Towards Understanding Diagnostic Work During the Detection and Investigation of Security Incidents

Rodrigo Werlinger[1], Kasia Muldner[2], Kirstie Hawkey[1],
Konstantin Beznosov[1]

University of British Columbia[1], Arizona State University[2]
{rodrigow,hawkey,beznosov}@ece.ubc.ca; kmuldner@asu.edu

Abstract

This study investigates how security practitioners perform diagnostic work during the identification of security incidents. Based on empirical data from 16 interviews with security practitioners, we identify the tasks, skills, strategies and tools that security practitioners use to diagnose security incidents. Our analysis shows that diagnosis is a highly collaborative activity, which may involve practitioners developing their own tools to perform specific tasks. Our results also show that diagnosis during incident response is complicated by practitioners' need to rely on tacit knowledge, as well as usability issues with security tools. We offer recommendations to improve technology that supports the diagnosis of security incidents.

Keywords

Diagnosis, Security Incident Response, Qualitative Analysis, Collaboration.

1. Introduction

Diagnostic work, i.e., the practice of noticing and categorizing problems, as well as defining the scope of remediation, is a pervasive feature of Information Technology Security Management (ITSM). Diagnosis is particularly prevalent during security incident response, one of the primary responsibilities of security practitioners (Botta et al. 2007b, Kandogan & Haber 2005). Despite its prominence as an activity, the field of security incident response is still in its infancy (Killcrece et al. 2005). In fact, based on a retrospective comparison of the 1998 Internet Worm incident with the state of IT security in 2003, Spafford (2003) concludes that several security-related aspects worsened during that time. In particular, Spafford highlights that the security community has been unable to learn the importance of communication during incident response. He proposes that the security community should find better ways to not only coordinate during incidents, but also to distribute incident-related information. While a number of organizations provide guidelines for the incident response process (e.g., Computer Emergency Response Team (CERT), National Institute of Standards and Technology (NIST)), there are few empirical investigations on how security practitioners respond to incidents (for exceptions, see, for instance (Goodall et al. 2004a, Riden 2006)). The research presented in this paper aims to fill this gap.

Our results extend the findings of Werlinger et al. (2009), who identify nine activities that require security practitioners to interact with other stakeholders, one of which is security incident response. We extend those results by (1) analyzing security incident response from a broader perspective, rather than focusing only on interactions, and (2) identifying the diagnostic aspects during interactions involved in the detection and investigation of security incidents. Specifically, the contribution of our work presented here is twofold. First, using empirical data, we analyze and describe the tasks, skills, strategies, and tools that security practitioners use to diagnose security incidents (Section 4). Our findings enhance the research community's understanding of the diagnostic work during security incident response. Second, we identify opportunities for future research directions related to improving security tools (Section 5). For instance, our analysis shows that regardless of how advanced a security tool is for supporting diagnostic work, practitioners must still customize that tool to fit the specific needs of their organization. Today's tools, however, provide very little if any support for this customization process. We next present the related work (Section 2), followed by an explanation of our study methodology (Section 3).

2. Related Work

Given the challenges of managing security incidents, a number of guidelines (e.g., (Casey 2002, Stephenson 2004)) and associations (e.g., CERT and NIST) exist that provide support for the incident response process. Recently, Mitropoulos et al. (2006) synthesized the information from the various standards and existing research to propose a general incident response management framework. While these various efforts may provide some support for the incident response process, Bailey et al. (2007) discuss how best practices and formal standards for IT work tend to be either so high level that they provide little guidance on work practices, or so low level that they are inflexible to rapid changes in the technology and organization.

One of the tools designed to support practitioners during the detection of security incidents is an intrusion detection system (IDS). Goodall et al. (2004b) and Thompson et al. (2006) rely on data from nine and two semi-structured interviews, respectively, to identify the phases of intrusion detection work. Goodall et al. (2004b) suggest that intrusion detection is challenging due to the need for analysts to coordinate with other stakeholders and the need for high expertise, both technical and organizational. Werlinger et al. (2008) analyze data from nine interviews to identify security practitioners' perceptions of the advantages and disadvantages of IDSs. They also analyze data from participatory observation to show that IDS usability is hindered by lack of technical resources and ITSM's distributed nature.

