Developing Expertise for Network Intrusion Detection

John R. Goodall, Wayne G. Lutters, and Anita Komlodi

Published in *Information Technology & People* 22 (2), 92-108

© Emerald Group Publishing Limited

Article URL: http://www.emeraldinsight.com/10.1108/09593840910962186

Category: Research paper

Keywords: intrusion detection, network security, work practice, socio-technical, situated expertise, field study.

Purpose: To provide a foundational understanding of the socio-technical system that is computer network intrusion detection, including the nature of the knowledge work, situated expertise, and processes of learning as supported by information technology.

Design/methodology/approach: We conducted a field study to explore the work of computer network intrusion detection using multiple data collection methods, including semi-structured interviews, examination of security tools and resources, analysis of information security mailing list posts, and attendance at several domain-specific user group meetings.

Findings: The work practice of intrusion detection analysts involves both domain expertise of networking and security and a high degree of situated expertise and problem solving activities that are not predefined and evolve with the dynamically changing context of the analyst's environment. This paper highlights the learning process needed to acquire these two types of knowledge, contrasting this work practice with that of computer systems administrators.

Research limitations/implications: This research establishes a baseline for future research into the domain and practice of intrusion detection, and, more broadly, information security.

Practical implications: The results presented in this paper provide a critical examination of current security practices that will be useful to developers of intrusion detection support tools, information security training programs, information security management, and the practitioners themselves.

Originality/value: There has been no research examining the work or expertise development processes specific to the increasingly important information security practice of intrusion detection. The research prepares a foundation for future research into understanding this highly complex, dynamic work.

Citation: Goodall, John R., Wayne G. Lutters, and Anita Komlodi. "Developing Expertise for Network Intrusion Detection." *Information Technology & People* 22(2), Emerald Group Publishing Limited, 2009, 92-108.

Contact: John Goodall, johng@securedecisions.avi.com

INTRODUCTION

Insights into the complex dynamics of socio-technical systems often arise from a thorough examination of particular breakdowns or mismatches between social actors and technical mediation. Informed by this approach, the research reported in this paper furthers our understanding of the development and practice of situated expertise in network security through an exploration of the work of security analysts, those charged with protecting computer networks from attacks. Specifically, we examine intrusion detection (ID) analysts, who perform the critical, knowledge-intensive security service of monitoring and responding to events to determine their accuracy and severity. The central breakdown that motivates our analysis is an industry-wide attempt to automate all network security activities, without regard for the critical role of human expertise, from network monitoring through automated response via intrusion prevention systems (IPS). The friction between this ideal of complete automation and the actual daily work worlds of security analysts reveals much about the delicate balance required between reliance on specialized human expertise and intelligent computational tools. Along the way this illustrative case study examines the predominant role of situated expertise in work practice, and ultimately calls for new "third path" solutions – augmenting human capacity through better support of external cognition via information visualization and supplemental support systems that continuously collect, store, and present contextual information about the state of the network environment.

The problem of network security is a practical and pressing concern; a report calculated that the cost to organizations for each security breach is nearly $14 million per incident (Hall, 2007). Perhaps more troubling than the financial cost to organizations is a new form of cyber warfare; the nation of Estonia sustained a prolonged electronic attack on government web sites, requiring a digital quarantine, cutting off all access to the outside world (Kirk, 2007).

Intrusion detection is the primarily reactive security work of monitoring network activity for signs of malicious or abnormal behavior. ID has become an increasingly critical activity within the overall security policies and practices of most organizations. To help this process scale in order to secure ever-expanding networks, human analysts rely on intrusion detection systems (IDSs). In the words of one participant in this study, these systems automatically detect "intrusions and behavioral misuse" (Participant 8; quotes from participants are labeled P# in the text) by matching patterns of known attacks against ongoing network activity to produce security alerts detailing those events. Unlike IPSs, they do not automatically take action, instead they provide analysts with "awareness and control" (P1) over the operating environment and give "some indication if you are vulnerable" (P8). However, because of false positives and the potential for self-damaging responses to inaccurate alerts, all of our participants agreed that fully automated IDSs are never a completely effective solution, and despite attempts at automated solutions, there is no substitute for the intuition that human analysts bring to bear on the process. A survey found that while 91% of organizations sampled used an IDS, only 50% of the respondents described their IDS as being effective in comparison to other security-related technologies (E-Crime Watch survey, 2005). The sheer number of IDS alerts can be overwhelming: "IDSs can easily trigger thousands of alarms per day, up to 99% of which are false positives" (Julisch, 2003). Instead IDSs require vigilant oversight by expert human analysts. As one ID mailing list message noted:

> An IDS, i.e. a monitoring system, needs to display only events of interest, that a human operator can and should resolve.

These human operators face challenges that differentiate their practice from the work of systems, database, and network administrators, roles that security analysts are (or were) often also engaged in. This paper is about those human experts and the development of their critical expertise.

In order to understand the work practice of ID analysts, we conducted a field study to determine how they develop and leverage their expertise in performing that work. We hoped to identify the types of expertise required in ID work and the ways in which it was developed. This research was part of a larger study to improve the design of tools that support ID work.

MOTIVATION

The use of field study methods to inform the design of technological support tools has been successfully employed in information-rich and high-reliability domains; that is, workplaces where information needs are demanding and the consequences for error are severe. Some examples include air traffic control (e.g., Bentley et al., 1992; Hughes et al., 1992), subway transit coordination centers (e.g., Heath and Luff, 1992), and aerospace service engineering (e.g., Lutters and Ackerman, 2007). This field study has yielded a foundational understanding of the work practice in high-reliability organizations and has led to conceptual designs for new socio-technical systems to support those tasks. These diverse settings share much in common with ID work, including processes for monitoring of system activities and responding to abnormal events.

