPs to archive logs and keep a history of communications for audit and accountability purposes. Furthermore, archives can be used to build an understanding of other stakeholders' actions. For example, in some organizations access control policies are changed by multiple SPs; keeping track of changes will help other security sanctioners maintain a working knowledge of the implemented policy [3]. Finally, historical information can be used for trend analysis, learning about the network, and evaluating the outcome of actions that span time and space [52].

From the theoretical perspective, artifacts in an activity carry a history with them. Awareness of this history impacts the way those artifacts are used. Hollan et al. [20] studied experts working in complex environments and found that usage histories are incorporated in cognitively important processes. Historical information could be in the form of the usage histories of the user himself or of other users of the system. Usage histories can be employed to reflect on work, and to get feedback from peers [48].

Heuristic 3 - Flexible representation of information: *"Allow changing the representation of information to suit the target audience and their current task. Support flexible reports. Allow tools to change the representation of their input/output for flexible combination with other tools."*

Discussion: SPs often use inferential analysis and pattern recognition to develop policies, audit security, or troubleshoot security incidents [6]. For example, they need to look for certain patterns in network traffic to detect an anomaly; or they need to analyze users' access to different resources in order to build an effective set of role-based access control (RBAC) roles. To perform these activities, SPs often use their tools in creative ways that were not anticipated by tool developers; or alternatively, they combine their tools. Botta et al. [6] identified SP's practice of bricolage (i.e., combining different tools in new ways) to address complex problems and argue that ITSM tools should survive in the arena of bricolage. Haber et al. [17] and Beal [4] also point to the need for better integration between IT security tools.

Tools should also be flexible in representing information to allow stakeholders to use them in different ways based on the task at hand or on their background and knowledge about the tool. In prior ITSM research, the need for flexible interaction methods (e.g., Command Line Interface and Graphical User Interface) [6, 51], flexible reporting [57, 6, 53], visualization techniques [9], and multiple views [17] are highlighted.

From the theoretical perspective, ITSM activities involve a distributed cognition process [5, 28]. ITSM tools mediate artifacts between stakeholders, stakeholders and artifacts, and other artifacts. According to Norman [40], artifacts have two types of representation: the internal representation that is not accessible by the outside world, and the surface representation that is their interface to the world. In IT security, tools participate in actions that cannot be anticipated by tool developers [5]. Therefore, tools should be able to flexibly provide different surface representations in order to act as mediating artifacts in various scenarios. This signifies the need to provide a wide range of representations, as well as a way to build customized representations. Prior activity theory research [23, 43] also shows that users combine and adapt different artifacts to build instruments to perform their actions in unexpected and unknown conditions, and it argues in favor of highly customizable and open tools.

Heuristic 4 - Rules and constraints: *"Promote rules and constraints on ITSM activities, but provide freedom to choose different paths that respect the constraints. Constraints can be enforced in multiple layers. For example, a tool could constrain the possible actions based on the task, the chosen strategy for performing the task (e.g., the order of performing actions), the social and organizational structure (e.g., number of stakeholders involved in the task, policies, standards), and the competency of the user."*

Discussion: ITSM tools are used in an organizational environment with certain rules, norms, and constraints. Violating these constraints might result in sub-optimal situations; therefore, tools can help enforce such constraints. Botta et al. [5] show that enforcing norms by ITSM tools in the form of procedures for notification and support for particular templates and standards can prevent communication and collaboration breakdowns. Werlinger et al. [56] argue that ITSM tools can promote security culture in organizations and address the lack of training in stakeholders by enforcing policies.

From the activity theory perspective, there are rules and norms that govern every activity. Promoting rules and norms by tools can lead to awareness and internalization of those norms by stakeholders [23]. Vicente [54] points out the importance of enforcing rules and constraints by tools, while allowing users to flexibly explore the possible action space. This helps users be aware of constraints, and gives them flexibility to adapt to unexpected situations. Vicente argues that constraints can be expressed at five different levels: work domain, control tasks, strategies, social-organizational, and worker competencies. Rules in ITSM can include security and privacy policies or standards, organizational constraints, and organizational culture.

Heuristic 5 - Planning and dividing work between users: *"Facilitate dividing work between the users involved in an activity. For routine and pre-determined tasks, allow incorporation of a workflow. For unknown conditions, allow generation of new work plans and incorporation of new users."*

Discussion: SPs work in an environment that requires fast responses to unknown conditions. Furthermore, SPs and managers work with very tight schedules in which security has a low priority [6]. Therefore, it is important to help the planning and division of work between different stakeholders [56]. SPs often need to coordinate activities with multiple stakeholders involving other SPs, IT admins, managers, end-users, and external stakeholders. For example, to address a security incident, SPs often need to collect data from end-users or other IT specialists; analyze the in-

cident; coordinate and collaborate with IT specialists, who own the impacted sub-systems (e.g., database admins, web-server admins); communicate with managers to warn them about the risks associated with the incident and possible disruptions in service; and even collaborate with external SPs to solve the problem. In all of these cases, proper planning tools should be available to quickly involve stakeholders and divide work between them.

Activity theory points to the division of labour as an important aspect of activity. Furthermore, division of labour should take into account constraints at the social organizational level, as well as possible methods for generating plans and collaborating together considering those constraints.

Heuristic 6 - Capturing, sharing, and discovery of knowledge: *"Allow users to capture and store their knowledge. This could be explicit by means of generating documents, web-pages, scripts, and notes, or implicit by providing access to a history of their previous actions. Tools should then facilitate sharing such knowledge with other users. Furthermore, tools should facilitate discovery of the required knowledge source. The knowledge source can be an artifact (e.g., document, web-page, script) or a person who possesses the knowledge. Provide means of communicating with the person who possess the knowledge."*

Discussion: SPs rely heavily on tacit knowledge in order to perform their tasks [5]. For example, in order to implement security access controls, a security practitioner needs to know about different activities that a stakeholder needs to perform, and the resources in the system to which access should be granted in order to allow the stakeholder to perform those activities. To address problems in the complex and evolving scene of ITSM, SPs need to use the knowledge and experience of other stakeholders involved in the activity. This can be in the form of other stakeholders' tacit knowledge in the organization or the knowledge distributed in the community. Prior research in ITSM points out the importance of managing tacit knowledge [5] and suggests policy specification as a method to transfer such knowledge [56]. Kesh et al. [25] demonstrate the importance of knowledge management in IT security. Rogers [44] discusses the need for transmitting knowledge at the "window of opportunity" during troubleshooting in a network environment that involves multiple stakeholders and describes it as a challenging task.

From the theoretical perspective, the relationship between different actors in the activity is mediated by artifacts. As a result, in order to transfer knowledge, users should be able to externalize their knowledge as artifacts [10]. Facilities for identification and access to the required knowledge sources must then be provided. If externalization of knowledge is not feasible, a method for finding and starting collaboration with the person who possesses the knowledge should be provided. In this case, the communication channel is considered the mediating artifact.

Heuristic 7 - Verification of knowledge: *"For critical ITSM activities, tools should help SPs validate their knowledge about the actions required for performing the activity. Allow users to perform actions on a test environment and validate the results of these actions before applying them to the real system. Allow users to document the required actions in the form of a note or a script. This helps the users or their colleagues to review the required actions before applying them to the system."*

Discussion: Many actions in ITSM are responses to new, unseen, and complex situations [6, 5]. These actions are performed on artifacts that are critical to the organization. Moreover, the actions are distributed in time and space and the result of an action cannot be evaluated in real time. Therefore, errors in ITSM activities could lead to a security breach or disrupt services to the organization's employees, which might impose high costs on the organization. For example, an error during deployment of a patch to address a serious security vulnerability might disrupt service and conflict with an organization's business activities [5]. On the other hand, it is hard to predict, or instantly determine, the outcome of the patching process, as other stakeholders need to confirm that the service is not impacted by the patching process. To mitigate this, SPs employ "rehearsal and planning" [17], by rehearsing the actions on a test system before performing it on a production system.

This practice can be clarified from a theoretical perspective. To find a solution to a new or complex problem, an SP usually consults several information sources and combines them into a single artefact (e.g., a plan, a guide document, a check list). This artefact acts as an external memory to the subject. Moreover, in this process, the SP internalizes knowledge from different sources. The internalized knowledge might not be completely correct or applicable to the situation at hand. Therefore, it should be verified before applying it to the system. Activity theory asserts that the process of revising knowledge involves externalization of knowledge, performing revision, and internalizing the revised knowledge again [10]. In the context of ITSM, SPs perform externalization when they employ rehearsal. If something goes wrong in the rehearsal, SPs re-examine their interpretation of the external knowledge sources and go through the rehearsal and revision cycle again. After successful rehearsal, SPs can perform the rehearsed actions on the critical artefact.