Some research focuses on descriptive case studies of real-life examples related to security incidents. Casey (2005) presents a case study of an intrusion against one organization and stresses the role of collaboration during incident diagnosis and containment. Gibson (2001) describes a denial of service attack on his company. The diagnosis of the incident included both technical troubleshooting as well as interaction with various parties. Riden (2006) describes a series of security incidents on a large academic network. Key factors contributing to the incidents included ineffective communication and collaboration between the organization's security professionals, which led to inconsistent preventative measures and untimely notification of vulnerabilities. Schultz (2007) describes a variety of sources of information that had to be combined in order to diagnose an incident in one organization. While these case studies can provide useful data, they only involved a single organization, and have not relied on formal evaluation methodologies to collect and analyze their data. As far as we are aware, the only formal studies that exist investigate a small subset of security incident response, namely a specific tool used to detect security incidents (an intrusion detection system), as described above.

Table 1 : Participant Information

Organization Type	Position Type			
	Security manager	**Security specialist**	**IT Practitioner with Security Tasks**	**Total**
Academic (3)	I2	I3, I9, I11, I24	I7, I8, I22	8
Financial Services (1)	-	I4	-	1
Scientific Services (1)	-	-	I12, I13	2
Manufacturing (1)	-	I21	-	1
Telecommunications	-	I32	-	1
IT Consulting Firm (1)	-	-	I26	1
Insurance (1)	I38	I39	-	2
Total	2	8	6	16

We can also draw from research of diagnostic work within organizations. Orr (1986) investigated the diagnostic process for copier repair and found that story telling was used both as a cooperative diagnostic activity and later to provide organizational knowledge of interesting cases. Yamauchi et al. (2003) described the problem-solving practices of service repair technicians and found that they rarely followed instructions from existing documentation, but rather gleaned information from a variety of sources such as colleagues, systems, and informal documents.

3. Methodology

We framed our study with the following research questions: (1) How do security practitioners perform diagnostic work when responding to security incidents? (2) What tools do security practitioners need to perform this type of diagnostic work? (3) How can such tools be improved to better support security practitioners?

To answer our research questions, we analyzed our interview data corpus from the HOT Admin project; see Hawkey et al. (2008) for an overview of other themes of analysis. HOT Admin researchers have conducted 39 semi-structured in situ interviews with security practitioners, who worked for a variety of organizations (11 different organizations from 7 sectors). Participants were asked a variety of security-related questions (e.g., ITSM challenges, ITSM tasks and tools, organizational influences, to name a few). Each interview lasted approximately one hour and was subsequently transcribed and sanitized to preserve the participants' anonymity. As is typically the case with semi-structured interviews, not all participants were asked the same questions, and not all discussed topics relevant to our research questions on diagnostic work during security incident response. Table 1 summarizes information on the 16 participants who did discuss diagnostic work and whose data we considered for the analysis presented here. For presentation purposes, we identify our interview participants according to their interview number (i.e., I1...I39).

We used qualitative description (Sandelowski 2000) to analyze our data, as follows. First, we analyzed the interview transcriptions to identify excerpts pertaining to diagnostic work, focusing on work related to security incidents. We used CERT's definition of a security incident: "*any real or suspected adverse event in relation to the security of computer systems or computer networks*" (Killcrece et al. 2003). Second, we organized the excerpts into different stories or "memos" (Charmaz 2006) describing how security practitioners perform diagnostic work and the key challenges they face during this process.

4. Results

We first provide an overview of the diagnostic process during security incident response (see Figure 1 (A, B, C, D), adapted from Werlinger et al. (2009)). This process starts with the *detection* of an anomaly in an organization's IT systems (e.g., users experiencing slow access to Internet). During this process, our participants performed two types of activities: *monitoring* (Figure 1, A.1) and sending and receiving *notifications* (Figure 1, B.1 and C.1). Monitoring involves intensive use of IT tools (e.g., IDSs, antivirus) and requires knowledge to identify patterns of anomalous activity in the networks. Such knowledge is often *tacit*, in that people are unaware of possessing it and/or how it could be valuable to others; furthermore, tacit knowledge is not easily shared (Polanyi 1966). Notification involves extensive collaboration with other stakeholders, who are either directly monitoring systems or indirectly receiving notifications from other stakeholders. After noticing an anomaly in the IT infrastructure, participants moved to *analysis* of the anomaly. This stage included diagnostic tasks such as: *verification* (Figure 1, A.2), *assessment* (Figure 1, A.3), and *tracking the source of the anomaly* (Figure 1, A.4 and B.2). To perform these tasks, participants required effective (i) communication skills to collaborate with other stakeholders and (ii) analytical skills to generate hypothesis about the causes of the anomaly. When the cause of the anomaly was found, participants moved to containing the incident. We now describe these activities.

