Towards Understanding IT Security Professionals and Their Tools

David Botta, Rodrigo Werlinger, André Gagné
Konstantin Beznosov, Lee Iverson, Sidney Fels, Brian Fisher

University of British Columbia, Vancouver, Canada
{botta,rodrigow,andreg,beznosov,leei,sfels}@ece.ubc.ca, fisher@cs.ubc.ca

ABSTRACT

We report preliminary results of our ongoing field study of IT professionals who are involved in security management. We interviewed a dozen practitioners from five organizations to understand their workplace and tools. We analyzed the interviews using a variation of Grounded Theory and predesigned themes. Our results suggest that the job of IT security management is distributed across multiple employees, often affiliated with different organizational units or groups within a unit and responsible for different aspects of it. The workplace of our participants can be characterized by their responsibilities, goals, tasks, and skills. Three skills stand out as significant in the IT security management workplace: inferential analysis, pattern recognition, and *bricolage*.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Interaction Styles*; H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*Collaborative Computing*

General Terms

Security Management, Design, Human Factor

Keywords

Security Management, Ethnography, Security Tasks, Security Tools, Usable Security, Collaboration

1. INTRODUCTION

The management of information technology (IT) security in organizations is an enormous, difficult, and costly problem, with over US$100 billion USD to be spent by organizations worldwide solely on IT security in 2007.[1] The challenges of IT security management (ITSM) arise from the increasingly high numbers of application instances, resources, and users interacting with business processes that are growing in complexity. With small- and medium-sized businesses starting to outsource their IT security to managed security service providers (MSSP), which provide security management for multiple organizations, the scale of the problem is only expected to grow.

Yet little is known about IT security professionals, their roles and responsibilities with regards to security management, and how effective their tools and practices are in protecting organizations and employees while still allowing productive collaborative work in the context of real environments [14, 4]. As a result, HCISec researchers and tool developers lack an understanding of what support is needed for those who manage IT security, which tools they use, and how they use those tools in their work [26, 11].

This paper is about a field study with the objective to *build* theory about how IT professionals practice security management, given their human limitations, and the realities of their workplaces. We report here on our those aspects of our early results that are of direct relevance to the interests of HCISec researchers and tool developers: the distributed nature of ITSM; divisions of responsibility that characterize the ITSM workplace; an inventory of tools used to accomplish various ITSM tasks; the kinds of skill necessary to perform many ITSM tasks; and what made tools more (or less) effective for our participants.

The field study is the first phase of the project *HOT Admin: Human, Organization, and Technology Centred Improvement of IT Security Administration.*[2] The project investigates methods and techniques for developing better tools for managing IT security from the perspective that human, organizational and technological factors influence the ability of security practitioners to do their job well.

We employed an ethnographic approach in this study. Our data collection comprised an initial questionnaire followed up with an audio-recorded semi-structured interview for some of the subjects in their workplace, also known as *contextual inquiry*. We administered the questionnaire to 24 participants and conducted 14 semi-structured interviews. We asked administrators to tell us how security issues are handled—distribution of responsibilities, tools used, and how the security management task plays out within

[1] This estimate is based on reports by Forrester Research, Inc. that 7-9% of organizations' IT budgets will be spent solely on security [15], with US$1.55 trillion to be spent on IT worldwide in 2007 [3].

[2] `hotadmin.org`

Symposium On Usable Privacy and Security (SOUPS) 2007, July 18-20, 2007, Pittsburgh, PA, USA.

the context of their particular organization. We took a mixed approach to analysis, using open-coding and pre-designed themes. The open-coding approach was a variation of Grounded Theory (GT) [10]. Our use of pre-designed themes was closely related to case-study approaches [24], in that we organized relevant data into predetermined themes based on the pragmatics of the security management tasks in context.

We found that IT security management is a job distributed among several professionals—or even groups of them with dedicated "coordinators"—scattered throughout organizational units. An expert in a distinct IT technology, each group member is responsible for particular IT security aspects, systems, or devices, but commonly also has responsibilities outside of security. Our analysis showed that the workplace of our participants can be characterized by the *responsibilities* that determine activities of the participants, *goals* of the activities, *tasks* they perform to achieve the goals, and *skills* needed for the tasks. Three skills stand out as significant in the IT security management workplace: inferential analysis, pattern recognition, and *bricolage*.

The rest of the paper is organized as follows. The next section discusses related work. Section 3 describes the research questions, as well as data collection and analysis. Section 4 reports the results. Section 5 analyzes the results. We conclude and discuss future work in Section 6.

2. RELATED WORK

Barrett et al. [2] use ethnographic methods to study system administrators in context "to find opportunities for supporting work through appropriate design and technology." Since a good deal of security management is done by system administrators, their findings are relevant to ours. Although these findings touch upon a broad spectrum of IT administration (e.g., databases, web servers, operating systems), they can not necessarily be directly used to understand the practices, tasks, and needs of IT practitioners who manage security.

Kandogan and Haber [14] used ethnographic methods to study security administrators in particular, and propose relevant directions for tool development. Although our study is similar to theirs, we focused on modeling the workplace of security professionals. By proposing such a model, we aim at both understanding the relationship between tasks and tools, and assessing the effectiveness and usability of security tools.

Björck [4] uses Grounded Theory to empirically answer two research questions: (1) *"What problems do organisations face and what processes do they go through as they are aiming to establish a balanced management system for information security?"* and (2) *"What perceptions do information security managers hold as regards the management of information security in organisations?"* The data for Björck's study came from semi-structured interviews with 8 IT security managers, 13 consultants and 8 auditors—29 in total—in Swedish companies. Unlike Björck, we did not study organizational behavior in relation to IT security management. Rather, we focused on the practices, tasks, and tools that security practitioners use to manage IT security.

Wool [23] uses quantitative analysis of sets of firewall rules to positively correlate errors and rule-set complexity. We share Wool's investigation of errors but do not report our error-related findings in this paper.

Our study follows Zurko et al. [25] in explicitly placing security and usability as peer goals. We extend this commitment to usability to include an awareness of the practice of security in context. To illustrate, it is possible that a good technology that is well implemented could be rejected in favor of a relatively inadequate technology that is poorly implemented, because the latter includes a feature to compose reports for management—an aspect that has nothing to do with the technology *per se*, but rather with the context of organizational demands on the IT security professional.

Zurko and Simon [26] establish the usability of security technology as an equivalent goal to the technology of security, because people are the weakest link in the security chain. Both Zurko and Simon [26] and Holmstrom [11] develop technology that is justified by scenarios. Zurko and Simon construct Use Case scenarios, whereas Holmstrom develops scenarios from interviews and focus groups. The technologies are then subjected to usability testing. We expect that our ethnographic study of IT security professionals in context will provide further scenarios for technology development.