The original intent of this study was to perform a deep requirements-gathering exercise to develop more task-relevant support tools for ID work. It rapidly grew into an extended examination of the nature of that work, in the tradition of studies like Harper (1998). It also wrestled with the task of transforming these findings from the field into concrete guidelines, functional prototypes, and eventual open-source software[1]. As per Hughes, et al. (1992) and Martin and Sommerville (2004), this is not a trivial task. Our process for doing this and the initial results are partially reported in (Goodall et al., 2006).

RELATED LITERATURE

As few research papers examine the work practice specific to ID analysts, the following studies of system administrators and information security specialists helped frame our analysis. In this review we compare and contrast the nature of expertise and learning in these two cognitively demanding disciplines while highlighting the specific challenges of security work.

System Administration Work

Sandusky (1997) explained computer network management work as a combination of both real-time supervisory work and design work. Real-time supervisory work is described as event-driven and time-constrained, while design work is typically self-paced and reflective. This division of network management work also applies more broadly to other categories of system administration work. Similarly, Halprin (1998) defined two types of system administration work, proactive (similar in concept to design work) and reactive (similar to real-time supervisory work), the latter of which is described as being repetitive and unchallenging after the initial learning curve. These distinctions are similar to the more general observation by Barley (1996) that the knowledge work of all technicians can be described through two key arcs of "transformation" taking feedback from the system and abstracting it (e.g., to compare an IDS alert with network traffic data to reason if it is an attack or false positive) and "caretaking" using that knowledge to modify the system (e.g., tweaking a signature).

Anderson (2002) performed a meta-analysis based on a literature review of twelve years of the Usenix system administration conference. Automation emerged as particularly important in common administration tasks. The automation-related tasks that infuse system administration work are another key difference between that work and the work of ID. Intrusion detection analysts use automation tools, but a human is always a central part of the process; in system administration, scalability dictates the need for as much hands-off automation as possible.

Anderson's meta-analysis also raised the issue of training, which is particularly relevant when comparing expertise in system administration work to ID work:

> Training tasks may be partially transferable out of the organization and into the schools. Users could be trained in the tools they will be using, and administrators could be trained in system administration. Earlier education would mean people would only have to learn the specifics of a site rather than the general knowledge. (Anderson, 2002)

[1] http://tnv.sourceforge.net/

Learning the tools and being able to transfer that knowledge into other environments was a central goal of system administration work. The transference of skills between a simulated training environment and a real-world environment, or between different field settings, is more problematic in ID work, a challenge that will be discussed later in this paper. In system administration work, knowing the tools equals a large portion of the work. Individual environments need to be configured, monitored, and maintained, but the tools and practices remain constant across environments.

Barrett, et al. (2004) described a field study of system administrators, defined broadly as those who architect, configure, administer, and maintain computer systems, focusing specifically on database and web administrators. The challenge of systems administration in this work is defined not just by its demanding cognitive activities, but by the high social demands of coordinating tasks and collaborating with others. This research highlighted the reliance of systems administrators on human communication to accomplish this work: 90% of the troubleshooting time for one case was spent on human-human interaction, while only 9% was spent on human-computer interaction. This importance of human-human interaction is also central to the findings of our own research on ID work.

Information Security Work

Yurcik, Barlow and Rosendale's (2003) workshop paper results were derived from their personal experiences performing security at the National Center for Supercomputing Applications (NCSA). This work framed the central challenge in information security – the asymmetry between attackers and defenders, which gives the former the advantage in an escalating battle. For example, defenders must continually identify and repair every vulnerability, while an attacker need only find a single unpatched vulnerability to exploit. This asymmetry hints at the level of expertise demanded from ID analysts defending their network. While many attackers are unsophisticated and use automated tools to locate and exploit vulnerabilities, the defender must not only be vigilant and quick to repair, but must also know what it is that needs to be patched. This suggests that analysts must be experts not just in information security, but also in knowing their environment, which is discussed in detail later in this paper. This research, like our own, points out the irreplaceable human element in providing network security.

Researchers at IBM investigated network operators' problem-solving tasks in the 24/7 monitoring of multiple customers' networks in a security operations center, or SOC (Stolze et al., 2003a; Stolze et al., 2003b). The goal in this type of managed security environment is to identify and report to the customer any anomalous network activity from data collected by sensors on the customer's network. Their operators related three core challenges: problem solving, learning, and cooperation. The learning challenges are pertinent to our research, and highlight the need for ongoing learning in the highly dynamic environment. Although work in a managed security SOC is different from the work we studied (which was done locally by analysts defending their own network), each of these distinct areas can learn from the practices of the other. In a SOC, analysts do security-related work all of the time, whereas the analysts we studied ranged in the amount of security-related tasks they performed daily

The most in-depth research to date examining security-related work was a cognitive task analysis of security analysts within the Department of Defense (D'Amico et al., 2005). In that analysis, researchers were looking at the entire process of Information Assurance (outside of government, called Information Security), whereas our research examines the tasks specific to ID. In addition to the more focused approach we take here, we included a more diverse sampling of analysts and examined not just the individual cognitive tasks, but also the social construction and use of knowledge and expertise within the community of ID practitioners. Their work highlighted several core challenges analysts face, including the need to develop site-specific mental models of what is normal, which we expand on and discuss at length in this paper. This is problematic because of the steep learning curve, the dynamic nature of computer networks, and the difficulty in articulating these mental models to others, which slows the learning rates of novice analysts.