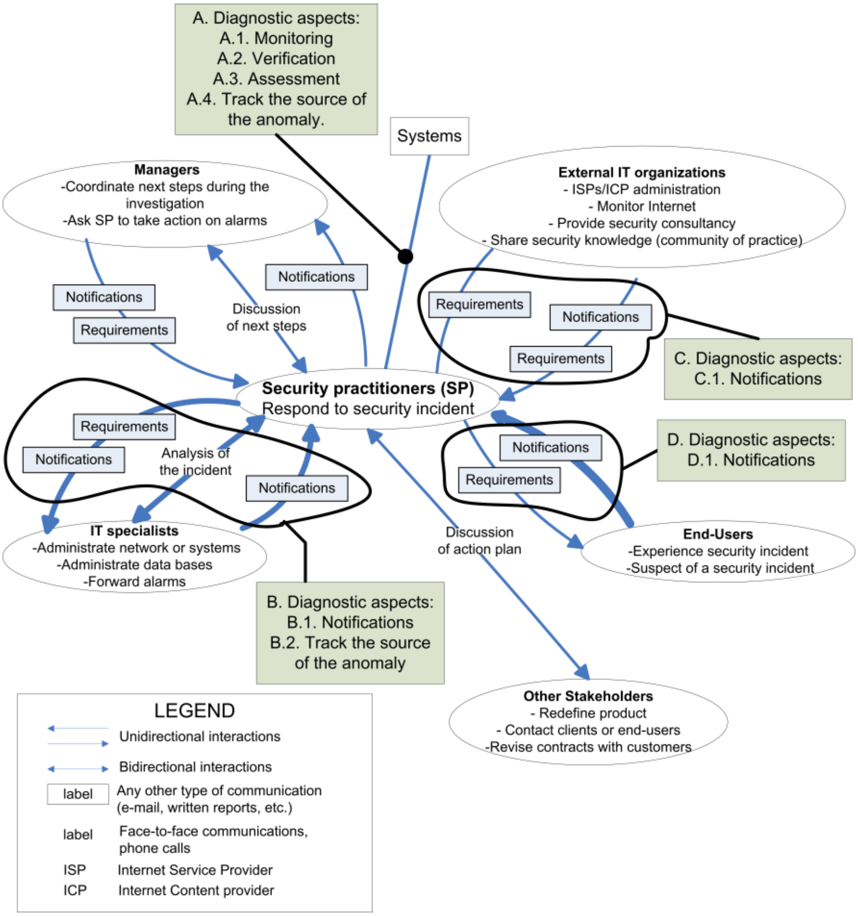

Figure 1 : Collaboration among stakeholders during security incident response, thicker arrows indicate more frequent collaboration. Diagnostic aspects are highlighted (A, B, C, D).

4.1. Detection of an Anomaly

To detect security incidents, our participants actively monitored their organizations' IT systems (Figure 1, A.1). Monitoring involved a variety of tools, as well as tacit knowledge about the organizations' IT systems and services. For example, one participant (I3) knew that end-users in his organization typically generate less than 50 e-mails per day, and so a higher number of e- mails signaled a potential anomaly. Examples of the tools security practitioners used to monitor IT systems included antivirus software and IDSs. Antivirus software was used to detect viruses and to generate reports about virus activity in the infrastructure (I3, I4, I12, I24). IDSs were used to 'sniff ' network traffic to find matches with the signatures of known attacks.

The usability of monitoring tools hindered their effective use. For instance, some participants (I4, I9, I12, I24) found it very challenging to use IDSs to generate meaningful reports on monitoring outcomes, largely due to the overwhelming amount of false positives generated by IDSs (for a more in-depth discussion of IDS usability, see Werlinger et al (2008)). To reduce false-positives, an IDS needs to be customized to fit a given organization's characteristics, a time-consuming and difficult process that some of our participants preferred to avoid (I3, I4, I9, I24, I12). Other monitoring tools were less complex than IDSs, although these tools also suffered from usability issues. For example, SmokePing was used to identify when systems were up or down (I13). This tool minimized false positives, and its output was easy to interpret. The tool, however, also had a disadvantage, namely that the alarms it generated did not include any information on the cause of the problem.