3. ETHNOGRAPHIC STUDY OF IT SECURITY PROFESSIONALS

Our study of the situated practice of IT security management required ethnographic methods to obtain rich accounts of IT security management. This section describes our research questions, the kind of data that we needed to answer them, the kind of participants we engaged, how we recruited them, the nature of our study instrument, and, finally, how we analyzed the data.

3.1 Research Questions

Our ethnographic approach was intended to elicit the human, organizational, and technical (HOT) aspects of security management so that we can design better tools and techniques for practitioners. Our first step, reported here, was to discover how IT security professionals perceive their jobs and the tools they use [5]. We pursued this objective in four stages: (1) recruitment; (2) initial questionnaire; (3) semi-structured interviews; and (4) data analysis. Each step is discussed in detail below.

3.2 Data Collection

Twenty four participants completed the questionnaire, and fourteen semi-structured interviews were conducted. Of the participants interviewed, most had worked in their current position for around five years. The longest someone had held their position was thirteen years, while the shortest was two years. In addition to their post-secondary education, all of the participants had professional training, including certifications for technical specializations (e.g., CISSP) or vendor certifications (e.g., MSCE). The one participant who did not have a formal post-secondary education had seven certifications. The number of machines (workstations and servers) administered by the non-managerial interviewees ranged from one dozen to 3,800.

3.2.1 Recruitment

We approached postsecondary educational institutions, research organizations, financial, insurance, and energy organizations in Greater Vancouver, Canada for our study.

There are three key challenges we observed that we had to overcome in recruiting subjects: (1) participation in the study was seen by the chronically overworked IT professionals, and especially by their supervisors, as an uncompensated burden, (2) the potential disclosure of IT security procedures, practices, and even tools in use went against common organizational culture of carefully restricting outside parties access to such details, and (3) since our participants were the backstage people whose contact information was not published on the company web sites or other publicly accessible sources, just finding ways to make first contact with them would be next to impossible without buy-in from the gatekeepers, i.e. management personnel.

To address the first challenge, we developed a graduated recruitment strategy so that the work burden was minimal to begin with. We initially asked potential participants only to answer a short questionnaire, the final question of which asked if the participant is willing to give a one-hour interview. At the end of this contextual interview, we asked some participants if they would be willing to allow us to shadow them in their workplace.

Graduated recruitment also helped in building trust between participants and the researchers in order to overcome the second challenge. We also actively educated potential participants about the purely academic (not commercial) and worthwhile goals of the HOT Admin project and the study itself. In addition, prior background of the principle investigator as a security professional himself seemed to aid with both (a) building trust through speaking the language (and jargon) of IT security, (b) developing professional contacts.

To address the third challenge, we used two approaches. Some participants were recruited directly, through professional contacts of the research team. Project team members developed and maintained such contacts by participating in the meetings of a regional security special interest group and presenting at a regional forum for IT security professionals. Although professional contacts ended up being most effective in the recruitment, they were too few.

To recruit other participants, we contacted managers of IT departments and met or interviewed them to solicit their cooperation. With their cooperation, we asked for recommendations of employees they felt would be knowledgeable and/or were involved with security management in their organization. In all cases, we obtained—directly or through the participants—management permission before involving our participants in the study.

Once identified, we contacted participants by e-mail. Our letter of first contact contained a brief description of the project and its goals, its policy about the privacy of the participants and the confidentiality of the collected data, and an invitation to complete the online questionnaire.

3.2.2 Questionnaire

The questionnaire was included in the tail of the e-mailed contact letter. Interested participants responded by replying with the completed answers within the body of the e-mail, or clicking on a link with the web version of the questionnaire. We wanted to provide this simple and convenient interface for responding to the questionnaire, because we expected the potential participants to be unable to devote much time or attention to a questionnaire [14]. It turned out that most participants preferred the web version, despite the popularity of text-only interfaces among them, as our results in Section 4.5 indicate.

The pilot-tested questionnaire had 21 questions ranging from general background and responsibilities to questions about the IT system and security management in addition to requesting participation in the follow-up interview. established through pilot testing. The questionnaire was not intended to gather quantitative data; rather, it was used to gather information that would help us better focus the semi-structured interview. For example, if in the questionnaire the participant mentioned interacting significantly with other individuals in the organization, we would be alerted to ask about the nature of these interactions.

The questionnaire was administered to 24 participants from various sectors. Table 1 shows the demographics of the 14 participants who agreed to be interviewed.

3.2.3 Semi-Structured Interview

The semi-structured interview allowed participants to tell stories that provided information beyond the current situation or time-frame. The interviewer had the opportunity to inquire about a wide range of aspects of security management, from minute routine details to long-term goals.

The following is a small sample of the questions comprising our semi-structured interview:

- What did you do yesterday?
- How do you interact with different types of people during the course of your work?
- Is there anything special about your organization that makes IT security administration more difficult; for example, a rapid turnover of users, or special relationships with other organizations, or something else?
- What do you wish for in your tools?

As is common with semi-structured interviews, the format and number of questions changed as we gained experience with this particular set of participants. Completing interviews within the promised time limit proved problematic, and so we substantially reduced the number of interview questions. Depending on the job roles that the interviewees played, we found it useful to quickly move to the topic of tools—preferences, dislikes, difficulties, wishes—because stories about tool use (1) tended to be detailed and concrete, and (2) led easily into detours concerning communication with other people, prioritization of tasks, and organizational idiosyncrasies.

3.3 Data Analysis

The discursive nature of the data we collected, combined with a lack of pre-existing theories of ITSM *per se* suggested a bottom-up approach such as Grounded Theory (GT) [10]. In classical Grounded Theory, a theory is developed from the data (grounded in the data) through coding of observed behaviors without reference to pre-existing theory. Insight comes about through reviewing the mapping of codes to data, inferring core variables, and building theory.

It soon became apparent that, while we lacked a specific theory of ITSM workplace, there was a good deal of information available on the nature of security threats and countermeasures, and on varieties of security tools and how they are used. There was also a substantial business literature on

Table 1: Demographics of the interviewed participants.

Job Position	Organization type	% of time spent on IT security	Years in organization
Security Specialist / Business Continuity Process Specialist	Banking	N/A[a]	N/A
Tech Specialist II	Insurance	20	5
Network Security Manager/System Administrator/Videoconferencing	Research Organization	10	25
Application Programmer		40	7
Director, IT Services	Postsecondary Educational Institution	N/A	5
Information Security Officer		N/A	14
IT Security Officer		N/A	17
Senior Systems Analyst		25	14
Senior Systems Analyst		N/A	3
IT Security Officer		N/A	N/A
Security Analyst		N/A	7
System Administrator		60	8
Network/Security Lead		40-60	4
Systems Analyst		20	20

[a]N/A — Exact information was not obtained.

structures of organizations and organizational behavior, and general work on social cognition and communication. All of these seemed relevant to the distribution of security responsibilities among multiple professionals that our participants reported.