Our own prior work (Goodall et al., 2004) began to examine the nature of collaboration and the role of expertise in ID work, particularly, the importance of learning what is normal within a given environment. The research presented here builds and expands upon those preliminary findings.

METHODOLOGY

To unpack both the mundane and exceptional processes of analysts' ID activities, we conducted a field study triangulating among multiple data collection methods, including semi-structured interviews, analysis of security and ID mailing list posts, examination of security tools and resources, and attendance at ID user group meetings. Our unit of analysis was the individual analyst operating within his routine work context. While we uncovered organizational themes and a surprising degree of collaboration, these were outside the scope of the study design and were secondary themes to those we identified for the individual and the computational tool suite that supports his work.

Our participants included a diverse cross-section of ID experts, some working as stand-alone analysts, some as members of teams. Although organizational mission, primary role, and ID expertise varied across the participants, there was a consistent pattern to their work. All participants possessed a working knowledge of at least one IDS, with a common reference point of Snort, an open-source network IDS (Roesch, 1999). Recruiting participants was a significant challenge for this project; there are few ID experts in any given geographic region, their knowledge is proprietary, and access to their work area is restricted, all of which combined to prevent extended onsite observations.

Our primary data collection method was semi-structured interviews, following a prepared interview guide though allowing off-topic elaboration. Interviews were conducted *in situ* when possible, encouraging participants to demonstrate their interactions with their IDSs, support tools, and commonly used resources. We conducted twelve interviews, which have been transcribed, coded, and analyzed. Data analysis was informed by Grounded Theory approaches (Strauss and Corbin, 1998). However, as noted in Lings and Lundell (2005), when studying socio-technical phenomena, it can be problematic to enter the research with no background in the area and they suggest that a strong technical "pre-knowledge" is necessary to understand the issues being addressed and to avoid negatively affecting interaction with participants. We entered analysis with no preconceived expectations or hypotheses, but with a working knowledge of the domain being studied. This knowledge was essential in this research to be able to understand the technical component of the work and the jargon of the security world. Being able to "speak the language" of information security and ID was also crucial in gaining the trust of participants. The first author's domain knowledge was also derived from first hand experience as a systems administrator, experience that goes beyond a theoretical understanding of the domain. So while Grounded Theory guided data analysis, it was not used prescriptively.

In addition to the primary data collection method of interviews, we examined the tools used by the analysts to understand how those tools worked and put interviewee explanations into context. This led us to explore the various security-related resources, particularly on the web, that were common among the analysts. We content coded an ID practitioner mailing list for a period of one year, specifically looking for broad themes, such as expertise and understanding "normal" network behavior, that had been uncovered during the interviews. During the one-year period we recorded a total of 1178 messages, averaging about 98 message posts per month. We also analyzed posts from various other security-related mailing lists for shorter periods of time to provide additional context. The threaded conversations between members of the ID community provided a context for understanding this community and in many cases served to confirm our findings with a broader audience. We also attended two Snort user group meetings to gain an understanding of the face-to-face interactions among a community typically linked by electronic connections.

PEOPLE

Our participants had wide-ranging organizational security needs, primary job duties, and levels of ID experience. The participants' organizations and corresponding security practices ranged from the relatively open environments of university settings to the highly secure defense contractors, financial companies, and managed security service providers. The primary roles of the participants varied: most were network or systems administrators whose duties included ID, few were dedicated information security analysts, and two were IDS developers consulting for other IT departments in their organization.

Interacting with an IDS is just one part of a job that typically includes a myriad of other systems, network, or security related tasks: "most people aren't just [IDS] analysts" (P3). This is particularly true in smaller

companies where a dedicated security person is not likely to be cost effective because the organization does not believe its risk level warrants it. For example:

> That's how I describe myself now, more of a systems administrator who does security work, because my company isn't big enough to have a security person full time. Even with a hundred employees, I think I could spend my entire day, every day of the year doing security stuff, making things better than they are now, but from the company's point of view, they don't need that. (P9)

This analyst works primarily as the systems administrator for several dozen machines, but is also responsible for all of the organization's information security needs, including ID responsibilities. This is typical of analysts being pulled in multiple directions by their organizational responsibilities. Security is often sacrificed for more visible organizational needs. Security in general, but ID in particular, is rarely noticed by management unless an attack is successful or a system is compromised – that is, when security fails and the notification is too late. As such, ID work is largely invisible (Star and Strauss, 1999). The analysts had the sense that their supervisors do not recognize the majority of their work, especially the mundane, routine work, until there is a crisis. As such, it is difficult to measure the productivity of an ID analyst, as much of the work involves the unseen ongoing processes of learning, which will be discussed below.

WORK PRACTICE

As noted in the literature review, although many security analysts were formerly systems or network administrators, the practices of those primary roles and their ID work were markedly different. While much of the knowledge required, such as understanding system and networking protocols, and even some of the skill set, such as a proficiency with scripting languages, overlap, the work practice itself is fundamentally different. Their administration work consisted of architecting and designing new solutions (proactive), and monitoring and supervising existing implementations (reactive), to use Halprin's distinction. By contrast, intrusion detection work is, by definition, almost entirely reactive. The work involves reacting to some potentially damaging or malicious event, such as an IDS alert. The reactive nature of ID places it squarely in the Sandusky's "real-time supervisory" category. However, ID analysts do perform tasks related to design, particularly the configuring and reconfiguring (referred to as tuning) an IDS. In this sense, ID work is similar to system administration work: both have an event-driven and a design element. The key difference is that in the ID work described in this paper, even the "design" tasks are event-driven and reactive in nature; an IDS alert, for example, may trigger a system reconfiguration – a design task – as part of the analyst's reaction to the event, but the task is unplanned, unscheduled, and reactive. Another key difference is that the reactive tasks associated with ID work, unlike system administration work, can be challenging, even to experts. While some ID work is repetitive, such as quickly diagnosing an alert as a false positive because the analyst has seen it before, the majority of analysts' effort and time doing ID is dedicated to solving novel problems. This event-driven, reactive work demanded participants' attention as it happened, whereas other forms of systems administration could often be scheduled or put off.