As the above examples demonstrate, IT tools typically have pros and cons. In some in- stances, security practitioners combined tools in unique ways to maximize their utility. For instance, one participant (I12) combined two tools (TCPDump and Ethereal) to generate and analyze, respectively, the log files he needed. He alternated between the advantages of portability (TCPDump) and good visualization (Ethereal): *"[TCPDump provides] common analysis format ... it's also a portable format ... it [Ethereal] shows the SYN and RESET in one color and then the PUSH commands in another color. So it is obvious there is content in there*.".

The monitored traffic's characteristics also limited the usability of security tools, e.g., high-volume traffic made it impossible to monitor some network areas (I3, I24). In these cases, participants had to select specific networks to monitor, based not only on the capacity of the security tools but also on historical network data, i.e., where the most critical incidents occurred in the past (I37). Encrypted traffic created yet another constraint; without access to encrypted data, participants had limited options to detect malicious code in the packets (I36).

In-house tools. Due to usability issues and budget constraints, our participants often resorted to creating their own tools to detect anomalies in the IT infrastructure (I2, I3, I8, I9, I12, I22, I24). These tools were *scripts* - programs for the command interpreter of an operating system. One participant (I3) noted that scripts relieved the burden of manually analyzing raw log files. To create effective scripts, participants needed both technical expertise and knowledge about the IT infrastructure within their organization. For example, one participant (I3) could list the network addresses of the computers with suspiciously high number of e-mails; this allowed him to selectively monitor some systems more than others. The same participant developed a script to generate only one alarm upon detection of abnormal traffic, to avoid having vast volumes of alarms associated with the same anomaly. Another participant (I2) explained how he used scripts to detect denial of service attacks, and to notify the appropriate administrators, alleviating the burden of a practitioner having to deal with the notification (Figure 1, B.1).

Notifications. The complexity of IT systems and the lack of resources to monitor *all* systems meant that our participants relied on *notifications* to detect security incidents (Figure 1, B.1 and C.1). Our participants received notifications from various stakeholders, including IT professionals and end users. Often, these notifications required communication among stakeholders. For example, participant I12 described how an external organization (MyNetWatchman) had detected malicious traffic generated from one of the systems he administered (Figure 1, C.1). He received this notification from another colleague (Figure 1, B.1) who was notified by MyNet-Watchman. This chain of notifications among different security practitioners was also mentioned by a participant who was involved in a response to a phishing attack (I4): "*we had a person, not even a member of any of our organizations or customers, who emailed our privacy office ... then the privacy office contacted me directly*" (Figure 1, C.1). Another participant (I38) received notifications from his organization's problem management system. Our participants also received notifications about incidents from end-users (Figure 1, D.1), in the form of complaints that the Internet access was blocked (I11, I22). In some instances, monitoring and/or receiving notifications led security practitioners to the detection of anomalies, and their subsequent investigation. In one organization, Microsoft's monthly patch release day was treated as notification of a security incident to initiate coordination with the relevant stakeholders (I38).

4.2. Analysis of an Anomaly

Once a potential anomaly was detected, practitioners investigated it further, which comprised at least three tasks: *verification* (Figure 1, A.2), *assessment* (Figure 1, A.3) and *tracking the source of the anomaly* (Figure 1, A.4 and B.2).

Anomaly Verification. During anomaly *verification*, participants tried to confirm, often with alternate data sources, that a compromise actually occurred. One (I3) described this verification: "*I always try and verify by a second or third source. So [I would] go back to the Argus [IDS] .. check the Argus logs and see what's actually happened; .. then I would go to one of my other logs; what have I seen in the logs of the Windows box; was that a real compromise or not*." Verification may also require collaboration with external organizations. One participant (I20) was investigating traffic from an external server that was generating malicious traffic to his organization. With the external organization's consent, he used *nmap* to determine the ports that the server had open. This showed him that the server had been compromised, which led him to access the server to check its internal status. Another (I28) performed similar steps when dealing with a server that was generating high quantities of traffic to the Internet.