In this section, we described seven heuristics for the evaluation of IT security tools. We described each ITSM heuristic, provided evidence about the usefulness of the heuristic in the IT security domain, and the theoretical support for the heuristic. In the next section, we describe the methodology we employed when we examined the use of heuristics for the evaluation of an IT security tool.

4. EVALUATION METHODOLOGY

While the ITSM heuristics are grounded in empirical data and supported by theory, the effectiveness of the heuristics must be validated by using them in a standard heuristic evaluation process. The ultimate criteria for the effectiveness of a set of heuristics (or a usability evaluation method in general) is finding real problems that users will encounter in real work contexts, which will have an impact on usability (e.g., user performance, productivity, and/or satisfaction) [18]. However, it is not possible to determine if each usability problem is real or not [41]. The best we can do is to estimate the impact of the potential problem on the users who will use the system. We evaluated our approach based on the the following criteria for comparison: (1) thoroughness, the ability of the method to find most of the known problems; (2) reliability, the ability of the method to find severe problems; and (3) validity, the ability of the method to find valid problems.

Besides investigating the effectiveness of the ITSM heuristics, we wanted to investigate the characteristics of an eval-

uation which uses them; and we wanted to compare them to the characteristics of evaluation using Nielsen's heuristics. The evaluation characteristics we considered include: (1) the number of evaluators required; (2) background knowledge required; and (3) the usefulness, ease of learning, and ease of applying heuristics.

To achieve the aforementioned goals, we performed a comparative study of the ITSM heuristics with Nielsen's heuristics. The design of the study was between-subjects, and participants were divided into two groups: those that used Nielsen's heuristics (Nielsen condition, 14 participants) and those that used the ITSM heuristics (ITSM condition, 14 participants). For the ITSM condition, we performed 3 in person evaluation sessions (3, 3, and 1 participants per session), and 7 remote evaluation sessions (1 participant per session). For the Nielsen condition, we performed 4 in person evaluation sessions (3, 2, 2, and 1 participants per session) and 6 remote evaluation sessions (1 participant per session).

Ethical Considerations: According to the University of British Columbia's (UBC) policy, any research project that involves human subjects must be reviewed and approved by the Behavioral Research Ethics Board (BREB). Our study involved HCI professionals and experts who are categorized as a "very low" vulnerability group. Furthermore, our research procedures did not involve any participant risk (e.g., physical, emotional, psychological, financial, legal, privacy, reputation, group or social status). Therefore, our study was approved as a minimal risk study. We were required to submit our study materials, consent forms, and study procedures to the BREB. Furthermore, all members of the research team completed a mandatory online tutorial in research ethics before the submission of the ethics application.

Recruitment: The main inclusion criteria for our study was a human computer interaction background, and familiarity with heuristic evaluation. To recruit participants, we sent emails to all graduate students in the Computer Science and Electrical and Computer Engineering departments of UBC. We also sent emails to the user experience mailing list in Vancouver, to online HCI communities, and the CHI-Announce mailing list, in order to reach participants with professional and academic HCI experience; and to Usable Security mailing lists, in order to reach participants with a background in both security and usability. All participants were given a $50 (Cdn) honorarium for their participation.

Participants: In an attempt to balance the expertise of participants in each group, we screened them to assess their HCI and computer security background. In Table 1 we present our participants' demographics in terms of age, gender, level of education, and educational background. We also indicate the years of professional and research experience our participants had in HCI and computer security. All but one participant had received formal HCI training, with the majority (17) receiving formal training specifically on heuristic evaluation. Also, the majority of participants (19) had performed at least one heuristic evaluation in the past. In order to validate whether the expertise of the two groups was balanced, we calculated scores regarding participants' experience in HCI and computer security, using the weighted average of the different experience indicators. To quantify HCI experience, we used years of experience, number of courses, prior HE training, and number of prior heuristic evaluations. To quantify computer security experience, we used years of experience, number of courses, prior experience with IdM systems, and prior experience in working in a organization that uses role-based access control. Independent sample t-tests revealed no statistically significant difference in HCI and computer security scores between the two groups, as measured by our scoring system (HCI: Nielsen's (M = 5.93, SD = 4.01), ITSM (M = 6.48, SD = 4.81), t(25.16)=-0.33, p=0.75; Computer security: Nielsen's (M = 2.42, SD = 1.64), ITSM (M = 3.39, SD = 3.55), t(18.31)=-0.92, p=0.37).

Table 1: Participants' demographics for each condition.

Condition		ITSM	Nielsen	Total
Group Size (N)		14	14	28
Age	19-24	2	2	4
	25-30	6	7	13
	31-35	4	1	5
	36-45	2	4	6
Gender	Female	6	6	12
	Male	8	8	16
Educational Level	Diploma	1	0	1
	Undergraduate	6	8	14
	Graduate	7	6	13
Years of experience (avg.)	HCI research and professional	3.57	3.29	3.43
	Computer security research	0.64	0.50	0.57
	Computer security professional	1.0	0.32	0.66

Target System: We chose an Identity Management (IdM) system as the target system for performing the heuristic evaluation. An IdM system is used to manage the digital identities of users in an enterprise, and control the accesses of those identities to resources. Furthermore, the system allows the request and approval of access to resources, auditing, reporting, and compliance. We installed CA Identity Manager 12.0 CR3 in a laboratory environment on a virtual machine using VMWare Server. Access to the system was through a web interface.

Study protocol: An overview of the study protocol is provided in Figure 2; we now describe the details of each step.

Figure 2: Study protocol overview

We began by obtaining the participants' consent, and then asked them to complete a background questionnaire (Appendix 1). In the questionnaire, we obtained demographic information and collected data to assess the background of the participants on HCI and computer security. This questionnaire was similar to the screening questionnaire, but we also collected qualitative data to clarify quantitative data (e.g., besides the number of HCI courses taken, we asked participants to provide the list of such courses).

We then provided training on heuristic evaluation for the participants, and described the specific heuristic set to be used during the heuristic evaluation. We demonstrated the application of the heuristics in a running example of evaluating a network firewall system. We concluded the training

session with an introduction on the IdM system to be evaluated. In all cases, training material was presented to the participants through online slides with vocal narratives. Using slides with recorded narratives allowed us to provide exactly the same training to all participants in each condition, whether they participated in person or remotely.

During the heuristic evaluation, participants inspected the interface individually. Each had access to an instance of the IdM system, and all the instances were identical. We limited the scope of the evaluation to a few typical usage scenarios [45]. Using scenarios has two main benefits. First, the IdM system is complex, and it is not possible to cover the whole software in one evaluation session. Second, scenarios can guide evaluators who are not domain experts in performing specific tasks on the IdM system. The scenarios were designed for an example organization, and each IdM instance was configured with the users and structure of the example organization. We also described the characteristics (e.g., background, knowledge, work schedule) of the stakeholders referred to in the scenarios, and we provided their usernames and passwords in the IdM system. The evaluators could then log-in to the system as the various stakeholders while they performed the steps of the scenarios. An overview of the four scenarios used in the study is presented in Table 2 (Appendix 2 includes the complete evaluation guide). We provided participants with the list of scenarios and asked them (1) to identify usability problems with the provided set of heuristics; and (2) for each problem, to specify the scenario and the heuristic. The participants entered the identified problems in an online form (Appendix 3). Participants had two hours to perform the evaluation.

After the evaluation session, participants were provided with a post-evaluation questionnaire (Appendix 4) to rate their experience in using heuristics. We then conducted either a focus group session (for sessions with multiple in-person participants) or an interview (for remote or single in-person participants) to discuss participants' experience in using the heuristics, capture their suggestions for improving heuristics, and discuss usability problems that cannot be associated with any of the heuristics. We added this step so that we could gather qualitative data, which can better reveal the reasons behind whether or not the heuristics are useful, easy to use, and easy to apply. Furthermore, we were able to probe the usability of the interface in general and to discuss usability issues that might not be related to either set of heuristics.

We piloted and refined our study protocol and materials through several iterations. We performed two complete pilot study sessions (6 and 2 participants); and we held several pilot tests as we iterated upon the individual study components, including the background questionnaire (total of 6 participants), the description of the heuristics (6), the training materials (2) and the evaluation guide (7).