Intrusion detection involves more than reviewing IDS alerts and occasionally responding to critical events. It cannot be accomplished in isolation. It also requires monitoring and analyzing systems tangential to the IDS as well as keeping abreast of the latest security-related information. Although there are subtle differences in how security analysts perform ID, all of our participants followed a similar process, which we abstracted into four task phases: *monitoring*, *triage*, *analysis*, and *response*. Analysts often move fluidly from one to the other.

Monitoring includes the ongoing surveillance of both the analyst's environment, looking for indications of anomalous or malicious activity, and community resources, looking for vulnerabilities and new attacks. These mundane tasks constitute the majority of analysts' time. They are described in the following exchange:

> P5: First thing every morning, Tripwire, Snort, and ISS. Everyday, we go through web logs and look at stuff, if [anything is] out of whack.
> *Interviewer: What do you look for? web logs?*

> P5: Something that looks malicious that we know is not on our system, some kind of file. Generally you will see it, like cmd.exe, or something like that.
> *Interviewer: Can you setup an intrusion detection system [to look for this]?*
> P5: We still go through them in case, because the signatures aren't catch[ing] everything. Everyday, we look at Bugtraq, Full Disclosure, some of the other product lists that we subscribe to. I look for any vulnerability that come out. Any kind of attack.

This daily ritual involves examining the security and application log files generated overnight within the analyst's environment and the community resources that include security and product mailing lists to keep abreast of the latest attacks and vulnerabilities. The security logs alone are not enough, as some unknown attacks will be missed "because the signatures aren't catch[ing] everything". (Signatures are the rules of an IDS that define a pattern to compare network traffic to in order to identify and classify an attack.)

The continuous monitoring task is typically interrupted by an event, which then triggers each of the other tasks in turn. The term *event* here is used to permit a common framework between seemingly disparate terms. An event could be, most obviously, an alert generated by an IDS. It is also used to describe the announcement of a new vulnerability, a phone call from an end-user, an email describing a new attack method, or more abstract markers such as sluggish network speeds. Each of these triggers the ID processes described here. As such, a discussion of the work of ID focuses on the event-driven work – from the discovery of an initial trigger event through its triage, analysis, and response. We define triage as either the quick dismissal (as a false positive) or prioritization of events, such as IDS alerts, that analysts perform after discovering a new event during monitoring. Analysis, the most complex task and the crux of ID work, requires a thorough understanding of context. A key component of this context rests in the analyst's situated expertise, embodied in phrases such as "knowing what's normal traffic" (P12). Response most commonly results in intervention, feedback, and reporting. Response also introduces an opportunity for design, which, unlike in system administration work, is part of the reactive process. Accomplishing each of these tasks quickly and effectively requires a high level of expertise, the nature of which is the focus of the remainder of this paper.

EXPERTISE

Intrusion detection is complex, cognitively demanding work. The expertise required to accomplish ID comprises domain knowledge in ID, networking, and security and local knowledge grounded in an analyst's unique environment. Domain knowledge is explicit knowledge that analysts can describe and document, and forms the basis for the language of analysts' communications among themselves. The latter is tacit knowledge (Polanyi, 1967) that is difficult to articulate and describe, but is essential in accomplishing their work. This dichotomy is similar to Orr's (1996) copy repair technicians, who need both the general knowledge of a certain class of machine, balanced with a local knowledge of the idiosyncratic operating environment of each machine. This section discusses these two different types of knowledge that make up ID expertise, their acquisition, and the resultant problem of transferring that expertise to new contexts. Analysts' skill development is explicated in the conclusion using the Dreyfus model of expertise. (Dreyfus and Dreyfus, 2005)

Domain Knowledge: "Playing Around"

> From my own experience, it's been fun just watching my own local wire. I mean, just a handful of machines, so if something happens there I'm not like, ahhhh. Because nothing [malicious] ever does [happen], but is interesting. (P4)

This analyst installed an IDS on his small home network to see what the traffic looks like. He does so not necessarily to increase security to his home network, but because it is an interesting learning opportunity. Mastering ID requires long learning cycles, as analysts need to learn not only the intricacies of network protocols and system operations, but also the manifestations of these within a particular environment that is constantly changing. Achieving this expertise first requires foundational domain knowledge in networking and security. Few of the participants had any formal education, training, or certification in networking or security, and those who did described it more as an afterthought than as part of the learning process: "after the fact, after a couple of years on the job" (P6). The participants learned primarily by experimenting, and

also by self-directed readings related to ID, especially mailing list archives, but also web sites, protocol descriptions, and documentation.

The learning process generally begins by experimenting with network packet capture tools away from the work environment to examine the data passing through a network and decipher what the low level details mean. Once a proficiency in networking has been reached, the participants describe beginning to experiment with an IDS: "[using] Snort, I'm going to learn a lot by building the packets, and learn a lot about protocols, and get a better understanding of how hackers try to get into the system" (P5). All the analysts interviewed enjoyed their work – especially the challenge of mastering these complex technologies.