When participants had access to machines that stakeholders reported infected by malicious software, they did not necessarily need tools to confirm the infection. One (I26) used his experience to identify patterns that indicated the machine had malicious software (e.g., "funny" icons or processes running). He also explained how his experience taught him to run the tools to remove the malicious software at least twice. Another (I28) indicated how during verification, he relied on his experience to know what type of connection pattern was normal from one server to another: "*This is based on experience ... we consider [it] is normal [connections] from one public IP address [to] all websites. But if IP address goes to every port of the IP address and it is a website then this is not normal.*"

Anomaly Assessment. If an incident was indeed confirmed, during its *assessment*, security practitioners estimated the incident's magnitude and consequences (I3, I4, I39). In some organizations, the policy is for the potential cost of the incident to the organization to be communicated to managers who will make a determination of whether to proceed; however, one participant (I38) described how some incidents that did not meet the organization's criteria for high risk may still be investigated by the security team in order to protect their systems. One participant (I3) described the assessment process and how it shaped the next steps: "*I might go through the logs to see what kind of traffic I'm getting from this IP address–is it scans? is it a successful compromise? So it depends on what I find, depends on what I do.*" Another (I14) described assessing a phishing attack by checking how many e-mails were sent from the organization's e-mail server.

Tracking the Anomaly Source. In this step, participants aimed to determine the source of the incident. Two (I9, I12) used their knowledge about hacking patterns to diagnose the source of an anomaly related to malicious software. One (I9) mentioned that diagnosing denial of service attacks was straightforward and could be accomplished by inspecting the volumes of specific network traffic: "*denial of services are easy to spot, cause it's sending mil lions of the same thing actually over and over and over again, with very little iteration*". Another participant (I12) identified hacking activity by looking for specific type of traffic: "*there is some content here and it looks like IRC [Internet Relay Chat]. So I figure that this is somebody controlling it, the machine ... [IRC is] very popular with hackers as a control mechanism*". Participants also relied on their technical knowledge to perform forensic tasks on compromised servers. If the source of an incident was due to the actions of an internal employee, stakeholders within human resources may be contacted (I39).

When the source of an incident was difficult to diagnose, participants found it especially helpful to interact with other specialists, particularly ones who could offer a novel perspective as they were new to the investigation or had a different background. As an illustration, participant I13 had to investigate an incident related to loss of service from the organization's IT systems. He decided to check the systems in situ, and asked for help from another specialist, "*because two eyes are better than one*". However, the hardware looked normal, and they decided to involve another specialist in the analysis. She thought that the problem was with a small network switch that had not been checked during an earlier inspection; they reset the switch and the network recovered. Another participant (I11) described needing help from a specialist in a different department to trace the flow of traffic in an under-performing network. Through this collaboration, they were able to isolate the device that was slowing traffic: "*We also contacted IT services [to] see if they could see, based on traffic utilization on the network, where it was coming from ... we finally isolated—hey, it's that new firewall that we brought up*." In one organization, recent security incidents are discussed weekly so that security practitioners can learn about the current threats and help brainstorm the resolution of challenging incidents (I39).

In addition to collaboration, another strategy participants used to identify the cause of an incident involved simulation of the incident. One participant (I13) mentioned how he was collecting information from actual situations where he repeated the conditions of failure: "*So we try to put a proxy in between .. and then it started crashing ... [but] as soon as we put in no filtering ... bad things stop happening* " In another case, a participant (I12) wanted more specific information about the type of malicious traffic that was causing anomalies. He explained how he downloaded the same suspected malicious software to provide such information: "*It's saying ... downloading a tool from some website. Okay, so I do that, download this tool and run it through the antivirus and it says okay, this is some dial-up*".

Some of the security incidents we described were solved during the analysis process. In other instances, incident *containment* was necessary. This was accomplished in various ways, including: by turning off ports or services in external organizations (I4) and by cleaning up IT systems by reinstalling software (I9).

5. Discussion

Security incident response is a multi-faceted activity, where the corresponding diagnosis requires a mix of both strong technical and communication skills. Our participants faced many challenges when diagnosing security-related problems, at least some of which stemmed from insufficient tool support. We now rely on our analysis to offer suggestions on research directions for improving security tools, grounding our discussion in our participants' experiences and related work.