Table 2: Scenarios Details

Scenario	Description
Self-serve user creation	A *contractor* just arrived at a company and wants to create a user account. He should use the self-service feature in the IdM system to create an account. Then *a member of IT security team* should review and approve his request.
Bulk user creation	When an employee is hired by the company, or information about an employee changes, or an employee leaves the company, the changes are reflected in the *HR (Human Resources)* system. A member of the *security team* receives a file containing all the changes from the HR system, uploads the file to the system and troubleshoots errors.
Request privileges	When an employee needs access to certain resources he initiates a request. The request should first be approved by his manager. After the manager's approval, the request should be reviewed, and implemented by a security admin.
Certification process	Security team frequently initiates employee certification. In this process, the manager of each employee receives a request that he should review and certify the privileges of his employee. The security team then reviews the result of certification and closes the certification process by revoking all of the non-certified privileges.

4.1 Data Analysis

A crucial step in heuristic evaluation is aggregating the problems found by different evaluators and determining the severity of the problems. Our pilot study revealed that evaluators find problems at different levels of granularity, find duplicate problems, and state problems using different terminology. As a result, before aggregating all of the problems, we needed to make the granularity of problems consistent across all of the evaluators. In order to have a consistent and repeatable methodology of aggregating the problems and rating their severity, we used the following steps. These steps were performed by two researchers, and any inconsistencies were resolved by consensus.

Aggregating problems: we performed the following steps to aggregate problems found in each condition and generate a list of known problems in both the ITSM and Nielsen conditions:

1. Problem Synthesis: We first decomposed problems into their finest level of granularity. Since each part of a compound problem might have a certain severity, and therefore a priority for fixing, they need to be decomposed into multiple problems. Compound problems include those that refer to different actions, different artifacts, or different mechanisms in the interface. In addition, we eliminated unknown problems and false positives. If a problem could not be reproduced by the researchers (e.g., it happened due to a sudden breakdown or crash in the system during the study, or the description of the problem was not understandable), we removed the problem from the list and marked it as unknown. We marked as false positives any problems that had any of the following characteristics: (1) the problem was caused by the constraints or requirements of the underlying operating system or hardware/software infrastructure, (2) the problem was caused by the business constraints or requirements of the program, or (3) the reasoning of the evaluator in describing the problem was fallacious.

2. Aggregating problems: After problem synthesis, each researcher began with an empty list of aggregated problems. Each identified problem was compared with the problems in the aggregated list. If the problem was not present in the list, it was added to the list. Otherwise, the description of the problem in the aggregated list was refined. Furthermore, if the set of heuristics associated with the problem in the list was different from the set associated with the problem to be aggregated with the list, the association was updated as the union of associations.

3. Tagging the problems with heuristics: Researchers re-

viewed each problem in the list, and tagged the problem with one or more of the heuristics that were the most relevant to the problem. This process was performed without looking at the original heuristic(s) which were provided by evaluators when each problem was found.

Assigning severity ratings: We used five levels of severity: 0-not a usability problem, 1-cosmetic, 2-minor, 3-major, and 4-catastrophe [34]. We asked four usable security researchers, who had training in heuristic evaluation, to independently determine the severity of each problem. We asked them to judge each problem based on its frequency, impact, and persistence. We then used the mean of their severity ratings as the severity for the problem. Based on the mean severity rating, we placed the problems into two categories: major (mean severity > 2) and minor (mean severity ≤ 2).

5. EVALUATION RESULTS

Overview: Table 3 shows the classification of the problems in each condition. The "Problem Reports" column shows the initial number of problems reported by the evaluators. The "Tokens" column shows the number of valid reported problem tokens for each condition after removing unknown problems, false positives, and problems that each evaluator reported multiple times. If a problem is reported by multiple evaluators, we counted each of the problem reports as a token. The "Known" column shows the number of problems after combining problems that are reported by multiple evaluators. For example, if a problem is found by multiple evaluators, we counted it as a single known problem. The table shows that, although the synthesis involved decomposition, it resulted in fewer problems in each condition. This was due to combining problem tokens that are found by multiple evaluators, and eliminating false positives and unknown problems. Table 3 also shows the classification of known problems as either major or minor severity. Table 4 shows the number of problems that are unique to each condition, classified by their type. Our analysis revealed that of the 131 total known problems, only 48 were identified in both conditions.

Effectiveness of heuristics: Based on the number and severity of the problems found in each condition (Table 3), and Table 4), we compared the effectiveness of the heuristics used in each condition. Since the final output of a heuristic evaluation is a combined list of problems (as opposed to individual lists by different evaluators), we calculate effectiveness metrics based on the aggregate of problems from different evaluators (known problems instead of individual problem tokens):

Thoroughness: We calculate *thoroughness* as the proportion of the problems identified in each condition. Our results show that the evaluation with the ITSM heuristics resulted in finding 71% of total known problems while the evaluation with Nielsen's heuristics resulted in finding 66% of them. In some cases, finding fewer, but more severe, problems might be more important than finding many minor problems. To examine this, we used the notion of Weighted Thoroughness (WT) by increasing the weight of the problems based on their severity [18]. Using Equation 1 (we used an equivalent equation for Nielsen condition), the weighted thoroughness of ITSM and Nielsen's heuristics are 77% and 60% respectively.

Table 4: Overview of problems unique to the conditions

Condition	Known	Major	Minor	FP	Unknown
ITSM	45	23	22	17	16
Nielsen	38	5	33	44	17

$$WT = \frac{\sum_{p \in KnownITSM} Severity(p)}{\sum_{p \in Known} Severity(p)} \times 100 \quad (1)$$

Reliability: It is important for a set of heuristics to be able to identify major usability issues as they may seriously hinder the ability of the user to operate the system effectively and efficiently. Table 3 shows the total number of problems identified in our study for each condition, as well as the number of major and minor problems. We conducted a chi-square test for independence in order to determine whether participants using the set of ITSM heuristics found more severe usability problems for the examined system than the ones using Nielsen's set. The result was statistically significant ($\chi^2(1, 179) = 5.54$, p = .019, phi = -.18).

Because we had a subset of problems that were common between the two conditions, before drawing conclusions about how much better participants performed in identifying major usability issues using our ITSM heuristics than those using Nielsen's set, we needed to determine the extent to which the major issues were identified by both sets. We compared the severity of unique problems (Table 4) in each condition. A chi-square test revealed that our ITSM heuristics were able to identify significantly more major usability issues for the examined IdM system than the Nielsen's heuristics ($\chi^2(1, n = 83) = 11.63$, p = .001, phi = -.4).

Validity: Another aspect of our ITSM heuristics that we examined was whether they yielded fewer false positives than in the evaluation using Nielsen's set of heuristics. Participants using the ITSM heuristics reported 201 known problem tokens and 18 false positives, whereas participants using Nielsen's heuristics reported 187 known problem tokens and 45 false positives. The ITSM heuristics yielded significantly fewer false positives (Table 3) than Nielsen's heuristics ($\chi^2(1, 451) = 10.8$, p = .001, phi = -.161). Comparing the number of unknown problems identified in each condition revealed no significant difference between conditions.

Next, we extended our analysis to the characteristics of the heuristic evaluation process using ITSM heuristics and compared them to Nielsen's heuristics. These characteristics included individual differences in the evaluators' ability to find usability problems, the number of evaluators required for performing evaluation, a comparison of individual problem finding ability between the two conditions, a comparison of the number of problems that cannot be associated to a heuristic, and the evaluator's opinions about the heuristics.

Individual differences in evaluators' ability to find usability problems: While analysis of the aggregated list of problems shows the overall performance of heuristics, it masks certain characteristics. Looking at the individual performance of evaluators in terms of the proportion of the total known problems that they found can reveal how the heuristics can be used in the real world. For example, if heuristics help

Table 3: Overview of identified problems

Condition	Problem Reports	Problem Tokens	Known Problems	Major	Minor	False Positive (FP)	Unknown
ITSM	239	201	93	38	55	18	16
Nielsen	233	187	86	20	66	45	17
All	472	388	131	43	88	62	33

most of the evaluators find most of the known problems, one might prefer to use only one evaluator for performing the evaluation (This is the case with usability guidelines that are not open to interpretation.) On the other hand, if heuristics allow each evaluator to find a subset of problems, one might use multiple evaluators to find many problems in the interface.

To investigate how evaluators find problems using ITSM heuristics, and to then compare ITSM heuristics to Nielsen's heuristics, we replicated the analysis performed by Nielsen in his original heuristic evaluation experiment [38], and compared our results with it. Since the Nielsen's experiments were performed on small scale, single user systems, we also compared our results to the results of heuristic evaluation of a groupware application by Baker et. al [2]. In this section, we consider evaluator B stronger than evaluator A if she finds more usability problems than A.