All participants shared a similar ethic of learning by doing, a self-taught, self-motivated approach that spoke of a genuine interest in learning about security and intrusion detection. This active, hands-on approach in which the analyst actively builds their ID knowledge follows a constructivist approach to learning (e.g., Bruner, 1990). This learning process occurred either in the workplace or in an outside environment. Many, like P4, described this experimentation process not as part of their job, but as a fun activity they carried out on their own time. This was very common among analysts and was often referred to as "playing," as in these two excerpts:

> *How did you learn intrusion detection, to set it up, administer it [IDS]?*
> I got a little into firewalls, and decided to play with Snort. Writing [signatures], and setting it up. Did stuff like Nessus [vulnerability scans] to test against it, Snort. Sticking Honeypots on the Internet. Playing around with it. Here and at home. (P5)
>
> *How did you learn to set up, administer, and use the intrusion detection systems, besides the SANS training?*
> Besides SANS, time on my own, you know at home, and at the office, just testing and playing with it. (P9)

These analysts are typical. They play with an IDS on their own time, often at home, learning by setting it up and then trying to understand the results. The first excerpt describes using Nessus, an open source vulnerability scanner, to test the IDS; this permits seeing the network traffic from both viewpoints – as an attacker would, using Nessus, and as a defender would, using the IDS. The second analyst describes this same type of interest in learning through playing and experimentation, although he also had the benefit of training.

Situated Knowledge: "Tuning"

> `What works on your network will not work on mine. (mailing list post)`

This informal learning process of "playing" lays an essential foundation of requisite knowledge, but by itself it is not sufficient to accomplish the work of ID, which depends on the analyst's understanding of his or her own idiosyncratic operating environment. The participants repeatedly described the most important aspect of ID as "knowing the environment" (P8). This learning process starts with a general familiarity with the local network itself and the systems and services running on it. It extends to interpreting the meaning of IDS alerts from different parts of the local environment, putting events into context and understanding if an event is abnormal by instinctively knowing what is normal. This expertise comes from practice: informal learning or experimentation on home networks is not enough; it must be lived, real-time on the job. This situated learning occurs in the context of the analysts' everyday activities (Lave and Wenger, 1990).

The process of learning by doing continues as a way for analysts to familiarize themselves with the nuances of a particular environment. This process is often described as "trial and error" (P7) and "on-the-fly education" (P6) as analysts learn how to configure an IDS in their work environment. It is often through the ongoing customization of the IDS that an analyst gains the situated expertise required to accomplish ID. Tuning the IDS acquaints analysts with what is normal in their environment. Continually tuning the signature set of the IDS involves removing signatures that do not apply to the environment or generate false positives, and writing new signatures to detect attacks peculiar to a specific operating environment. The initial installation of an IDS is trivial compared to the tuning process; the tuning process itself, which begins as soon as the system is turned on and continues for the life of the system, is the fundamental

design-like activity in ID work. It is through this reconfiguration that analysts demonstrate what constitutes normal on their network, as in “obviously [this alert] is not an attack, it’s very normal web traffic for us” (P5).

There is a circularity at play here – ID analysts need to know the environment in order to effectively tune the IDS, yet it is through this tuning of the IDS that they effectively learn the environment. The inherent difficulty of this, exemplified in the following mailing list post, generated 36 replies over a three-week period:

> My company is testing a few intrusion detection & prevention products. On the first few hours/days after deployment the machines alert on ten of thousands of events, which is way too much for us to ever go through, most of which are false alarms.
> The vendor’s solution is tuning the systems, which means shutting down signatures, detection mechanisms, omitting defragmentation tests and so on. These tunings do reduce dramatically the number of alerts, but it seems most of the detection capabilities have been shut off too, so things are nice and quiet [sic] but we've no idea what's really going on in our network apart from catching the trivial threats such as old worms, which don’t get false alarms.
> Has anyone encountered this situation? Anyone got a solution?

The majority of the community’s responses to this novice’s plea for help referred to the importance of knowing the environment. One typical response was:

> In my experience *it's much easier to tune a system if you are intimate with the network involved*. For example, on a Microsoft network you'll know where your DC's [Domain Controllers] etc. are located. If you are getting a high volume of alerts on the netbios or SMB ports from your internal hosts to your DC's then you could filter those servers as destinations and/or filter your internal network ranges as sources.... Tuning is a never ending process unless your network remained completely static. (Emphasis added)

An intimate understanding of the networking environment is the key to the ongoing process of tuning an IDS. Other responses generated similar examples to illustrate the importance of knowing the environment:

> ...an http connection from your internal network to the Internet. Is it malicious? From the IDS/IPS point of view, as long at this does not match any predefined malicious pattern, the traffic is OK. But maybe you realize that this is your mail server, and in your corporation, this mail server isn't supposed to establish http connections...
>
> Of course, you might have a particular configuration causing a predefined rule reporting a false positive, or a rule alerting you of real attacks, which however you know are not a real threat because you don't have any of the O.S. or applications the affect.
>
> The reason why IDS/IPS don't give useful information under these circumstances is always the same: they lack the capability to acquire and process all information that is specific to our business environment, which is also necessary to perform reasonably well.
>
> Therefore, no matter what many vendors say, you shouldn't expect any IDS/IPS to perform reasonably well without tuning. This of course is also the reason why many organizations don't like IDS/IPS, it is definitely time consuming.

This post demonstrates the need for the analyst to know what is normal, and to use that knowledge to tune the IDS. An untuned IDS does not have the required context of the operational environment to “know” that a mail server should never make outgoing http (web traffic) connections, even if that traffic is otherwise innocuous. The analyst brings that knowledge to bear on the problem, which is reflected in their tuning of the IDS to define what is permissible and what is potentially dangerous. This tuning process takes time and demands an intimacy with the networking environment, but it also gives analysts an opportunity to become intimate with the environment.