Task Complexity: A key challenge our participants mentioned pertained to security tools that monitored IT systems and generated alarms upon detection of anomalous events. These monitoring tools generated overwhelming numbers of false positives (i.e., alarms that corresponded to innocuous events), which placed a high burden on security practitioners who had to investigate the alarms. Our analysis suggests that task complexity influences tool reliability, and furthermore, that there is a tradeoff between the complexity of the task supported by a tool and the tool's reliability: the more complex the task, the less reliable the tool's output for that task. For example, IDS tools perform a variety of complex tasks; these tools generated many more false positives and so required more intervention from practitioners than SmokePing, a simple tool that only checked system availability. On the other hand, SmokePing's simplicity was not without disadvantages: its basic functionality meant that it did not provide information about incidents unrelated to the availability of systems, e.g., attacks to guess the users' passwords.

The above discussion highlights that the tradeoff between task complexity and tool reliability is a dimension that must be taken into account during tool evaluation. In particular, more research is needed to understand the pros and cons of security tools designed to perform complex tasks, as compared to tools that are intended for simple tasks. A second dimension that needs to be taken into account when evaluating tools is support for tool integration, as we describe shortly. First, however, we present a second factor influencing monitoring tool reliability.

Customization to Ensure Tool Fit: A practitioners' ability to configure a monitoring tool to a given organization's characteristics directly impacts the number of false positives produced by that tool. Recall that to configure monitoring tools, practitioners relied on generic lists of attacks and vulnerabilities. These are maintained by security practitioners around the world and are available on public servers (e.g., lists.sourceforge.net). Although these lists provide a good starting point and highlight the collaborative nature of ITSM, they correspond to huge quantities of generic data, making the customization task difficult for security practitioners. Lack of adequate tools and/or customization support also meant that our participants had to develop their own tools to perform tasks related to the diagnosis of security incidents. This illustrates how difficult it is to develop standard security tools that fit every organization's needs for the diagnosis of security incidents. Botta et al. (2007b) propose that security tools have to support *tailorability*, so that practitioners can customize tools via their own scripts.

The above discussion shows that no matter how advanced a security tool is, ITSM diagnostic work still requires customization of the tool to the specific reality of a given organization. The customization often requires access to a complete inventory of an organization's IT systems. Such an inventory is very costly to create and maintain, given the challenges of ITSM (Gagne et al. 2008). For instance, the dynamic nature of the IT environment means that systems are constantly being upgraded and/or replaced, requiring practitioners to continually update the system inventory. In general, to improve the efficiency of diagnostic ITSM work, more research is needed to investigate how the process of customizing a generic list of vulnerabilities could be optimized. One option is to rely on Artificial Intelligence techniques, and so have tools automatically adapt a generic vulnerability list to a given organization's characteristics (e.g., as is done in so-called anomaly-based IDSs).

As discussed, customization requires intensive use of knowledge that is typically not shared among practitioners and is often not explicitly documented. Gagne et al. (2008) suggest that it is necessary to provide support for transforming security practitioners' knowledge needed during tool configuration into explicit knowledge that can be shared with others. In addition to increasing the usability of a tool, support for customization via scripts has a second benefit: the scripts capture practitioners' tacit knowledge. This benefit was also noted by Halverson et al. (2004), who suggested that supporting the practice of bricolage (discussed next) can aid in the capture and transformation of knowledge within an organization.

Tool integration: Depending on the diagnostic work performed, our practitioners used scripts either as stand alone tools or in combination with other tools via bricolage, i.e., the re-use of existing tools in new and unanticipated ways. Halverson et al. (2004), who studied the trouble-shooting process at a helpdesk, discuss how the practice of bricolage for tools and processes is inherent to group work. Botta et al. (2007a) show that ITSM work in general involves bricolage, and our results illustrate how this skill is also practiced during diagnosis of security incident incidents. Note that bricolage is a special instance of vendor-designed tool integration.

How tools should be developed to support bricolage is an open question. Novel evaluation methodologies may be needed as there has been little study of how tool integration in general and bricolage in particular impact tool usability. Halverson et al. (2004) suggest that the practice of bricolage allows for reuse of expertise with the existing tools. However, integration must be considered in conjunction with task complexity, since the latter also impacts tool usability. To illustrate, bricolage support may be beneficial across the board, from simple to complex tasks; alternatively, bricolage could place high cognitive load on practitioners, making it only beneficial for complex tasks. We need to develop a richer understanding of the ways in which tools are used during diagnostic work when responding to security incidents; this understanding can support the specification of the complex scenarios in which these security tools should be evaluated (Redish 2007).