We show the summary of individual differences in the evaluators' ability to find problems for each condition in Table 5. In addition to the performance of the strongest and weakest evaluators, we calculated the proportion of problems found by the first and third quartile to eliminate the impact of outliers. We also listed the ratio between the values as an indication of the difference between individual performances. These proportions are calculated based on the total problems (131) found. Nielsen [38], in his four heuristic evaluation experiments, observed that the individual differences between evaluators is higher in systems that are more difficult to evaluate. Our results confirm Nielsen's observation as we observed larger individual differences than those reported in Nielsen's experiments (2.0 and 1.9 compared to 1.4, 1.6, 1.7, and 2.2). The evaluated IdM system was a domain specific system, and our participants didn't have any prior working experience with it or similar systems. Therefore, it can be considered harder to evaluate compared to the systems evaluated by Nielsen [38].

In Figure 3, we show the distribution of the proportion of identified problems by the proportion of the evaluators in each condition. Our results in the Nielsen condition show a similar pattern (bell-shaped) to Nielsen's original experiment [38], while the ITSM condition is different (skewed to left). Our result in the ITSM condition is similar to that seen by Baker et. al [2] in their heuristic evaluation of the Groupdraw interface, which was completed by regular usability specialists.

The number of evaluators required to perform the evaluation: To replicate Nielsen's original analysis [38], we formed aggregates of evaluators and found the proportion of usability problems identified by each size of aggregate. Following Nielsen's methodology, we calculated the proportion of found problems based on the total number of problems found in each condition. The result is depicted in Figure 4. The ITSM and Nielsen's graphs show that increasing the number of evaluators will increase the proportion of the identified problems, but the rate of the increase diminishes as we

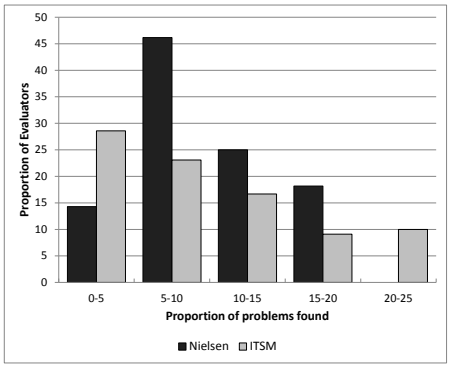


Figure 3: Distribution of the proportion of the identified problems in both conditions

increase the number of evaluators. We also overlayed the results from Nielsen's Mantel experiment [38], and Baker's Groove and GroupDraw [2] experiments to allow comparisons. Two graphs from our experiment are very similar and they show a similar trend compared to Mantel, Groove, and GroupDraw experiments.[1] Yet, Nielsen's experiment shows faster diminishment compared to our results. In our experiment, we observed a slower decrease in rate of finding problems. This can be attributed to the complexity of the system, and shows that we will need more evaluators to find most of the problems in a complex ITSM system, as compared to small-scale interfaces, such as Mantel studied by Nielsen [38]. The graphs for Groove and GroupDraw, which are more complex than Mantel, support this finding.

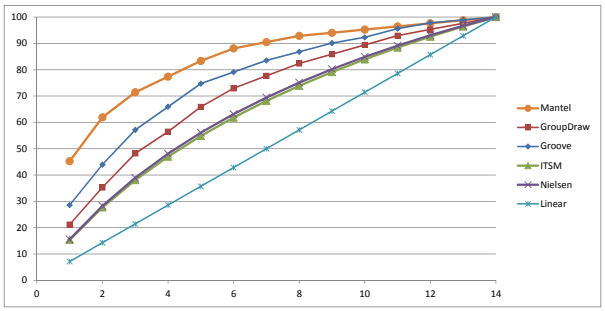


Figure 4: Average proportion of problems found by aggregate of evaluators

We illustrate the distribution of the known problems that are found by evaluators in the ITSM or Nielsen condition in Figure 5. Each row corresponds to an evaluator and each

[1]To allow comparison, and since the mentioned experiments employed more evaluators, we assumed that the total number of problems in each experiment was equal to the problems found by aggregate size of 14.

Table 5: Individual differences in evaluators' ability to find problems. The numbers show the proportion of known problems.

Condition	Max(%)	Min(%)	$Q_1(\%)$	$Q_3(\%)$	Max/Min	Q_3/Q_1
ITSM	23.7	3.82	7.1	13.9	6.2	2.0
Nielsen	18.3	3.1	5.9	11.5	6.0	1.9

column corresponds to a problem. To generate this diagram, we grouped evaluators based on their condition and then ranked them as weak to strong based on the number of identified problems. We also ranked the problems from easy to hard based on the number of evaluators who found the problem. In this figure, evaluators are sorted from bottom (weak) to top (strong) and problems are sorted from right (easy) to left (hard).

The diagram shows that, similar to Nielsen's original experiment [38], there are easy problems that are overlooked by strong evaluators, while there are hard problems that are only found by weak evaluators. This confirms Nielsen's argument that heuristic evaluation is a method that should be done collectively (i.e. there is no one very strong evaluator that can uncover all the problems). It also shows that there was relatively little duplication between participants in each condition. We further discuss the lack of duplication in Section 6.

Comparing individual performances between conditions: We calculated the average number of problem tokens in each condition and their average severity. When we use the number of problem tokens instead of known problems in the analysis, we analyze the individual problem finding behavior. Our analysis shows that there is no significant difference between the number of reported problem tokens by evaluators in ITSM condition compared to that of Nielsen condition. On the other hand, the severity of the problems reported by evaluators in ITSM condition is significantly higher compared to that of Nielsen (ITSM (M=2.07, SD=0.59) and Nielsen (M=1.69, SD=0.56); $t(386)=6.49$, $p<0.001$).

Problems that cannot be assigned to a heuristic: We asked the evaluators to associate each problem with one or more heuristics which they used to find the problem. We also gave them the option to specify a problem and mentioned that they cannot associate it with any of the heuristics. A high number of problems that cannot be associated with a heuristic can be an indication of: (1) The complexity of heuristics (2) Problems that are not related to the heuristics, but can be found using evaluators' expertise. The average number of problems that were not associated to a heuristic by an evaluator in ITSM condition was 3.29 (SD=5.25), and in Nielsen condition it was 0.71 (SD=1.20). We compared the two averages using an independent sample t-test. The result showed no statistically significant difference between the two conditions.

Participants Experience: Finally we asked our participants in each condition to evaluate with a 5-point Likert scale (1=agree strongly, 5-disagree strongly) how effective each set of heuristics is in identifying usability problems, how easy it is to understand and learn the heuristics, and how easy it is to apply the heuristics to the IdM system. We present the means of the Likert scale scores for each condition in Table 6. We conducted a Mann-Whitney U test to evaluate whether the set of heuristics used would have any impact on the efficiency, learnability, and ease of application as reported by our participants. Although, our set of heuristics was new to our participants there was no significant difference between the ratings for the two sets of heuristics in terms of learnability, effectiveness in identifying problems, and ease of applying them to the IdM system.

Table 6: Mean scores of participants' reported effectiveness, learnability, and ease of application for the heuristics (1=agree strongly, 5=disagree strongly).

Condition	Effectiveness	Easy to understand	Easy to apply
ITSM	2.86	2.64	3.14
Nielsen	2.64	2.43	2.50

6. DISCUSSION

Our results show that ITSM heuristics performed well in finding usability problems in the ITSM tools. However, there are aspects of our results that require discussion. First, compared to Nielsen's original heuristic evaluation experiment, there were fewer overlaps between identified problems by evaluators in each condition. For example, there were only three problems that were identified by the majority of evaluators in ITSM condition; this was the same situation in the Nielsen condition. Furthermore, in both conditions, about half of the problems were identified by more than one evaluator (46 problems in each condition). In Nielsen's evaluation on the Mantel and Savings systems [38] there were only one and two problems respectively that were identified by only one evaluator. In Baker et. al's [2] evaluation of two collaborative shared-workspace software, GroupDraw and Groove, 14 out of 64 and 5 out of 43 problems were found only by one evaluator. Our results show fewer overlaps between problems identified by different evaluators, compared to Nielsen's and Baker's results. Two factors can contribute to this observation. First, the evaluated IdM system is not a small-scale system; the evaluators had to visit 20 different pages in order to successfully complete all of the scenarios and this provided an opportunity for finding more diverse problems than the systems evaluated by Nielsen or Baker (e.g., Mantel only had a single screen and a few system messages, GroupDraw had two screens). Second, we used fewer evaluators (14) compared to 77, 34, 25, 27 evaluators in Mantel, Savings, GroupDraw, and Groove systems. This can result in fewer overlaps between problems.