Thus, as the locus of work, the IDS signature file becomes both a physical manifestation of and a visible social marker of the analyst's expertise, in a role similar to that of blueprints in manufacturing environments (Bechky, 2003). What is different is that signature files function less as a boundary spanning object,

mediating amongst different actors such as design and manufacturing engineers, and more as a personal reflection of a given analyst.

In addition to the continuing tuning process, many analysts configure other components of their environmental infrastructure as part of their systems or network administration duties. This experience affords them local knowledge of the operating environment that is leveraged in their ID tasks. The following example demonstrates the value of knowledge gained by playing the dual role of ID analyst and system administrator:

> An attack against an IIS-based web server is a pretty severe alert, but if I have a filter that sits on that web server that just throws those things [i.e., network packets] aside before it gets passed to the web engine, then I consider that a false positive. You know, me knowing my environment I can say, this web server, I set up this web server, it's fine. (P3)

When a host receives an alert describing an attack targeting its specific type of server, in this example, the Internet Information Services (IIS) web server, there is normally immediate concern. However, the analyst knows that the particular targeted machine has a filtering mechanism in place that cancels the request that generated the alert. The analyst knows this because he set up the web server and knows his environment; if another analyst were to get the alert, he might have a very different reaction, since in and of itself it is, as the analyst in the above quote said, "a pretty severe alert." The analyst knows the environment and thus was able to judge the severity of the alert in context.

Transferring Expertise

> When I do consulting work, it [the learning] gets to be really hard. You come in and …they say, 'well look, I'm getting all these alerts' and I'm like, I know, but that's normal on my network, what the hell?" (P3)

The situated nature of ID expertise makes transferring it to other organizational contexts problematic. An illustration of this is the challenge of ID consulting; the analyst in the example above relayed his experience as a consultant expected to immediately diagnose the source of a high number of alerts in a new environment. This is different from system administration work. For example, configuring an Apache web server is an identical process in most environments because the same tools and processes are in place. The difference from one environment to another comes from the different design requirements for those environments, but the process in configuring the server is identical; a system administrator does not need to know the nuances of an environment, she only needs to be able to translate the required capabilities into a configuration language the system can understand. In ID work, then, learning the environment – not the tools and processes, which are learned once – is the challenge.

The difficulty in transferring expertise is also reflected in the challenges of introducing new analysts into an environment:

> *Interviewer: So an operator couldn't come in and see your console and get that much out of it?*
> P3: They'd probably shit their pants! 'All these alerts, what's going on?' And I'm like, 'oh, this is fine – alright I'm going to lunch.'

As discussed above, knowing what constitutes normal is essential to detecting what is abnormal – the fundamental task of ID. In order to be able to define normal and abnormal, a deep understanding of the operating environment is required, but unpacking this understanding is problematic.

The analysts themselves have trouble describing exactly what this contextualized knowledge is, or how one knows that they have it, is difficult for the analysts:

> I have developed this innate knowledge of what's good and what's bad and I have a terrible time articulating it to you, let alone to some of the interns we have wandering in here. (P6)

This tacit knowledge creates challenges for novice analysts as well as for tool designers.

CONCLUSIONS

Intrusion detection is markedly different from other system and network administration work. While there are elements of both proactive design and reactive monitoring, ID is fundamentally an event-driven process. ID lacks the opportunity for self-paced, reflective design. The sole design-like activity, tuning the IDS, is itself an event-driven process, typically triggered by a false positive IDS alert. Unlike the trend in much of system administration work of coping with the increasing size of networks by automating as many manual processes as possible, ID work remains a human-driven process. Successful ID implementations are not pure automated solutions like IPS, but rather a fully integrated socio-technical system; tool designers and vendors need to acknowledge the crucial role of human expertise in ID work. IDSs cue human analysts and provide them with targeted (but often overwhelming) network data to aid in their investigation. Because of this human-centered nature of ID work, it is especially important to understand the types of expertise needed and the ways this expertise is acquired. Understanding these will help ID tool designers, trainers, managers, and ID practitioners.

Intrusion detection analysts learn the *requisite foundational knowledge* of networking, security, and ID through a process often referred to as playing, typically on their own time within their own home networks. This playing process reveals the self-directed, self-motivated learning process. While this foundational knowledge is crucial, few practitioners were trained or educated in networking or security, and those who were described the training as an afterthought. The acquisition of foundational knowledge could thus be described as a byproduct of the analysts' interest in and enthusiasm for security work. This type of foundational knowledge of networking and security, however, unlike local knowledge, *could* be learned in outside of the analysts' environments.

The ongoing IDS tuning process is at the core of developing the *situated expertise* required to accomplish ID work. The maturing of this expertise aligns well with the Dreyfus model of skill development (Dreyfus and Dreyfus, 2005), in which novices learn in context-free environment and follow explicit rules for determining actions. They then may progress through successive stages of development (advanced beginner, competence, proficient) to the point where they can take action intuitively and confidently. While novice security analysts may have an understanding of the fundamentals of the field, they follow a rule-based approach to ID, much like an IDS itself; that is, they respond to a given alert the same way regardless of the context. Advanced beginners engage the complexity of different operating environments, often building confidence in safe, constrained settings like their own home network. At they reach the competence stage, analysts face an overwhelming number of elements; in ID work, lacking knowledge of what is important in a given situation dictates that analysts rely on the *a priori* priority of IDS signatures. This approach, however, almost ensures that attacks will be missed. For or analysts to develop further they learn to take risks by choosing which alerts warrant attention. When these high-risk decisions of prioritization occur through intuition rather than explicit decision making, the analysts have reached the stage of proficiency. As ID analysts become experts with deep knowledge of their network, they may immediately disregard an alert based on their intuition and situated expertise, as demonstrated in the statement "obviously [this alert] is not an attack, it's very normal web traffic for us" (P5). This reflexive knowing of "normal" and its guidance through resolving crisis situations is what differentiates experts from novices, beginners, and IDSs themselves.