Verification of incidents via data correlation: To diagnose security incidents, our participants had to correlate different sources of information. To do so, they not only had to understand how various IT systems were related, but also needed security tools that were able to process and relate information from these different sources. To satisfy this need, security tools need to process information from a variety of sources with different formats and structure. For instance, a tool developed by Cisco, a major vendor of network devices and monitoring tools, can integrate with different tools to correlate information and generate consolidated reports.

In the same vein, security tools that integrate data need to process very large data volumes, which in turn must be reified in a meaningful way. Unfortunately, our participants found the on-line reports needed during diagnosis different to generate. To deal with this limitation, one option is to abstract the tasks of data synthesis and visualization away from the standard security tools towards specialized tools that only focus on these tasks. Abstraction has the advantage of providing a separation of functionality, i.e., raw data collection vs. data processing. This in turn provides flexibility to plug in a variety of devices into the specialized reification tools.

Multi-faceted simulation support: As we described above, diagnostic work during security incidents involves security practitioners performing simulations to verify or investigate an anomaly. Complicating simulation work is that in some instances, it needs to be performed in production systems that needed to remain operational. To address this issue, Fisler et al. (2005) describes an approach for a specific type of simulation involving access control rules. Along a similar vein, Chiasson et al. (2007) propose that any security-system changes should be easily reversible; this guideline ensures that any simulation-introduced problem in a production system is easily reversed. Our results show that diagnostic work during security incident response requires practitioners to perform simulations in distributed systems administered by various practitioners, and so requires collaboration. Since collaboration complicates the simulation process, we propose that tool support for simulation need to address not only the technical factors, but also include functionality that supports collaboration between different IT practitioners as they track the simulations and evaluate their consequences.

6. Conclusion

Our qualitative analysis shows the importance of diagnostic work during security incident response. The diagnostic process required active collaboration among our participants and other stakeholders. Participants used different technologies to support their tasks, developing their own tools when they did not have the required security tools for specific tasks. In our discussion, we offer several recommendations to improve security tools support for diagnostic work during responses to security incidents. These recommendations include criteria for evaluating usability of security tools in complex scenarios. Further research is needed to expand and refine our understanding on how technology can best provide the required support to security practitioners when they respond to security incidents.

Acknowledgments

The authors would like to thank our participants for taking part in our study, members of the Laboratory for Education and Research in Secure Systems Engineering (LERSSE) for feedback on earlier versions of this paper, and the anonymous reviewers of our paper for their valuable comments. This work has been supported by the Canadian NSERC Strategic Partnership Program, Grant STPGP 322192-05.

References

J. Bailey, E. Kandogan, E. Haber, and P. Maglio (2007). Activity-based management of it service delivery. In CHIMIT ’07: Proceedings of Symposium on Computer human interaction for the management of information technology, pages 1–5.

D. Botta, R. Werlinger, A. Gagne, K. Beznosov, L. Iverson, S. Fels, and B. Fisher (2007a). Studying IT security professionals: Research design and lessons learned. CHI Workshop on Security User Studies: Methodologies and Best Practices.

D. Botta, R. Werlinger, A. Gagne, K. Beznosov, L. Iverson, S. Fels, and B. Fisher (2007b). Towards understanding IT security professionals and their tools. In Proceedings of Symposium On Usable Privacy and Security (SOUPS), pages 100–111.

E. Casey (2002). ‘Error Uncertainty and Loss in Digital Evidence’. International Journal of Digital Evidence, 1(2).

E. Casey (2005). ‘Case study: Network intrusion investigation – lessons in forensic preparation’. Digital Investigation, 2(4):254–260.

K. Charmaz (2006). Constructing Grounded Theory. SAGE publications.

S. Chiasson, P. C. van Oorschot, and R. Biddle (2007). ‘Even Experts Deserve Usable Security: Design guidelines for security management systems’. In SOUPS Workshop on Usable IT Security Management (USM), pp. 1–4.

K. Fisler, S. Krishnamurthi, L. A. Meyerovich, and M. C. Tschantz. (2005). ‘Verification and change-impact analysis of access-control policies’. In ICSE ’05: Proceedings of 27th International Conference on Software engineering, pp. 196–205.

A. Gagne, K. Muldner, and K. Beznosov. (2008). ‘Identifying Differences between Security and other IT Professionals: a Qualitative Analysis’. In Proceedings of HAISA’08: Human Aspects of Information Security and Assurance, pp. 69–80.