Since there was a good deal of existing theory, and observation alone would be unlikely to reveal the social and organizational factors characteristic of the organization, in our methodology we adapted GT to take into account our understanding of the security administration tools and tasks together with a general framework for social cognition, Clark's [6] theory of psycholinguistic pragmatics. This perspective allowed us to characterize behavior in terms of basic principles of human communication: the need to achieve a shared understanding of the workplace situation, the constraints imposed by the organization and the tasks performed by individuals in it, and the communication mechanisms by which that shared understanding comes about. We can consider this approach a coding strategy on the order of open coding, axial coding etc.; however, it is a departure from GT in its use of a pre-existing coding strategy.

The first step in our analysis was for two researchers to code a sample of the interviews independently using open coding. The second step was to merge the codes and categories. Analysis had to be championed by an appointed team member, who also coordinated parallel analyses.

Initially we attempted to utilize team coding features of a particular qualitative analysis application (Qualrus [21]) to coordinate the coding, with mixed results. We found that the most effective method for merging was to organize the information in a table, with one column describing the tasks mentioned by security practitioners and another column the tool(s) used (if any) to perform such tasks. This approach was very effective in accounting for regularities in the data, and as a mechanism for reaching an agreement among the researchers about, for instance, categorization of the tasks that the security practitioners reported.

Sessions proceeded as follows: One researcher related their analysis of a specific interview using the table *tasks and tools* in a text editor. After receiving feedback from the group, the tasks were classified into common codes and categories, which were then tested for their ability to conceptualize data from subsequent interviews. In this process, sets of interviews were randomly assigned to a given researcher for coding. Each analysis was cross checked during meetings, where the codes and categories used for analysis were scrutinized by the project team, and by having selected interviews recoded by another researcher for comparison, and any differences discussed and rationalized. This triangulation process resulted in refinements, rather than dramatic changes, to the initial list of categories.

The result of the data analysis is shown in the next section. Specifically, Section 4.2 explains the categories chosen to describe the tasks of security practitioners, their responsibilities and use of tools. It is important to note that our analysis of organizational and human dimensions of security administration is still in progress. Our choice of Clark's framework for analysis of communication will provide the conceptual structure of this next phase of analysis, as well as allowing us to better address the ways in which the interviews we conducted with the security practitioners may be affected by the social and communicative aspects of the the interview situation (e.g., demand characteristics) as compared to less interactive methods of data elicitation [7].

4. RESULTS

The interviews enabled us to gain insight into the workplace of our participants, the kinds of activities they engage in on a daily basis in managing IT security, the tools they use, and the skills required.

4.1 Security Management Teams

Initially we aimed at studying mainly those who consider themselves *security administrators*. Perhaps surprisingly, we found it difficult to find IT personnel with "security administrator" as their job title, or who would describe themselves as such. Instead, we found system, application, business, or technical analysts, system administrators, ap-

plication programmers, auditors, IT managers, security and network leads, etc., but no *security administrators*. As one participant explained about the differences, "*I think a security administrator's job generally has an established set of procedures and polices and it's in their job description to administer the application of those procedures or policies . . .*"

We found that the job of security administration is only one of the goals of IT security management. Furthermore, "security administration" was not even articulated as a distinct responsibility of any of the participants we interviewed. Instead, it is intertwined with many other responsibilities IT security professionals have day in and day out. Some of these responsibilities extend beyond just security administration: "*. . . what makes me [a security] analyst is that I'm also involved in developing the policies and procedures. . . an analyst is also someone who's doing a certain amount of troubleshooting and someone who's, I guess, a little bit more portable in terms of what their daily responsibilities are going to be like.*" On the other hand, their other responsibilities are completely outside of IT security: "*[I provide] third-level support for some of my team; not my security team but my other team, I have to deal with other personnel as well to help them out.*"

The different goals of IT security management can be found by looking at the tasks that our participants undertook. For example, one participant had to "bring on a secondary unit" of a VPN server. This task is much more a "security administration" kind; they were responsible for bringing the server up and then checking that the settings were correct. But tasks with very different goals also came up, e.g., "to investigate employee violations of policy."

Furthermore, the management of IT security is not concentrated in the hands of any particular person or close group. We found that the job of security management is distributed across multiple employees, often affiliated with different organizational units or groups within a unit and responsible for different aspects of it, e.g., "*He's responsible for the firewall image on the control system. And then there's another two people who work with Windows systems and look after the antivirus products on Windows and they do some forensics and diagnostics on the Windows systems.*" There is typically one "coordinator"—not necessarily a manager—who commonly has more technical expertise in computer security and coordinates such collaborations: "*I have a security team that I work with. They don't report to me but I actually work with them and they sort of are represented by the different areas.*" Responsibilities of the security coordinators are directly related to security management, whereas only some responsibilities of others are relevant to security. The diversity of responsibilities, goals, tasks, and skills IT professionals involved in security management have could be the key to understanding the reasons behind the distributed structure of IT security management.

4.2 Workplace Characteristics

Our analysis showed that the workplace of our participants can be characterized by the *responsibilities* that determine activities of the participants, the *goals* of the activities, the *tasks* they perform to achieve the goals, and the *skills* needed for the tasks. For example, one participant was responsible for designing a solution for authenticating clients connected via either wireless or wired networks to a web server using passwords. In order to achieve the goal of setting up X.509 public key certificates to authenticate the SSL server to the clients, this security professional performed several tasks, including the following: (1) identifying the use of certificates as part of the solution to protect users' passwords; (2) finding documentation about generating certificates and understanding how they can be used in a local environment; (3) writing scripts to automate the generation of certificates; and (4) processing requests from users who requested certificates. In doing this work, the security professional exercised the skills of *bricolage* and *pattern recognition*. He was able to recognize how to automate aspects of the authentication, and to create an ad-hoc set of tools to carry it off. He was also familiar with the patterns of his networks and users, and could see how best to apply certificates within his organization.

We found that a responsibility can be of one of three kinds: (1) responding to events (e.g., responding to reports of security incidents); (2) designing a solution (e.g., developing policies and procedures, or evaluating how to mesh technology with the existing environment); (3) and maintaining a system (e.g., maintaining firewalls, VPN servers, remote access systems). Appendix A provides three stories synthesized from recollections of our participants. These stories illustrate in detail the three kinds of responsibilities and show how our participants use tools to accomplish related tasks.

In the following sections, we discuss the results of our analysis with regard to the tasks, skills, and tools of our participants.

4.3 Tasks

The main tasks performed by our participants, along with the tools they use for those tasks, are shown in Table 2. To search information about configurations and IT security in general, any browser and search engine sufficed. General purpose IT tools were usually used for monitoring (e.g., SmokePing), verifying configurations (e.g., SpamAssassin), executing re-configuration responses (network devices' operating systems), and updating operating systems. Some specific security tools were required, such as: antivirus software (e.g., Kasperski), vulnerability scanners (e.g., Nessus), intrusion detection systems (e.g., Snort), and fingerprinting tools (e.g., Nmap).