The contextual nature of ID work and the difficulty in transferring expertise also differentiate ID from administration work, which is largely transferable between environments. Knowing how to configure and monitor a Linux server running a Sendmail server is nearly identical no matter what the environment. By contrast, being able to tune an IDS is wholly dependent on having an intimacy with the environment that it operates in; even being able to analyze and diagnose an event requires "knowing the environment." (P8) ID work certainly requires an understanding of knowledge about networking protocols, attack methods, and system configurations, but these context-free skills are only the foundational. ID work, to a large extent, is not about the technical skills or domain knowledge, but about being familiar with the environment being defended. This familiarity can, in many cases, be intimate; given an IP address analysts can often recall the purpose of applications running on dozens of computers without relying on any external memory aids.

Because of these differences, tool designers should be focusing on developing tools that support human diagnosis of intrusions as opposed to automating the human out of the loop as is typical with system

administration work. This trend is reflected in the current marketing of IPS as the next generation of IDS that automatically detects and blocks attacks as they happen. In reality, despite most of our participants having a background in system administration, all of those in our study had a deep mistrust for automated response.

Analogous to the findings in Muller's (1999) study of telephone operators' work, automation is not desirable and could actually obstruct successful intrusion detection. Situated expertise is acquired through the manual processes that make up ID work; without the tuning process, for example, analysts would not have occasion to learn the idiosyncrasies of their networks. On the other extreme, analysts need some sort of automation to help deal with the expanding size of networks. Instead of the blind drive towards automation, ID tool designers should focus on a "third path" that takes advantage of the automation capabilities of computers and acknowledges the human basis of successful ID.

Multiple possible solutions suggest themselves. One would be to develop systems that increase the amount of data that humans can quickly and accurately interpret, and provide support for translating that human interpretation into rules that can modify automated systems such as IDSs. Information visualization offers one promising approach to increasing the efficiency of human processing for this type of data. We have used this approach in designing a tool to help analysts manage copious amounts of raw network traffic in order to learn what is normal in their network (Goodall et al., 2006). The next logical step for tool designers is to use innovative visualizations to drive the tuning process by using the understanding analysts glean from the analytic process to create and update the rule bases in automated systems. This would facilitate the human analysts in learning their environment, while also automating the mundane aspects of manipulating esoteric signature syntax.

A second promising solution is systems to collect, fuse, and make available to analysts some portion of the contextual data required for decision making. Such a system could collect data passing through computer networks or collect data directly from systems using software agents. These data streams could be fused together to provide a unified data store to facilitate the development of situated expertise and provide a repository for such knowledge. For example, network data can reveal interdependencies between systems. System data could include the operating system, the applications and services running on a system, and details about the active system users. This contextual information is necessary to ID work, and is typically manually collected and stored in analysts' memory. Automatically making this contextual information available to analysts would help lessen analysts' overburdened memory load. While such a system would clearly not be a substitute for developing expertise through practice, it could facilitate learning some of the required situated knowledge more quickly. This process cannot collect the entire context required in ID work, nor should it. The process of tuning is crucial to developing situated expertise, but making some of the context more readily available to analysts could help them deal with ever expanding networks. It also could assist novices in coming up to speed, albeit still following simple rules rather than the intuition of an expert.

This research also has practical implications that extend beyond system design to information security training programs, information security management, and the practitioners themselves. As much as possible, training programs should be conducted on-site within an analysts' network, emphasizing not just the technology, but also the process of ID, particularly the importance of experimentation in the tuning process. While training is not a substitute for self-learning, it can help to guide novices in learning how to develop the crucial knowledge of their own networks.

Management should also be aware of the need for analysts to learn, and provide the time and resources for analysts to play and experiment. One means of providing this might be to give analysts a "sandbox" machine on the network that would allow analysts to experiment and learn without the risk of accidentally misconfiguring a production machine. This process is crucial to learning the network and should be an integral part of sanctioned practice.

Management also needs to be aware that even expert analysts cannot immediately step into a new environment and be expected to perform well before they become intimately familiar with the idiosyncrasies of that environment. The analysts themselves should be aware of the importance of learning the networking environment and strive to develop new mechanisms to facilitate this process, such as

communicating regularly with the administrators who configure and manage computing resources and the network infrastructure.

This research has identified two types knowledge and the process of acquiring them. We described the practical implications for system design, organizational and management practices, and the practice of both novice and expert practitioners. Designers, managers, and practitioners should be cognizant of the role of expertise – how it is developed and maintained, and the problems of its transfer to different environments – and use that understanding to design tools and policies that fit with practice. They should actively support expertise development and situated learning even if there is no immediately measurable productivity benefit, and construct innovative methods to aid in acquiring the domain and situated expertise required to do their work.

REFERENCES

CSO magazine / U.S. Secret Service / CERT Coordination Center (2005), "2005 e-crime watch survey". http://www.csoonline.com/info/ecrimesurvey05.html.

Anderson, E. A. (2002), "Researching system administration", University of California, Berkeley.