It is important to note that for both sets of heuristics 14 evaluators were not enough to achieve saturation in the identified problems. We had expected to follow Nielsen's recommendation of 3-5 evaluators for each condition [38], and had thought that 14 per condition would be more than

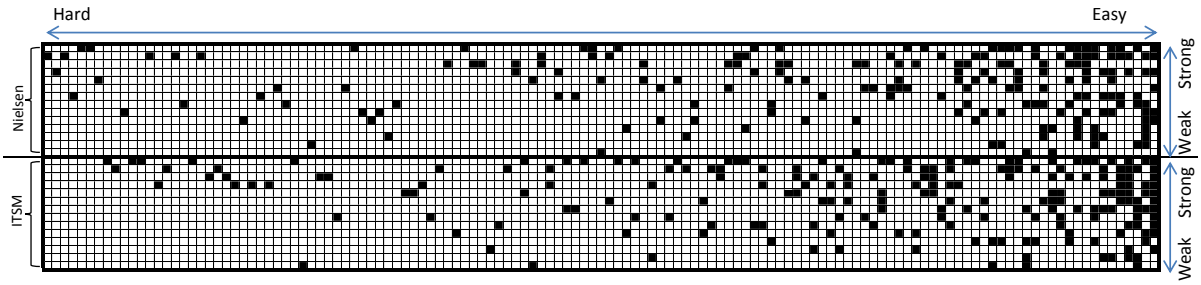


Figure 5: Problems identified by each evaluator in each condition.

enough. We believe that the complexity and scale of the system led to participants finding diverse problems. In addition, the evaluated system was a commercial product rather than a prototype. As a result, the target system did not contain many obvious usability problems that would be found by many evaluators. The mathematical model proposed by Nielsen [37] can be applied to our results to predict the number of evaluators at which we reach the point of diminishing returns.

Before the study, we expected to have very few overlapping problems (<5%) between the two conditions. Our results show that 48 problems (37%) were found in both conditions. Based on the feedback from the focus groups and interviews with participants, we found several reasons for the overlap. Many of our evaluators in the ITSM condition could remember Nielsen's heuristics from their prior heuristic evaluation experiences. That impacted the results of their evaluation by helping them see problems related to the Nielsen's heuristics. Furthermore, some of our participants mentioned that during the heuristic evaluation, they first found a problem based on their experience and then tried to fit it into one of the heuristics. This again resulted in finding problems (mostly in ITSM condition) that overlap with those that had been found in Nielsen condition. Finally, some of the identified problems could be found using both sets of heuristics. For example, the ITSM "Visibility of Activity Status" and Nielsen's "Visibility of System Status" heuristics can both find a subset of visibility problems.

The ITSM heuristics are designed to find problems that are more specific to the ITSM domain. Therefore, we expected the participants in the ITSM condition to find fewer problems as compared to the Nielsen condition. Surprisingly, our results showed no difference in the number of identified problems between evaluators in two conditions. By reviewing the identified problems and analyzing the feedback of the participants, we realized that the participants in the ITSM condition found domain specific problems using ITSM heuristics, as well as finding problems at the level of interaction with users via their background in HCI and Nielsen's heuristics. On the other hand, participants in the Nielsen condition, found certain problems at the level of interaction using their HCI background. Taking into account the classification of problems by researchers, there were 35 problems that were identified in the ITSM condition that we believe should be classified as Nielsen's, and 21 problems were found in the Nielsen condition that we believe should be classified as ITSM. Of those, 7 and 12 problems were only identified in the Nielsen's and ITSM conditions respectively; but our research team classified them as ITSM and Nielsen's. This led to the relatively large overlap between two conditions. Looking at the non-overlapping problems, our results show that evaluators in the ITSM condition mostly focused on problems that are ranked as major, while evaluators in the Nielsen condition mostly focused on cosmetic problems that might not be as important to those who work with the IdM system.

These results suggest that using the ITSM heuristics can result in finding more severe problems by individual evaluators in ITSM tools, while using Nielsen's heuristics might result in ignoring certain domain-specific major problems. Consequently, we suggest using both the ITSM and Nielsen's heuristics together. In our study the inclusion criteria were (1) an HCI background and (2) familiarity with heuristic evaluation. This led some participants in the ITSM condition to find problems based on their prior HCI and heuristic evaluation experience. On the other hand, people who perform heuristic evaluation might not always have an HCI background (e.g., a programmer or a software architect). Providing Nielsen's heuristics will help them find a wider range of problems.

Each individual ITSM heuristic might be applicable to other work domains. The "Visibility of activity status" and "Planning and Dividing Work" heuristics can be important in the evaluation of any collaborative software. "History of actions on artifacts", and "Flexible Mediation" are applicable to domains that require intensive inferential analysis, pattern recognition, and addressing previously unknown conditions. "Knowledge sharing" and "Rules and constraints" are particularly important in evaluating software that is deployed in the organizations. "Verification of knowledge" is important to software that operates on critical information. One particular domain that our proposed heuristics can be used in is IT management. Prior research [6] shows that many SPs perform general IT activities as well as IT security. Furthermore, like IT security, IT involves complexity, collaboration, and dealing with different stakeholders [17]. On the other hand, while these two activities are similar, there are number of factors that make IT security more challenging [13]; and as a result, IT security requires better support from tools. First, IT security involves a higher degree of complexity due to uncertainty, reliance on tacit knowledge, the sensitive nature of the process, and so tools should support SPs in addressing the complexity. Second, IT is perceived more positively than IT security in organizations. IT security tools can play a role in promoting security norms and culture. Third, IT security requires a fast response, and up-to-date knowledge about security issues. This makes knowledge sharing important in IT security. Finally, IT se-

curity requires maintaining a wide and deep overview of organization. Therefore, providing the awareness about other stakeholders' activities should be one of the aspects of IT security tools.

Finally, our ITSM heuristics were novel to the participants. It is encouraging that despite this novelty, participants found the heuristics to be no less effective, easy to use, or easy to learn than Nielsen's heuristics. However, it should be noted that overall, participants were neutral in their ratings for both sets of heuristics. We will continue to refine the heuristics based on the feedback that participants gave during the post-session focus group and interviews.

7. FUTURE WORK

There are several opportunities for future work. First, during the problem synthesis stage, the severity of problems was determined by four severity raters with a background in usable security. While this is a standard approach for determining the severity of problems in heuristic evaluation, it is only an approximation of severity. Asking real users of the tool (who are more familiar with the context in which the tool is used) to determine the severity of the problems is another method of approximating the severity of the problems. While neither of these approximations might be precise, combining the ratings would increase the confidence in determining the severity of the problems. In our future work, we plan to ask real users of the tool to go through the problems and rank their severity. We will then triangulate the results with the rankings from usable security experts to better determine the severity of problems, and consequently compare ITSM and Nielsen's heuristics.

Another future direction of our research is to perform either a controlled laboratory study or naturalistic observation of tool use to find the actual usability problems with the IdM system. Comparing the result of the lab study or observation with the result of heuristic evaluation would show which of the problems identified using heuristic evaluation could impact the performance of the users using the IdM system. However, prior research shows that heuristic evaluation tends to identify different problems than those identified in the lab study or observation. Therefore, we expect few overlaps.

In this study, we chose to compare our heuristics with Nielsen's heuristics for various reasons. They are claimed to be applicable to any interface. In contrast, other related heuristics (see Section 2) are designed to be applicable to specific domains. While the characteristics of these domains might overlap with those of ITSM, they will not address all important aspects of the ITSM domain. Furthermore, those who used Nielsen's heuristics can be considered to be a control group who used standard heuristic evaluation. This allowed us to show improvements over the standard. In a future work, ITSM heuristics can be compared to other domain-specific heuristics or a combination of them.

While one of the challenges in our study was recruiting participants with HCI background, finding participants with both an HCI and computer security background was even more challenging. The majority (17 out of 24) of our participants had no research or professional computer security experience. Nielsen [35] suggests that domain expertise has an impact on the ability of evaluators to find usability problems. In future research, we plan to investigate the impact of the participants' computer security background on the number and severity of the problems they find.

We collected qualitative data during the post-evaluation feedback session. Analysis of data will help us better understand which ITSM and Nielsen's heuristics were particularly useful from the participants' viewpoint, and how they can be improved. Furthermore, more analysis can be performed on quantitative data collected during the study to determine the performance of individual heuristics. Due to page limitations here, we consider this further analysis to be the focus of future research.