Two other important observations can be made from Table 2. At a glance, it would seem that the number of tools mentioned by the participants was not high. This can be misleading, because participants usually wrote scripts to complement the functionality of some tools (e.g., Snort), or to perform specific tasks (e.g., analysis of logs, correlation of events). One participant had accumulated about 2,000 scripts over 25 years. Regarding the complexity of the tasks, it would seem from the table that the output of the tools is enough to perform the tasks, e.g., the task "receive and process notifications." However, this is rarely the case. The output is filtered and re-filtered, and compared with output from other sources. The practitioner normally engages the skills of inferential analysis, pattern recognition, and *bricolage*, as described below.

4.4 Skills

Three skills stand out as significant in the IT security management workplace: inferential analysis, pattern recognition, and *bricolage*. These skills are highlighted here be-

Table 2: Tasks that constitute IT security management and the tools used for these tasks.

Task	Type of tool: Examples	Example of using a tool to perform the task
Receive and process notifications	E-mail: Pine, Outlook, Mutt	To receive an e-mail from myNetWatchman reporting a worm in one of the organization's machines
Monitor the network	Intrusion detection system (IDS): Snort, Argus	To set Snort to monitor network traffic and alert if attack's signatures are found
	Network sniffers: tcpdump, Ethereal	To capture and analyze the traffic using tcpdump and Ethereal.
Monitor systems	Monitoring tools: Cacti, SmokePing, MET Stat, Active Ports	To configure Cacti to monitor every host SNMP enabled
	Fingerprinting tools: Nmap	To scan ports of the network using NMAP
Prioritize activities	E-mail: Pine, Outlook, Mutt	To use e-mail filters to classify e-mails in different folders, detect anomalies by checking if quantity of e-mails received in one folder exceeds normal levels and start taking actions if this is the case.
Verify configuration of e-mail services	Anti-Spam tools: SpamAssassin	To ensure that spam filter does not filter wanted e-mails
Analyze logs and network traffic	Home made scripts: Perl, Shell	To use scripts written in Perl or Shell to analyze different log files and tcpdump and Ethereal to analyze packets that go through the network
Verify veracity of incident report	IDS: Argus	To use Argus to validate that malicious traffic is being generated from the internal network
Detect and report viruses in the systems	Antivirus software: Kasperski, AV, McAfee EPO	To use an Antivirus to confirm that the behavior of a machine was because of a malicious SW. To send reports with the status of the virus activity in the network
Detect and report vulnerabilities in the network	Vulnerability Scanners: Nessus, ISS	To detect vulnerabilities and generate reports of network's vulnerabilities with Nessus
Respond to events		To use Argus to detect anomalies in the network and sends a report to the network guys
Search information	Browser	To look for device's documentation on the web.
Patch or upgrade systems	Operating systems' feature: MS Windows update, other	To use Windows update to know about patches to the operative system and to install them
Correlate different sources of information	Home made scripts: Perl, Shell	To correlate different logs and come up with theories about the causes of security incidents
Use documentation	Browser	Scan documentation from different sources to decide which one is more useful
Execute re-configuration responses	Device management tools	To disable ports of the device by using its management console

cause they are related to the use of tools, and we think they are more strongly emphasized with ITSM versus IT systems administration. An example of a skill that all of our participants utilized but which we feel has little impact with respect to tools is *good communication*. *Design* skills at the level of planning new systems are also very important, but don't seem to impact tools, nor do we think they are more emphasized in ITSM.

4.4.1 Inferential Analysis

Various responsibilities, goals and tasks of our participants rely on circumstantial evidence and prior conclusions for their execution; that is, they require inferential analysis. Examples of such responsibilities are: (1) find and evaluate tools that enable the organization to see if its policies are being followed; and (2) make sure that a particular kind of incident never happens again. Examples of goals are: (1) keep a low profile (so as to not invite attacks); and (2) balance preserving what the organization has with planning what it is going to do. Examples of tasks are: (1) determine that a machine really is sending packets; (2) figure out what is crashing a system; (3) retroactively analyze traffic; (4) resolve an IP address to a name; (5) from network logs, find when an incident started, plus any other relevant information; (6) figure out what all the bits of an infection are; and (7) explain why a certain combination of technology works.

4.4.2 Pattern Recognition

> "*I can look and I can see anomalies, and I'm like, 'Oh yeah, this one over here, we gotta follow this trail and see where this goes'.*" (study participant)

Our participants commonly used pattern recognition for hypothesis formation during inferential analysis. To be-

gin with, our participants would recognize what a problem would involve. Examples are: (1) recognize that, to ascertain whether a machine is infected, a needle in a haystack of data from a network sniffer (like tcpdump) will have to be found, and therefore select the tool Ethereal to visualize the traffic and "burrow into the different levels"; (2) while refining a spam filter, from previous feedback from end users, know to focus on e-mail that scores 6 or 7, rather than 4 or 3; (3) be able to "*quickly parse about 500 pages*" of documentation. They could see significant similarities between situations: "*I didn't realize until I read the other bug report that what I had thought was irrelevant may very well be relevant.*" They could see significant differences between information: "*. . . make sure that that's consistent with what we think it should look like.*" They could see significance based on context: "*I would know based on what I read the other day that there is something wrong.*" Finally, based on the emerging pattern, they would suspect, think, and hypothesize: "*I don't think it's malicious . . . so we hypothesize that it was a malformed packet.*"

4.4.3 Bricolage

> "*. . . this is why I have a test machine here; sometimes you play a little bit with the technology or get it working, and after that you come up with the explanation of why it did work.*"
> (a wireless network security engineer)

Bricolage can be defined as "construction (as of a sculpture or a structure of ideas) achieved by using whatever comes to hand" [18].

Our participants use tools to perceive events and pursue analysis. Adaptation in a scenario that requires inferential analysis will involve learning by trial and error. The kinds of responsibilities that exhibit *response to events* show the clearest manifestations of adaptation. The IT security practitioner looks for specific patterns like *too many authentication errors* or *the same message over and over*, but he or she also looks for *unusual behavior*. Various factors will cause the practitioner to adjust the scripts that look for specific patterns. For example, an academic organization may base the significance of certain kinds of events on a particular threshold of traffic, and ignore events that don't threaten the IT service—scripts may have to be adapted to a change in policy or demographics. But in order to follow a new trail, the practitioner will also design new scripts.

To *play a little with the technology* also means using things in new ways. Being able to apply a model of security to a situation can mean being able to use things in new combinations, for example, proactively promoting the ability to audit access to SharePoint. Since SharePoint does not support auditing very effectively, one participant put a proxy server in front of the SharePoint server to keep detailed logs of who had access to the site.