Barley, S. R. (1996), "Technicians in the workplace: Ethnographic evidence for bringing work into organizational studies", *Administrative Science Quarterly*, Vol. 41 No. 3, pp. 404-441.

Barrett, R., Kandogan, E., Maglio, P. P., Haber, E. M., Takayama, L. A. and Prabaker, M. (2004), "Field studies of computer system administrators:Â analysis of system management tools and practices", *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW)*, pp. 388-395.

Bechky, B. A. (2003), "Sharing meaning across occupational communities: The transformation of understanding on a production floor", *Organization Science*, Vol. 14 No. 3, pp. 312-330.

Bentley, R., Hughes, J. A., Randall, D., Rodden, T., Sawyer, P., Shapiro, D. and Sommerville, I. (1992), "Ethnographically-informed systems design for air traffic control", *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW)*, pp. 123-129.

Bruner, J. (1990), *Acts of meaning*, Harvard University Press, Cambridge, MA.

D'Amico, A., Whitley, K., Tesone, D., O'Brien, B. and Roth, E. (2005), "Achieving cyber defense situational awareness: A cognitive task analysis of information assurance analysts", *Proceedings of the Human Factors and Ergonomics Society 49th Annual Meeting*, pp. 229-233.

Dreyfus, H. L. and Dreyfus, S. E. (2005), "Peripheral vision: Expertise in real world contexts", *Organization Studies*, Vol. 26 No. 5, pp. 779-792.

Goodall, J. R., Lutters, W. G. and Komlodi, A. (2004), "I know my network: Collaboration and expertise in intrusion detection", *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW)*, pp. 342-345.

Goodall, J. R., Lutters, W. G., Rheingans, P. and Komlodi, A. (2006), "Focusing on context in network traffic analysis", *IEEE Computer Graphics and Applications*, Vol. 26 No. 2, pp. 72-80.

Hall, M. (2007), "Price of security breaches...". http://www.computerworld.com/securitytopics/security/story/0,10801,106180,00.html.

Halprin, G. (1998), "The workflow of system administration", *Proceedings of the Conference of the System Administrators Guild of Australia (SAGE-AU)*.

Harper, R. H. R. (1998), *Inside the IMF: An ethnography of documents, technology and organisational action,* Academic Press, San Diego, CA.

Heath, C. and Luff, P. (1992), "Collaboration and control: Crisis management and multimedia technology in london underground control rooms", *Journal of Computer Supported Cooperative Work,* Vol. 1 No. 1, pp. 24-48.

Hughes, J., Randall, D. and Shapiro, D. (1992), "Faltering from ethnography to design", *Proceedings of the ACM Conference on Computer-Supported Cooperative Work (CSCW),* pp. 115-122.

Julisch, K. (2003), "Clustering intrusion detection alarms to support root cause analysis", *ACM Transactions on Information and System Security,* Vol. 6 No. 4, pp. 443-471.

Kirk, J. (2007), "Estonia recovers from massive denial-of-service attack". http://www.networkworld.com/news/2007/051707-estonia-recovers-from-massive-denial-of-service.html.

Lave, J. and Wenger, E. (1990), *Situated learning: Legitimate peripheral participation,* Cambridge University Press, Cambridge, UK.

Lings, B. and Lundell, B. Ã. r. (2005), "On the adaptation of grounded theory procedures: Insights from the evolution of the 2G method", *Information Technology & People,* Vol. 18 No. 3, pp. 196-211.

Lutters, W. G. and Ackerman, M. S. (2007), "Beyond boundary objects: Collaborative reuse in aircraft technical support", *Journal of Computer Supported Cooperative Work,* Vol. 16 No. 3, pp. 341-372.

Martin, D. and Sommerville, I. (2004), "Patterns of cooperative interaction: Linking ethnomethodology and design", *ACM Transactions on Computer-Human Interaction,* Vol. 11 No. 1, pp. 59-89.

Muller, M. J. (1999), "Invisible work of telephone operators: An ethnocritical analysis", *Journal of Computer Supported Cooperative Work,* Vol. 8 No. 1/2, pp. 31-61.

Polanyi, M. (1967), *The tacit dimension,* Anchor Books, New York.

Roesch, M. (1999), "Snort - lightweight intrusion detection for networks", *Proceedings of Thirteenth Systems Administration Conference (LISA),* pp. 229-238.

Sandusky, R. J. (1997), "Infrastructure management as cooperative work: Implications for systems design", *Proceedings of the ACM Conference on Supporting Group Work (GROUP),* pp. 91-100.

Star, S. L. and Strauss, A. (1999), "Layers of silence, arenas of voice: The ecology of visible and invisible work", *Journal of Computer Supported Cooperative Work,* Vol. 8 No. 1/2, pp. 9-30.

Stolze, M., Pawlitzek, R. and Hild, S. (2003a), "Task support for network security monitoring", *ACM CHI Workshop on System Administrators Are Users, Too: Designing Workspaces for Managing Internet-Scale Systems*.

Stolze, M., Pawlitzek, R. and Wespi, A. (2003b), "Visual problem-solving support for new event triage in centralized network security monitoring: Challenges, tools and benefits", *GI-SIDAR conference IT-Incident Management & IT-Forensics (IMF)*.

Strauss, A. and Corbin, J. (1998), *Basics of qualitative research: Techniques and procedures for developing grounded theory,* SAGE Publications, Thousand Oaks, CA.

Yurcik, W., Barlow, J. and Rosendale, J. (2003), "Maintaining perspective on who is the enemy in the security systems administration of computer networks", *ACM CHI Workshop on System Administrators Are Users, Too: Designing Workspaces for Managing Internet-Scale Systems*.