S. Gibson (2001). ‘The Strange Tale of the Denial of Service Attacks on GRC.com’, http://whitepapers.zdnet.co.uk/.

J. R. Goodall, W. G. Lutters, and A. Komlodi. (2004a). ‘I Know My Network: Collaboration and Expertise in Intrusion Detection’. In CSCW ’04: Proceedings of ACM Conference on Computer Supported Cooperative Work (CSCW), pp. 342–345.

J. R. Goodall, W. G. Lutters, and A. Komlodi. (2004b). ‘The work of intrusion detection: Rethinking the role of security analysts’. In Proceedings of Americas Conference on Information Systems (AMCIS), pp. 1421–1427.

C. A. Halverson, T. Erickson, and M. S. Ackerman (2004). ‘Behind the help desk: evolution of a knowledge management system in a large organization’. In CSCW ’04: Proceedings of ACM Conference on Computer Supported Cooperative Work (CSCW), pp. 304–313.

K. Hawkey, D. Botta, R. Werlinger, K. Muldner, A. Gagne, and K, Beznosov (2008). ‘Human, Organizational, and Technological Factors of IT Security.’ In Ext. Abstracts of ACM Conf. on Human Factors in Computing Systems (CHI 2008), pp. 3639-3644.

E. Kandogan & E. M. Haber (2005). ‘Security Administration Tools and Practices’. In L. F. Cranor & S. Garfinkel (eds.), ‘Security and Usability: Designing Secure Systems that People Can Use’, pp. 357–378.

G. Killcrece, K. Kossakowski, R. Ruefle, and M. Zajicek. (2003). ‘State of the practice of computer security incident response teams (CSIRTs)’, www.cert.org/archive/pdf/03tr001.pdf.

G. Killcrece, K. Kossakowski, R. Ruefle, and M. Zajicek. (2005). ‘Incident Management’. Tech. rep., U.S. Department of Homeland Security.

Mitropoulos, Patsos, and Douligeris] S. Mitropoulos, D. Patsos, and C. Douligeris (2006). ‘On Incident Handling and Response: A state of the Art approach’. Computers and Security 25(5):351–370.

J. E. Orr (1986). ‘Narratives at work: story telling as cooperative diagnostic activity’. In CSCW ’86: Proceedings of ACM Conference on Computer-Supported Cooperative Work (CSCW), pp. 62–72.

M. Polanyi (1966). The Tacit Dimension. Doubleday & Company, Inc., New York.

J. Redish (2007). ‘Expanding Usability Testing to Evaluate Complex Systems’. Journal of Usability Studies 2(3):102–111.

J. Riden (2006). ‘Responding to Security incidents on a large academic network’. www.infosecwriters.com/text_resources/.

M. Sandelowski (2000). ‘Whatever Happened to Qualitative Description?’ Research in Nursing & Health, 23(4):334–340.

E. E. Schultz (2007). ‘Computer forensics challenges in responding to incidents in real life setting’. Computer Fraud & Security, 12:12–16.

E. H. Spafford (2003). ‘A Failure to Learn from the Past’. In Annual Computer Security Applications Conference (ACSAC), pp. 217–233.

P. Stephenson (2004). ‘The application of formal methods to root cause analysis of digital incidents’. International Journal of Digital Evidence, 3(1).

R. S. Thompson, E. Rantanen, and W. Yurcik (2006). ‘Network intrusion detection cognitive task analysis: Textual and visual tool usage and recommendations’. In Proceedings of Human Factors and Ergonomics Society Annual Meeting (HFES), pp. 669–673.

R. Werlinger, K. Hawkey, K. Muldner, P. Jaferian, and K. Beznosov (2008). ‘The challenges of using an intrusion detection system: Is it worth the effort?’ In Proceedings of Symposium On Usable Privacy and Security (SOUPS), pages 107–116.

R. Werlinger, K. Hawkey, D. Botta and K. Beznosov (2009). ‘Security Practitioners in Context: Their Activities and Interactions with Other Stakeholders Within Organizations’. International Journal of Human Computer Studies, 67(7): 584-606.

Y. Yamauchi, J. Whalen, and D. G. Bobrow (2003). ‘Information use of service technicians in difficult cases’. In CHI ’03: Proceedings of Human Factors in Computing Systems, pp. 81–88.