Frequently, *play* will mean using the same tools to filter out different information for different reasons, for example, using *tcpdump* to look for users' passwords. Tool *tcpdump* is commonly used to sniff network traffic and find patterns related to TCP/IP headers. In this case, one of our participants needed to track, for syncronization purposes, when specific users connected to the server. To do so, he had to know the users' passwords for logging on to the server. So, he would use *tcpdump* to monitor when the particular user was connecting to his server, and get the password from the TCP/IP traffic.

To give a final example, normally, one uses antivirus software to identify whether viruses and the like are present. One participant used the tool in a different way. While investigating suspicious Internet Relay Chat (IRC) traffic, he noticed that a certain software program was downloaded. In order to find out more about that program, he also downloaded it, and ran it through his antivirus software.

Significantly, all of our participants would adapt their tools to obtain and compare alternate data sources to clarify and validate their hypotheses: "*That involves me using a variety of different methodologies to contact the VPN server and interpolate information that I'm getting from it.*" Concerning confirming how a Trojan entered a machine, in order to prevent this kind of thing happening again, one participant said, "*By talking to them* [the owner of the machine]*, one could figure out whether that happened.*" Redundancy of data sources protects against potential corruption due to system failure or tampering by intruders. Further, security-focused scripts that watch for security breeches may not report a security attack, while scripts that monitor responsiveness may pick it up.

4.5 Tools

How our participants felt about their tools is laid out here under the characteristics of organizational usability introduced by Elliot and Kling [8]. Elliot and Kling extend Nielsen's characteristics of usability [20] with characteristics of organizational usability. They add: (1) the ability to locate and gain access; (2) the ability to integrate into the preferred work practices of the individual and the group; (3) the ability to obtain training, consulting and help with problems of usage in combination with existing systems; and (4) hardware reliability. Finally, we relate the concerns that our participants expressed about various costs or overhead their work entailed.

4.5.1 Accessibility

All of our participants relied on e-mail and remote access to their systems. Their e-mail clients were sometimes the text-only Pine or Mutt so they could, with a keyboard tap, step through a large list of messages sent from automated scripts, tools, systems, and applications. Through remote login, they would use UNIX commands to investigate logs that were too vast to be sent by email. Some, as part of a group, maintained contact with each other by means of mobile devices such as Blackberry, or text or voice chats, e.g., iChat, Skype. Although none of our participants expressed likes or dislikes with respect to e-mail per se, nevertheless, it should be noted that e-mail is the first user interface in the workplace of our participants.

4.5.2 Integrability into Work Practices

Familiarity: Experts tend to be comfortable with textual interfaces because they are familiar with their problem domain and its necessary functionality. For example, when asked how he would teach an apprentice about configuring switches, one participant said:

> "*If you go to the CLI* [command line interface]*, and you have a black window, you don't even know what's there, right, so you don't know what*

> *to look for. I think it helps you in that regard, actually, to get your feet wet on your device... If you have a good understanding of what's happening here, I'd say that at that moment its totally irrelevant what kind of tool you use to change that.* "

But the more an organization distributes the handling of security management to non-experts, the more graphical metaphors need to be employed in the user interfaces of security tools: "*... as we try and distribute functionality out, we often, for the initial hardcore technical people, we will write tools that have very limited user interfaces, and the more we distribute those tools out, and the more we want administration to be handled by other than security experts, the tools have to be more user friendly with better user interfaces.*"

Our expert participants used both graphical and textual interfaces. They appreciated graphical interfaces that logically reflect the structure of the problem domain, such as the parts of a configuration file, and were not impressed by multiple ways of getting one thing. When dealing with vast amounts of data, the visualization of information afforded by colour coding was also appreciated: "*Ethereal colors things, which is kind of useful, so it shows the SYN and RESET [packets] in one colour, and then the Push commands in another colour—so it is obvious—there is content in there—it happens to be blue.*" With a graphical interface, the expert can easily look around when he or she does not know yet what is to be changed, and the novice can explore a high-level view. This advantage is also a disadvantage; the expert has to click through the structure to get at things, whereas a textual interface allows one direct access to any functionality. The important point is the play of tools, depending on the nature of the task. How typical IT security tasks weight the choices and transitions between tools, what has to be remembered or taken care of when transitioning, and so on, is not well understood. We feel this is an interesting problem for HCI researchers.

One tool should not try to be all things. "*We're all comfortable with using multiple tools. I don't want to use a hundred; I don't want to use just one; I want to use a handful that I know really well.*"

Since a tool should not try to be all things, its designers should understand how it is used in the environment. In the words of one of our participants, preferred security tools "*fit into the environment and not just the security landscape, but they need to be able to fit in with our other tools ... [and] be managed with our normal management processes.*" When shopping for security tools, an organization will look *first* at ones that are familiar.

Tailorability: Many of our participants were more comfortable working with a command line interface (CLI) than with graphical interfaces. One likely reason is that the CLIs inherently provide more opportunity for tailoring actions to [17]. One participant expressed that extracting from the results produced by Bourne-again shell (bash) scripts "*gives me no end of capabilities,*" while "*if you have a tool provided by a manufacturer, I would say that you would have only some pieces there, you would not have everything, what I might find interesting from the reporting point of view might be totally irrelevant to the security officer.*"

Flexible Reporting: "*Flexible reporting is something that a lot of tools lack. That is something we definitely need. We often turn our tools around and look for attacks that are leaving* [our organization] *and are going outside, and often vender tools just get confused and can't deal with that fact.*" Automated reports should be adjustable to the requirements at hand; they should not overwhelm the reader with unwanted details, nor should they be too vague.

Reports that indicate a problem should also provide a means to the solution. One participant praised the vulnerability scanner Nessus [19] for its meaningful and readable reports. Various export options are available: HTML, PDF, spreadsheet. The security practitioner can use the report to easily see which items are high priority (marked in red), and then reuse the report as an instruction list by handing it to a systems administrator to "fix the red items." The reader can select, by means of hyperlink, the level of detail commensurate with the task, such as what is the nature of the vulnerability, which reconfigurations are needed to eliminate it or where to obtain a required patch.

4.5.3 Social-Organizational Expertise

> "*Yeah I remember the last time I had to chase this stupid thing. I had already figured it out once but forgot because there is too much information ... Right now a lot of it's up here in my head, and due to lack of time to write it down.*"

Elliot and Kling define social-organizational expertise as "The extent to which people can obtain training and consulting... and can find help with problems in usage" [8]. Our participants relied heavily on documentation. Obtaining documentation can be painful. It can involve remembering numerous URLs with associated passwords, navigating confusing web sites, and collecting not only continuously updated material, but also associated white papers. Our participants also kept technical notes, such as what network nodes certain access points occupy. All this information would live in several places, because of the participant's mobility, and also in case some parts of the infrastructure were unavailable.