Finally, we plan to focus on one of the problems identified, and modify the IdM system interface to address it. We will then perform a comparative evaluation of the two systems to investigate the effectiveness of our changes.

8. CONCLUSION

In this paper, we presented heuristics for the evaluation of ITSM tools. The goal of these heuristics is to find usability problems that hinder tool use in the complex and collaborative ITSM context. To examine the applicability of the heuristics, we compared their use for the evaluation of an IdM system. Our results show that the output of heuristic evaluation of an IdM system using ITSM heuristics contained more severe problems than the output of the evaluation of the system using Nielsen's heuristics. Comparing the individual performance of evaluators also showed that the severity of the problems found by evaluators in the ITSM condition was higher compared to that of the Nielsen condition. Furthermore, our participants found the ITSM heuristics to be as relevant, easy to apply, and easy to learn as Nielsen's heuristics. The results of our evaluation also shed light on the use of the heuristic evaluation in general to evaluate a complex domain specific system. Compared to prior literature on heuristic evaluation, our results show that evaluation of the IdM system requires more evaluators. Additionally, the complexity and scale of the system can result in a lack of overlapping problems between evaluators. Finally, our results show that Nielsen's heuristics can also be effective in finding a class of problems in ITSM tools that are not found by ITSM heuristics. Therefore, we recommend using a combination of Nielsen's and ITSM heuristics.

The proposed heuristics are a component of tool usability evaluation, but we recommend employing other techniques in the overall usability engineering lifecycle. Heuristic evaluation can be used as a low-cost method to find usability problems in preliminary prototypes or actual ITSM tools. These problems can be further investigated by a user study or a contextual inquiry session. Design guidelines can then be used to address identified problems.

Acknowledgements

We thank study participants for their time, and members of the Laboratory for Education and Research in Secure Systems Engineering (LERSSE) who provided valuable feedback on the earlier drafts of this paper. Cormac Herley provided feedback in May 2010 on the design of the project. We thank Robert Biddle for his insightful feedback on several occasions throughout the project. Comments from the anonymous reviewers helped us to improve the paper. This research has been partially supported by CA Technologies and by the Canadian NSERC ISSNet Internetworked Systems Security Network Program.

9. REFERENCES

[1] K. Baker, S. Greenberg, and C. Gutwin. Heuristic evaluation of groupware based on the mechanics of collaboration. *Lecture Notes in Computer Science*, pages 123–140, 2001.

[2] K. Baker, S. Greenberg, and C. Gutwin. Empirical development of a heuristic evaluation methodology for shared workspace groupware. In *Proceedings of the 2002 ACM conference on Computer supported cooperative work*, CSCW '02, pages 96–105, New York, NY, USA, 2002. ACM.

[3] 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.

[4] B. Beal. IT security: the product vendor landscape. *Network Security*, 2005(5):9–10, 5 2005.

[5] D. Botta, K. Muldner, K. Hawkey, and K. Beznosov. Toward understanding distributed cognition in IT security management: the role of cues and norms. *Int. Journal of Cognition, Technology & Work*, Online First, September 2010. DOI: 10.1007/s10111-010-0159-y, 2010.

[6] 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.

[7] J. M. Carroll, D. C. Neale, P. L. Isenhour, M. B. Rosson, and D. S. McCrickard. Notification and awareness: synchronizing task-oriented collaborative activity. *International Journal of Human-Computer Studies*, 58(5):605 – 632, 2003. Notification User Interfaces.

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

[9] P. Dourish and D. Redmiles. An approach to usable security based on event monitoring and visualization. In *NSPW '02: Proceedings of the 2002 workshop on New security paradigms*, pages 75–81, New York, NY, USA, 2002. ACM.

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

[11] T. Erickson and W. A. Kellogg. Social translucence: an approach to designing systems that support social processes. *ACM Trans. Comput.-Hum. Interact.*, 7(1):59–83, 2000.

[12] G. Fitzpatrick, T. Mansfield, and S. Kaplan. Locales framework: Exploring foundations for collaboration support. In *CHI'96 Sixth Australian Conference on Computer-Human Interaction*, Hamilton, New Zealand, November 24–27, 34–41 1996.

[13] A. Gagné, K. Muldner, and K. Beznosov. Identifying differences between security and other IT professionals: a qualitative analysis. In *HAISA'08: Human Aspects of Information Security and Assurance*, pages 69–80, Plymouth, England, July 8-9 2008.

[14] J. R. Goodall, W. G. Lutters, and A. Komlodi. I know my network: Collaboration and expertise in intrusion detection. In *CSCW '04*, pages 342–345, November 2004.

[15] S. Greenberg, G. Fitzpatrick, C. Gutwin, and S. Kaplan. Adapting the locales framework for heuristic evaluation of groupware. *Australian Journal of Information Systems*, 7(2):102–108, 2000.

[16] C. Gutwin and S. Greenberg. The mechanics of collaboration: developing low cost usabilityevaluation methods for shared workspaces. *IEEE 9th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2000.(WET ICE 2000). Proceedings*, pages 98–103, 2000.

[17] E. M. Haber and J. Bailey. Design guidelines for system administration tools developed through ethnographic field studies. In *CHIMIT '07: Proceedings of the 2007 symposium on Computer human interaction for the management of information technology*, pages 1:1–1:9, New York, NY, USA, 2007. ACM.

[18] H. R. Hartson, T. S. Andre, and R. C. Williges. Criteria for evaluating usability evaluation methods. *International Journal of Human-Computer Interaction*, 13(4):373–410, 2001.

[19] K. Hawkey, D. Botta, R. Werlinger, K. Muldner, A. Gagne, and K. Beznosov. Human, Organizational, and Technological Factors of IT Security. In *CHI'08 extended abstract on Human factors in computing systems*, pages 3639–3644, Florence, Italy, 2008.

[20] 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.

[21] P. Jaferian, D. Botta, F. Raja, K. Hawkey, and K. Beznosov. Guidelines for Designing IT Security Management Tools. In *CHIMIT '08: Proceedings of the 2008 symposium on Computer Human Interaction for the Management of Information Technology*, pages 7:1–7:10. ACM, 2008.

[22] R. Jeffries, J. R. Miller, C. Wharton, and K. Uyeda. User interface evaluation in the real world: a comparison of four techniques. In *CHI '91: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 119–124, New York, NY, USA, 1991. ACM.

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

[24] V. Kaptelinin, B. Nardi, S. Bodker, J. Carroll, J. Hollan, E. Hutchins, and T. Winograd. Post-cognitivist HCI: second-wave theories. In *CHI '03: CHI '03 extended abstracts on Human factors in computing systems*, pages 692–693, New York, NY, USA, 2003. ACM.

[25] S. Kesh and P. Ratnasingam. A knowledge architecture for it security. *Commun. ACM*, 50(7):103–108, 2007.

[26] A. G. Kotulic and J. G. Clark. Why there aren't more information security research studies. *Information & Management*, 41(5):597–607, 2004.

[27] K. Kuutti. *Activity theory as a potential framework for human-computer interaction research*, pages 17–44. Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.

[28] P. P. Maglio, E. Kandogan, and E. Haber. Distributed cognition and joint activity in collaborative problem solving. In *Proceedings of the Twenty-fifth Annual Conference of the Cognitive Science Society*, 2003.

[29] J. Mankoff, A. K. Dey, G. Hsieh, J. Kientz, S. Lederer, and M. Ames. Heuristic evaluation of ambient displays. In *Proc. CHI '03*, pages 169–176, New York, NY, USA, 2003. ACM.

[30] M. J. Muller and A. McClard. Validating an extension to participatory heuristic evaluation: quality of work and quality of work life. In *CHI '95: Conference companion on Human factors in computing systems*, pages 115–116, New York, NY, USA, 1995. ACM.

[31] B. A. Nardi, editor. *Context and consciousness: activity theory and human-computer interaction.* Massachusetts Institute of Technology, Cambridge, MA, USA, 1995.

[32] B. A. Nardi, S. Whittaker, and H. Schwarz. NetWORKers and their activity in intensional networks. *Computer Supported Cooperative Work (CSCW)*, 11(1):205–242, 2002.

[33] D. C. Neale, J. M. Carroll, and M. B. Rosson. Evaluating computer–supported cooperative work: models and frameworks. In *CSCW '04*, pages 112–121. ACM Press, 2004.

[34] J. Nielsen. How to conduct a heuristic evaluation. http://www.useit.com/papers/heuristic/heuristic_evaluation.html.

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

[36] J. Nielsen. Usability inspection methods. In *CHI '94: Conference companion on Human factors in computing systems*, pages 413–414, New York, NY, USA, 1994. ACM.