Our participants would also actively forget: "*The syntax throughout Open SSL is sufficiently complicated that I can't actually remember it... If I can write a script to do something I will... I can script them and forget how I did it...*" Nevertheless, these scripts can be recalled: "*I can look and see what is running out of cron [table] on this machine, sort of vaguely sense all naming conventions.*"

Sufficiently complete records will sometimes pay off: "*I didn't realize until I read the other bug report that what I had thought was irrelevant may very well be relevant.*"

4.5.4 Reliability

Not only were our participants concerned with hardware reliability, they also expressed their dislike of software that increased the risk of them overlooking critical information, which is a kind of error. Therefore, we generalize Nielsen's Software Reliability (called *Errors* by Nielsen—"System should have low error rates with few user errors, few catastrophic errors, and easy recovery from common errors," as cited in [8]) and Elliot and Kling's Hardware Reliability, with just Reliability. Examples of increasing the risk of overlooking critical information are: (1) an intrusion detection tool that drops

packets when it is overloaded, without notifying the user; (2) a graphical user interface (GUI) that writes configuration files that sometimes don't take effect; (3) a GUI that writes unnecessary, noisy markup into a configuration file, thereby increasing the risk of not noticing a syntax error; and (4) a java client for writing configuration files that can cause major problems due to inconsistencies between its version and the server's. With a plain text editor like *vi*, the user can be confident that *what you see is what you get.*

4.5.5 Overhead

Our participants expressed a need to be relieved of various kinds of overhead that come with their work. For example, one participant expressed the need to be relieved of having to investigate every alarm. To investigate is costly, and to not investigate entails the cost of assessing the risk of those alarms that are not investigated. To illustrate:

> "*It takes a lot of time to investigate all the different anomalies... it has to be good data, because I've seen guys that will alter the firewall rules to block it. Now the question is, if it was a false positive, you just caused a denial of service on yourself... I would like a tool that could watch trends over time... what's normal patterns for our network, what's not normal patterns... You might have different traffic happening on different cycles that happen throughout the year, so you have to be aware of it, otherwise it all falls into a huge bulk of web traffic coming in; it's not normal for the average throughout the year...*"

Another well-known overhead is the error-prone process of creating and shutting down accounts, resetting passwords, revoking permissions, removing rules, and re-booting computers, in order to grant or revoke access privileges in the presence of the rapid turnover of workers. We found other kinds of overhead: one participant would normally receive about 250 e-mail reports from machines over a weekend: "*I mean we watch everything that we can but a lot of it is not that exciting. I wish my e-mail was more useful. All these log watches and all this—spamming myself basically—to the point where I'm not paying close enough attention and looking for missing anomalies. But if* sendmail *gets messed up on one of my hosts and is not sending me e-mail, I won't notice until I actually go hard looking.*"

There is the overhead of an organization's legacy: one participant wished for something that could keep track of installed software packages and correlate them against a vulnerability database; although something like this exists, the one program our participant tried could not cope with the old packages still in use in the organization, nor did the program represent the older viruses. There is also the overhead of an organization's complexity: "*There's this thing—trying to figure out who a machine belongs to, which is sometimes difficult.*" Even the raw data that our participants regularly deal with comes with an overhead: "*The log information that* tcpdump *produces is quite noisy; if you don't look into detail you don't know.*" These accounts of overhead represent opportunities for improvement.

5. DISCUSSION

5.1 Recruitment Issues

Although it is impossible to verify, we believe graduated recruitment helped us to build up trust between the researchers and participants and the appreciation of the research objectives among the participants. After answering the questionnaire, our participants saw the interview as an opportunity for input to the community and not so much as an interruption of a workday.

The main advantage of this recruitment strategy was the improved success rate that we gained by approaching the organization through its management and asking permission to engage employees. Unfortunately, this approach also has two serious drawbacks.

One drawback of gaining access to employees through their supervisors is the potential perception of coercion to participate. Employees could feel that participating is implicitly a requirement of their job, or might be used as a review of their performance. Accordingly, they could feel uncomfortable providing candid criticism of their employer. We mitigated this risk by ensuring that participants understood that all information was confidential and would not be released to employers, including information about whether they participated or not. We took extra precautions to conceal their identity and protect the confidentiality of the questionnaire and interview data.

Another drawback is the potential compromise of the participants' reports. To illustrate, if actual procedures differ from management policy, the involvement of managers in setting up the interview might compromise the participants' reports of what they actually do. Our semi-structured interview mitigated this possibility by discussing some of the organizational structures and the roles and responsibilities of the participant's colleagues. We found that our participants were candid about the persons in their respective organizations whose roles were relevant to IT security management.

5.2 Research Result Considerations

The HOT Admin project focuses on the development of tools to support ITSM in real-world situations. We began our investigation by asking security practitioners to tell us how their organizations manage security administration, what tools they use, and how those tools meet the needs of their particular organization. Grounded Theory and case study methods were chosen to enable us to evolve theory from data with minimal preconceptions, laying the groundwork for more focused user studies and tool design phases of the project. Keeping in mind that these findings are early, and therefore suggestive rather than definitive, we here summarize the findings that we think are important.

Our participants' reports of the shortcomings of existing tools focused on need for *tailorability.* IT security practitioners typically saw themselves as monitoring the pulse of their organizations, forming and investigating hypotheses, conducting tests, and diagnosing the results. The level of risk, nature of threats, and cost of false alarms or missed threats can vary greatly between organizations and over time, and a skilled administrator adjusts his or her actions accordingly. Our results suggest that the impact on tool use of this diversity of situations has been underestimated by the tool developers. They should better enable practitioners to tailor their tools for the situation of use; that is, for the worker

to "finish the design" [22, 9].

An unexpected finding in our study was the degree to which ITSM is distributed across the organization. We feel this distribution has an impact on tools as well, in the need for tools to support varying roles for individuals with different levels and scopes of expertise, and the need to support collaboration among them. This may lead to entirely new classes of tools, e.g. supervisory control tools for coordination of security management.

When we broadened our investigation to include responsibilities that have security implications, we found three different kinds—respond to events, design, and maintain systems—which in turn were largely performed by different roles within the organization. Altogether, the security of systems was modeled in most of these organizations not as a focus that required specialization, but merely as an aspect of the supporting most IT systems. We have therefore clarified the focus of this study to deal with the practice of "IT security management" and not merely with "IT security administration."

This view of security management as a *cross-cutting concern* was similarly reflected in the study participants' tools, most of which were a mix of system-specific configuration management and monitoring facilities and tailorable, generic text and information management tools (e.g., grep, shell scripts and e-mail).

We invite IT security tool developers and researchers to consider that the tools in question survive in an arena of *bricolage*. Our participants used tools that come readily to hand in different situations that arise out of the complexity of both the technology and the environment. Many of their tools are generic, like command line interfaces or interpreted scripts, which inherently offer versatility. Our participants had limited toolkits of tools that they knew well. The handful of trusted tools were, together, versatile. That is, they could be used together in creative ways to accomplish different tasks in various scenarios. This suggests a couple of design principles: (1) IT security tools should seamlessly incorporate interpreted scripts in order to extend their functionality; and (2) these tools should be customizable, for example, enabling the construction of different types of reports depending on the recipient.

We imagine that much can be done to help users articulate many of the patterns that they recognize in order for their tools to also recognize the patterns and thereby be in a better position to help. To illustrate, imagine that a security professional discovers a suspicious IrChat command while sifting through logs. The security professional highlights the term, and selects "Pattern" from a menu. The system then determines that the term is from IrChat, finds other IrChat terms that have been implicated in security incidents, and locates all instances of suspicious IrChat terms in all the archived logs. Another example: traffic on the practitioner's network follows periodic business trends; the practitioner configures a security tool to project business trends based on business indicators (e.g., previous years, sales pipe line). The tool can thereafter better estimate unusual traffic. Hopefully further research will reveal a taxonomy of typical kinds of pattern that security tools could profitably support. Likely much can be done to help practitioners compare logs (and other kinds of information) from different sources. This means that the tools not only should recognize known patterns in the data, but also combine and correlate them to find complex trends in the data. Tools with these abilities would be better able to advise security practitioners about when, and when not, to investigate.

Tool developers want to learn more about how their tools are used in coordination with other tools, and with existing ITSM practices. A goal would be to support versatility without succumbing to feature-creep. In order to accomplish this goal, many more examples of tools being used in versatile ways are needed. By *testing the use of suites of tools* in real-world task scenarios, one may uncover usability problems that result from differences in assumptions of the various tools, and difficulty in transferring information between them in order to coordinate a particular task.

6. CONCLUSION

We gained a better understanding of how practitioners of information technology security use their tools. We used ethnographic methods to acquire and analyze the data. Semi-structured interviews comprised our primary way to collect data, and we used both pre-defined themes and grounded theory to analyze it. Recruiting participants was difficult but their participation was active and fruitful.

Our findings can be summarized along the HOT dimensions as follows. Human: the high-level skills of inference, pattern matching, and bricolage distinguish ITSM. Organizational: ITSM is distributed among several professionals—or even groups of them with dedicated "coordinators"—scattered throughout organizational units. Technological: ITSM tools are used in coordination with each other, support for flexible reporting and tailorability are necessary.

We plan to validate and refine our findings further by involving more participants from diverse organizations, and to deepen our understanding by performing workplace shadowing to capture details of interaction and tool use that users find hard to articulate or to remember [16, 12, 6]. This information will be used in the design and testing of new interfaces for IT security management.

7. ACKNOWLEDGMENTS

The HOT Admin project is funded by the Canadian NSERC Strategic Partnership Program, grant STPGP 322192-05. The authors are grateful to all those IT professionals who donated their time and energy to participate in the field study reported in this paper. Rob Ross helped testing data collection instruments. Mary Ellen Zurko provided feedback on the initial design of the study.

8. REFERENCES

[1] Argus intrusion detection and prevention. http://www.qosient.com/argus/, February 2007.

[2] R. Barrett, E. Haber, E. Kandogan, P. Maglio, M. Prabaker, and L. Takayama. Field studies of computer system administrators: Analysis of system management tools and practices. In *Proceedings of the Conference on Computer Supported Collaborative Work*, 2004.

[3] A. Bartels, B. J. Holmes, and H. Lo. Global IT spending and investment forecast, 2006 to 2007. Forrester Research, 2006.

[4] F. J. Björck. *Discovering Information Security Management*. Doctoral thesis, Stockholm University, Royal Institute of Technology, 2005.

[5] S. Bodker. Human activity and human-computer interaction. In S. Bodker, editor, *Through the Interface: A Human Activity Approach to User Interface Design*, pages 18–56. Lawrence Erlbaum Associates, Publishers, Hillsdale, NJ, 1991.

[6] H. H. Clark. *Using Language*. Cambridge University Press, Cambridge, England, 1996.

[7] H. H. Clark and M. F. Schober. Asking questions and influencing answers. In J. M. Tanur, editor, *Questions about questions: Inquiries into the cognitive bases of surveys*. Russell Sage, New York, NY, 1992.

[8] M. Elliott and R. Kling. Organizational usability of digital libraries: Case study of legal research in civil and criminal courts. *American Society for Information Science*, 4(11):1023–1035, 1997.

[9] G. Fischer and E. Scharff. Meta-design: design for designers. In *Proceedings of the Conference on Designing Interactive Systems (DIS)*, pages 396–405, New York, NY, USA, 2000. ACM Press.

[10] B. Glaser and A. L. Strauss. *The Discovery of Grounded Theory, Strategies for Qualitative Research*. Aldine Publishing Company, Chicago, Illinois, 1967.

[11] U. Holmstrom. User-centered design of secure software. In *the 17th Symposium on Human Factors in Telecommunications*, Denmark, 1999.

[12] E. Hutchins. *Cognition in the Wild*. MIT Press, Cambridge, MA, 1995.

[13] Internet relay chat (irc) help archive. http://www.irchelp.org/, February 2007.

[14] E. Kandogan and E. M. Haber. Security administration tools and practices. In L. F. Cranor and S. Garfinkel, editors, *Security and Usability: Designing Secure Systems that People Can Use*, chapter 18, pages 357–378. O'Reilly Media, Inc., Sebastapol, 2005.

[15] K. Kark, C. McClean, L. Koetzle, J. Penn, and S. Bernhardt. 2007 security budgets increase: The transition to information risk management begins. Forrester Research, 2007.

[16] 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.

[17] T. Malone, K. Lai, and K. Grant. Two design principles for collaboration technology: Examples of semiformal systems and radical tailorability. *Coordination Theory and Collaboration Technology*, pages 125–160, 2001.

[18] Merriam-Webster. Merriam-webster's collegiate dictionary, 1994.

[19] Nessus security scanner. http://www.nessus.org/, February 2007.

[20] J. Nielsen. *Usability Engineering*. Morgan Kaufmann, San Francisco, 1994.

[21] Idea works: Qualrus software. http://www.ideaworks.com/qualrus/index.html, February 2007.

[22] K. J. . Vicente. *Cognitive Work Analysis: Toward Safe, Productive, and Healthy Computer-Based Work*. Mahwah, NJ: Lawrence Erlbaum Associates, Publishers, 1999.

[23] A. Wool. A quantitative study of firewall configuration errors. *Computer*, 37(6):62– 67, 2004.

[24] R. Yin. *Case study research: Design and methods (2nd ed.)*. Sage Publishing, Beverly Hills, CA, 1994.