[37] J. Nielsen and T. K. Landauer. A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT '93 and CHI '93 conference on Human factors in computing systems*, CHI '93, pages 206–213, New York, NY, USA, 1993. ACM.

[38] J. Nielsen and R. Molich. Heuristic evaluation of user interfaces. In *CHI '90: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 249–256, New York, NY, USA, 1990. ACM.

[39] D. A. Norman. *Cognitive Engineering.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.

[40] D. A. Norman. Cognitive artifacts. *Designing interaction: Psychology at the human-computer interface*, pages 17–38, 1991.

[41] G. Olson and T. Moran. Commentary on" Damaged Merchandise?". *Human-Computer Interaction*, 13(3):263–323, 1998.

[42] D. Pinelle, N. Wong, and T. Stach. Heuristic evaluation for games: usability principles for video game design. In *Proc. CHI '08*, pages 1453–1462, New York, NY, USA, 2008. ACM.

[43] P. Rabardel and G. Bourmaud. From computer to instrument system: a developmental perspective. *Interacting with Computers*, 15(5):665 – 691, 2003. From Computer Artefact to Instrument for Mediated Activity. Part 1 Organizational Issues.

[44] Y. Rogers. Ghosts in the network: distributed troubleshooting in a shared working environment. In *CSCW '92: Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, pages 346–355, Toronto, ON, Canada, 1992. ACM.

[45] M. B. Rosson and J. M. Carroll. *Usability engineering: scenario-based development of human-computer interaction.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2002.

[46] P. Sarbanes. Sarbanes-Oxley Act of 2002. In *The Public Company Accounting Reform and Investor Protection Act. Washington DC: US Congress*, 2002.

[47] B. Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1997.

[48] B. Shneiderman. Creating creativity: user interfaces for supporting innovation. *ACM Trans. Comput.-Hum. Interact.*, 7(1):114–138, 2000.

[49] A. Sutcliffe and B. Gault. Heuristic evaluation of virtual reality applications. *Interacting with Computers*, 16(4):831 – 849, 2004. Human Computer Interaction in Latin America.

[50] D. Te'eni, J. Carey, and P. Zhang. *Human Computer Interaction: developing effective organizational information systems.* Wiley, 2007.

[51] R. S. Thompson, E. M. Rantanen, W. Yurcik, and B. P. Bailey. Command line or pretty lines?: comparing textual and visual interfaces for intrusion detection. In *CHI '07: Proceedings of the SIGCHI conference on Human factors in computing systems*, page 1205, San Jose, CA, USA, 2007. ACM.

[52] N. F. Velasquez and A. Durcikova. Sysadmins and the need for verification information. In *CHiMiT '08: Proceedings of the 2nd ACM Symposium on Computer Human Interaction for Management of Information Technology*, pages 1–8, New York, NY, USA, 2008. ACM.

[53] N. F. Velasquez and S. P. Weisband. Work practices of system administrators: implications for tool design. In *CHiMiT '08: Proceedings of the 2nd ACM Symposium on Computer Human Interaction for Management of Information Technology*, pages 1–10, New York, NY, USA, 2008. ACM.

[54] K. J. Vicente. HCI in the global knowledge-based economy: designing to support worker adaptation. *ACM Trans. Comput.-Hum. Interact.*, 7(2):263–280, 2000.

[55] K. Vredenburg, J.-Y. Mao, P. W. Smith, and T. Carey. A survey of user-centered design practice. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 471–478, New York, NY, USA, 2002. ACM.

[56] R. Werlinger, K. Hawkey, and K. Beznosov. An integrated view of human, organizational, and technological challenges of IT security management. *Journal of Information Management & Computer Security*, 17(1):4–19, 2009.

[57] R. Werlinger, K. Hawkey, D. Botta, and K. Beznosov. Security practitioners in context: Their activities and interactions with other stakeholders within organizations. *International Journal of Human-Computer Studies*, 67(7):584–606, March 2009.

[58] D. Zager. Collaboration as an activity coordinating with pseudo-collective objects. *Computer Supported Cooperative Work (CSCW)*, 11(1):181–204, 2002.

[59] J. Zhang, T. R. Johnson, V. L. Patel, D. L. Paige, and T. Kubose. Using usability heuristics to evaluate patient safety of medical devices. *Journal of Biomedical Informatics*, 36(1-2):23 – 30, 2003. Patient Safety.

[60] A. T. Zhou, J. Blustein, and N. Zincir-Heywood. Improving intrusion detection systems through heuristic evaluation. In *in IEEE Canadian Conf. on Electrical B. and Computer Engineering (CCECE)*, pages 1641 – 1644, 2004.

Appendix 1- Background Questionnaire

PART I - General Information

Gender

☐Male ☐Female

Age		**Last educational degree**	
Major		**Current Occupation**	

PART II – Human computer interaction background

- How many years of professional or research experience do you have in the area of Human Computer Interaction (HCI)?
- Do you have formal training in human computer interaction (university courses, tutorials, workshops)? Please answer with "Yes" or "No". If your answer is "Yes" please specify the list of courses.
- Do you have professional experience in the area of human computer interaction? Please answer with "Yes" or "No". If your answer is "Yes", provide a summary of your experience in this field.
- Do you have research experience in the area of human computer interaction? Please answer with "Yes" or "No". If your answer is "Yes", provide a summary of your experience in this field.
- Have you specifically been trained to perform a heuristic evaluation?
  ☐Yes ☐No
- Have you performed heuristic evaluation before? Please answer with "Yes" or "No". If your answer is "Yes", provide a summary of your previous experience in performing heuristic evaluation including number and type of systems you have evaluated

Part III – Computer security background

- How many years of research experience do you have in the area of computer security?
- How many years of professional experience do you have in the area of computer security?
- Do you have formal training in computer security (university courses, tutorials, workshops)? Please answer with "Yes" or "No". If your answer is "Yes" please specify the list of courses.
- Do you have professional experience in the area of computer security? Please answer with "Yes" or "No". If your answer is "Yes", provide a summary of your experience in this field.
- Do you have research experience in the area of computer security? Please answer with "Yes" or "No". If your answer is "Yes", provide a summary of your experience in this field.
- Have you ever worked in an organization that uses role-based access control to manage users and their privileges?
  ☐Yes ☐No
- Have you ever used to manage users and their privileges using role-based access control?
  ☐No ☐Yes in a small group (less than 10 users)

  ☐Yes in a small organization (between 10 to 50) ☐Yes in a large organization (more than 50 users)

Appendix 2- Evaluation Guide

Evaluation Steps

1. Go through the list of heuristics to have a sense of each.
2. Read the description of the scenario and understand the business logic.
3. Perform each task as described on the IdM system.
4. Identify usability problems while doing each task or after finishing the task. For each problem, please record the task in which you found the problem, and the heuristic with which you identified the problem. Use the scenario description and heuristics to check if the system supports the activity described in the scenario.
5. Please record the problems in Here
6. If you want to edit any of the identified problems which you already entered in the form, you can do it from Here.

Recommendations

- I recommend exploring the IdM system first before going through specific tasks.
- When performing the tasks, you should login with different users as described in the scenario For example, if you want to login as a Security team member or a particular manager, use the organizational chart to find the right person to login as.
- If you want to login as a user, the user name is: first name + the first letter of the family name (e.g. James Beers -> jamesb) and password is "q1w2e3".
- If you couldn't finish a task or get the desired result, don't worry! The real user may have the same problem.
- If you face any problems, you can ask the person conducting the study.

Scenarios

Description of the actors

Steve Barlow is an employee in the operations department. He is responsible for reviewing the information about the contractors. He does not have technical information about the Identity Management System or role based access control.

James Beers is the manager of operations. His day mostly involves meeting with different stakeholders in the organization. He receives lots of emails and telephone calls every day therefore he needs lots of discipline to prioritize his tasks. He does not know technical information about the IdM system or the role based access control, but he knows if an employee should have access to some resources or not.

Kevin Klien and **Sandra Tsai** are both members of the security team. They are responsible for managing access to the resources in the organization and solving problems of different stakeholders. They work in the same office and they are very busy with these tasks.

Larry Gomez is a contractor that needs to work in NeteAuto for one month. He barely knows the structure of the company or other employees.

Scenario 1: Self-serve user registration

Larry Gomez is a contractor for the NeteAuto Company and just started his job. To be able to access the Internet, he wants to create a user account in the IdM system. Using company's intranet, he finds the link to the IdM system and creates a new user account.
His request is directed to the security department. All members of the security team receive the request in their task list, and they can review or edit the user information. Finally they can approve, reject, or reserve the task (reserving the task will remove it from the worklist of other security admins).

Steps for performing the scenario:

Larry Gomez accesses the IdM system. He uses the "create an account" link on the IdM login page and enters the required information.
Kevin Klien receives the request and after reviewing the information approves the request.

Scenario 2: Bulk loader

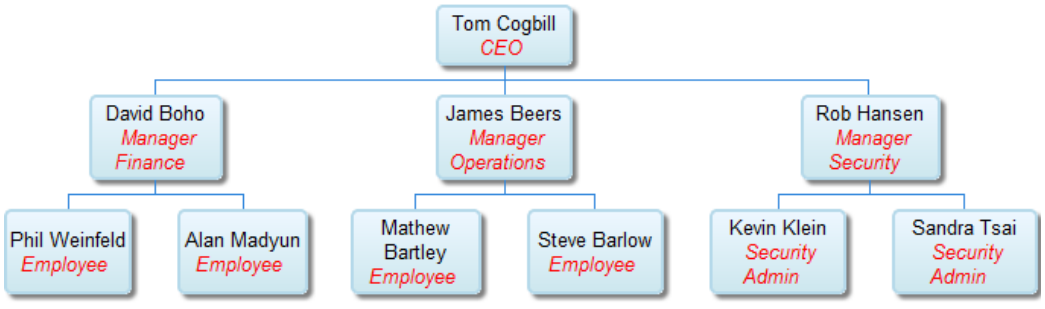
When an employee is hired by the NeteAuto Company, or information about an employee changes, or an employee leaves the company, the first system in which the changes are reflected is the HR (Human Resources) system. The HR system is separate from the IdM system; therefore, the changes in the HR system need to also be applied to the IdM system. Transferring changes from the HR system to the IdM system is performed by the security team. The security team receives a file containing all the changes (additions, modifications, and deletions) from the HR system.
Every morning, Sandra Tsai, a member of the security team, downloads the HR file from the HR website and uploads the file to the IdM system to apply all the changes made in the HR system.
She uses the "Bulk Loader" feature in the IdM system to upload the HR file. Then she configures the system to respond to different actions defined in the HR file. An important step after submitting the changes is to review the result of submission.
She goes through the system logs, finds appropriate records, and identifies and fixes the problems, if any.
Based on the organization's policy, if the number of changes in the HR file is more than 500, applying the changes should be postponed until further clarification by HR.

Steps for performing the scenario:

Sandra Tsai should first upload the HR file using the Bulk Loader in the System tab. In the next screen she chooses which field in the HR file describes the action that should be performed for each row in the file (in the example HR file it is the “action” row). Also she chooses which field uniquely identifies each row in the HR file (in the example HR file it is the “%USER_ID%” row).
In the next screen she identifies the primary object that HR file contains (choose USER as the file contains user information) and the mapping between actions in the HR file and actions in the IdM system (choose “Create User”, “Modify User”, and “Delete User” for any create, modify, and delete actions respectively).

Scenario 3: Requesting a role
Steve Barlow is going on a last minute vacation. He realizes that he does not have the required privileges to delegate his tasks to Jason Halpin, another member of the operations department. He does not have any technical information about the privileges required to perform the delegation. But, he knows that he can generate request for privileges in the identity management system. Therefore, he uses the IdM system to write a request. In the request, he describes that he needs the ability to delegate his role to another employee in his department.
When Steve submits the request, his manager needs to approve it before the request is implemented. The manager uses the IdM system to review and approves the request.
Once the manager approves the request, the request is directed to a member of security team who reviews the request, and, if it does not conflict with the security policy of the organization, tries to implement the request. Implementing the request requires the security admin to understand the content of the request (in this case, learn that Steve wants to delegate his role) and find the appropriate role that corresponds to the request (in this case, the “Delegation Manager” role). Then he can add Steve Barlow as a member of that role.
Steps for performing the scenario:
Steve Barlow: generate the request using the “Users>Manage Users>Create Online Request” and then select himself as the target user. Then he can describe and submit his request.
James Beers (Steve’s manager): log into the IdM system. Identify, review, and approve the request.
Kevin Klien (or other members of Security): log into the IdM system. Identify, review, and implement the request.
To implement the request, he needs to modify the user and provision the user with the "Delegation Manager" role.

Scenario 4: Certification
As a part of the organization's policy, the security team should certify the roles of the employees in each department every 6 months. The security team uses a shared calendar to mark the dates that they should perform the certification and the deadline for finishing the certification. Each member in the security team is able to start a “Certification Process” in the IdM system.
When the certification date approaches, a member of the security team (Kevin Klein in this scenario) logs into the IdM system and chooses employees that should be certified.
The manager of each department receives the notification about certification of his employees. In this scenario, the manager of operations (James Beers) receives an email that he should certify the roles of the employees of operations department. James put the email in his todo list.
After a while, James logs into the IdM system and tries to certify the roles of the employees. For all of the employees, he checks the roles and validates if the employee should possess the role or not.
It is important for the manager to perform the certification before the deadline. If the certification does not happen before the deadline, all the uncertified roles will be revoked from the employees. Therefore, before the deadline, a member of the security team sends reminders to perform the certification.
On the certification deadline, a member of the security team ends the certification process.
Steps for performing the scenario:
Kevin Klein: Login to the IdM system. Go to the certification tab and start the certification process for the employees in the Operations department. Also, send reminders about the certification.
James Beers: Assume you are going to certify users in your department. Login to the IdM system and search for the users that require certification using “Users>Manage Users>Certify Users”. Select users one by one, go to the “Certify Roles” tab, review their roles, and approve them.
Sandra Tsai: Login to the IdM system and end the certification process.



Appendix 3- Problem Specification Form (ITSM condition)

Problem specification: Please specify the identified usability problem.

Task: Please specify the task in which you identified the problem.

- ○ (1) Self-serve user creation
- ○ (2) Bulk user creation
- ○ (3) Requesting a role workflow
- ○ (4) Certification Process

Heuristic: Please choose the heuristic using which you identified a problem. If you can't associate the problem with a heuristic, please choose "Can't Specify"

- ☐ 1- Visibility of activity status
- ☐ 2- History of actions and changes on artefacts
- ☐ 3- Flexible representation of information
- ☐ 4- Rules and constraints
- ☐ 5- Planning and dividing work between users
- ☐ 6- Capturing, sharing, and discovery of knowledge
- ☐ 7- Verification of knowledge
- ☐ Can't Specify

Appendix 4- Post-evaluation Questionnaire (ITSM condition)

Please indicate the extent to which you agree with each of the following statements about the heuristics that you used in this study by using the scale below: 1= Strongly Agree, 2= Agree, 3= Undecided or unsure, 4= Disagree, 5= Strongly Disagree

The heuristics were very useful in finding all of the problems that you found in the IdM system.

	1	2	3	4	5	
Strongly Agree	○	○	○	○	○	Strongly Disagree

The heuristics were very easy to learn and understand.

	1	2	3	4	5	
Strongly Agree	○	○	○	○	○	Strongly Disagree

The heuristics were very easy to apply on the IdM system.

	1	2	3	4	5	
Strongly Agree	○	○	○	○	○	Strongly Disagree

The following heuristics were very useful in identifying problems that you found in the IdM system:

	1	2	3	4	5
1- Visibility of activity status	○	○	○	○	○
2- History of actions and changes on artifacts	○	○	○	○	○
3- Flexible representation of information	○	○	○	○	○
4- Rules and constraints	○	○	○	○	○
5- Planning and dividing work between users	○	○	○	○	○
6- Capturing, sharing, and discovery of knowledge	○	○	○	○	○
7- Verification of knowledge	○	○	○	○	○

The following heuristics were very easy to learn and understand.

	1	2	3	4	5
1- Visibility of activity status	○	○	○	○	○
2- History of actions and changes on artifacts	○	○	○	○	○
3- Flexible representation of information	○	○	○	○	○
4- Rules and constraints	○	○	○	○	○
5- Planning and dividing work between users	○	○	○	○	○
6- Capturing, sharing, and discovery of knowledge	○	○	○	○	○
7- Verification of knowledge	○	○	○	○	○

The following heuristics were very easy to apply on the IdM system.

	1	2	3	4	5
1- Visibility of activity status	○	○	○	○	○
2- History of actions and changes on artifacts	○	○	○	○	○
3- Flexible representation of information	○	○	○	○	○
4- Rules and constraints	○	○	○	○	○
5- Planning and dividing work between users	○	○	○	○	○
6- Capturing, sharing, and discovery of knowledge	○	○	○	○	○
7- Verification of knowledge	○	○	○	○	○