[25] M. Zurko, R. Simon, and T. Sanfilippo. A user-centered, modular authorization service built on an RBAC foundation. In *IEEE Symposium on Security and Privacy*, pages 57–71, Oakland, CA , USA, 1999.

[26] M. E. Zurko and R. T. Simon. User-centered security. In *New Security Paradigms Workshop*, pages 27–33, Lake Arrowhead, California, 1996. ACM Press.

APPENDIX

A. STORIES OF TOOL USE

A.1 Respond to Events

Events can be caused by entities that are external to the organization in question. For example, myNetWatchman may send an e-mail that will eventually be received by a particular security administrator. However, the IT security practitioner often constructs or refines the tools by which the events are perceived and determined to be significant enough to raise an alarm. An alarm may be a false alarm, thus alarms require analysis in order to verify that they are true. Finally, a true alarm requires a response.

In this story, the practitioner is notified of events through e-mail. The e-mail notification may come from an external service through an intermediary, such as a host organization. E-mail also comes from scripts. The scripts are handcrafted based on experience and knowledge. They are run by *cron*—the clock daemon in UNIX that executes commands at specified days and times—and are used to interpret system logs. E-mail also comes from sophisticated tools like Snort and Argus [1] that monitor live traffic. The e-mail subject line helps the practitioner to quickly assess whether something needs immediate action, e.g., "severity 1", and prioritize accordingly. Folders that the e-mail might be filtered into also inform. For example, folders might be devoted to firewalls and routers; in this case, the subject line is a statement about the folder. The number of messages in a folder may indicate significance–500 versus 5. The practitioner may have to compare information from different folders. For example, the practitioner may look at "the router in front of the firewall."

Failure of a routine message to show up can indicate significance. In this case, the practitioner has to notice that something is missing. One interviewee used the text-based Mutt e-mail client in order to rapidly step through the subject lines. The reports come in at random times by design so that all of the machines don't hit the update mirror at exactly the same time. The practitioner compares the absence with another data source, like a SmokePing indication of packet loss, in order to verify the alarm.

Apart from e-mail, the practitioner looks at logs from firewalls and servers, which are too big to be e-mailed. Argus and Snort, which monitor the live traffic, also provide logs. Argus only "looks" at the packet headers, and in this way provides a level of respect for privacy that may be required by the organization. System failure or tampering by intruders can damage server logs. Thus, some security practitioners use Argus as the primary data source, to be verified against the other sources. The practitioner will skillfully

use UNIX commands, scripts and likely an input IP address to find a timestamp–when the significant event started. The timestamp, known patterns of packets, and usual system behaviour are inputs to help the practitioner recognize patterns such as: unusual behaviour, funny messages, too many unreadable messages, too many authentication errors, the same message over and over, evidence of suspicious entry (many attempts on successive ports from the same IP address, and then no further attempts), too many e-mails in one hour from one machine, and suspicious logins (from home and office within one minute of each other). Success here leads to better design and implementation of scripts and refinement of Snort and Argus rules.

To illustrate: At home early one morning a security administrator notices, in his Pine e-mail client, a forwarded message that originated with myNetWatchman. It says that one of his machines is sending out suspicious packets. He puts aside his other duties and tries to determine if the warning is indeed true. He could do this from home by remote login. In this scenario, he would use *tcpdump* to create a secondary binary file that is about the suspect machine only. He would run the command *strings* on this file to retrieve any human-readable information. If he decides to go to the office, he could open the secondary binary file in the graphical interface of Ethereal, and dig down through the layers of protocols. Ethereal would colour the human-readable bits blue. He recognizes that some of the readable text looks like Internet Relay Chat [13], which he knows is common with hackers and rare with his organization. By going back through the logs of the offending machine (the logs are big, which limits how far back he can go), he gets an idea of when the suspicious behaviour started. He looks up where the machine is and who is responsible for it, and validates that this information is not stale. He then goes and speaks with that person to figure out what human behavior resulted in the machine being infected, in order to help prevent it from happening again.

The practitioner will respond to the alarm. For example, a security administrator may turn off the ports that are denying service, go and look for the offending Trojan or whatever and remove it, make sure that such an incident doesn't happen again, and perhaps initiate disciplinary action.

Sometimes other people need to be notified. For example, the security administrator may let a Windows administrator know about firewall rules that exclude an infected machine, and how to remove them once the machine is cleared.

A.2 Design Solutions

In this story, the designer of a wireless network has several goals: (1) increase the access points and redundancy based on expected growth of demand; (2) keep the same access design as the main network; (3) make it secure while allowing people with older computers to connect to it, which entails the sub-goals of upgrading the access mechanism and the encryption standard while accommodating older computers that cannot connect using the latest technology. The network designer sets up two laptops. One has a Web-based graphical interface that he uses to quickly survey a configuration file and poke down into parts of it, which is useful when he is not sure of what he is going to change. The interface reflects the structure of the configuration file, but from what he considers to be a 30,000-foot view. The other laptop has the test access point. On the other laptop, he uses the command line interface (CLI) to make changes to the configuration file and to the switch. With the CLI he can, in a sense, speak directly to any part of the switch or configuration, as opposed to having to point and click his way through the structure. With the CLI, he can more efficiently utilize his deep understanding of the structure. Manipulation of a configuration file by means of the CLI is clean, whereas the graphical interface will write noisily. Sometimes he will use *vi* to edit the configuration file. With *vi* he can copy something and paste it many times, then change little bits in the pasted parts, and thereby make big changes to the network. The alternative of having to fill out *all* the values in many copies of a structured object, especially if many of the values are the same or the structure is complicated, can be tedious and thereby error-prone. Copying and pasting from the graphical interface picks up unwanted HTML fragments.

The network designer keeps a notebook or file of the many URLs and their passwords to obtain support documentation and related literature like white papers. The Web sites are often confusing. A complete documentation often comprises a combination of files. The technology changes constantly. He prefers a printed document because it is easier to read and can be marked up, though he will read recently changed sections in electronic form. He keeps copies of his notes in multiple locations (disk, home folder in the Windows domain, laptop). He does so because he will often have to grab his laptop with his notes about IP addresses of key network devices, which access point is which network node, and so on, and take it to the machine room, since the documentation often helps with points that he doesn't anticipate. If he has broken a switch, he cannot rely on the network for getting the needed information.

A.3 Maintain Systems

In this story, a security analyst uses the Nessus vulnerability scanner [19] to probe the entire address space. Nessus gives the security analyst the level of detail consistent with his needs. For example, it will probe a port to find out what service is running on it, and discover the patch level of that service. If there is a problem, it will report the port, what the service is, what the vulnerability is, and provide a link to get the upgrade. The layout and styling of the information makes it easy to read. In contrast, another software would merely report that a port has a problem, and that's all. Yet another software would produce many pages of difficult-to-read, specialized technical information. The security analyst is able to hand the Nessus report directly to a system administrator as an instruction sheet about what to do.