

NETWORK INTRUSION DETECTION COGNITIVE TASK ANALYSIS: TEXTUAL AND VISUAL TOOL USAGE AND RECOMMENDATIONS

Ramona Su Thompson, Esa M. Rantanen and William Yurcik
University of Illinois at Urbana-Champaign

A task analysis is conducted for the complex task of network security engineers, intrusion detection (ID) of computer networks. ID helps engineers protect network from harmful attacks and can be broken down into the following phases: pre-processing information, monitoring the network, analyzing attacks, and responding to attacks. Different cognitive loads are placed on the engineer at each phase. Engineers also need to integrate information from a variety of tools and resources, which adds additional cognitive workload. Visualization tools have been developed to alleviate these workloads but they have had limited success. To address this problem, we make two recommendations: (1) these tools should be designed for use across the phases of ID; this reduces the number of resources used therefore reducing the workload of integrating information across sources, and (2) visualization tools should allow concurrent use of textual tools and resources that provide detailed information and a powerful interface.

INTRODUCTION

Network security engineers (a.k.a. system or security administrators) provide many services for companies, research laboratories, and universities protecting their internal computer networks from malicious attacks. These services may be classified as reactive, proactive, or security quality management (Killcrece, Kossakowski, Ruefle, & Zajicek, 2003). Reactive services, such as incident handling and attending to alerts, are required when an event or request that might indicate an attack has occurred on the network (Killcrece et al., 2003; Yurcik, Barlow, & Rosendale, 2003). Proactive services, for example installing and configuring protective software on the networks and disseminating security-related information to the users of the network, provide assistance and information to help prepare, protect, and secure constituent systems in anticipation of future attacks, problems, or events (Killcrece et al., 2003; Yurcik et al., 2003). Security quality management services augment existing and already well-established services that are independent of incident handling (Killcrece et al., 2003). Traditionally, these services have been performed by other areas of the organization and not by network security engineers (Killcrece et al., 2003).

In this paper, we focus on the specific task of incident analysis or intrusion detection (ID) in the reactive services. This is the core task for network security engineers. In ID, engineers need to monitor the network for attacks, analyze potential attacks, and respond to them. While automated intrusion detection systems (IDS) and firewall applications provide substantial support in this task, successful ID still relies heavily on the expertise and knowledge of human engineers (Goodall, Lutters, & Komlodi, 2004a). Network security engineers are able to adapt to changes in their environment and in the internal network and quickly integrate pertinent information from a variety of sources about the network for a given attack.

Despite these strengths, ID remains a very difficult and challenging task. Ethnographic studies of engineers have shown that they frequently experience information and cogni-

tive overload while using multiple information resources, most of which are in textual form (Goodall, Lutters, & Komlodi, 2004b; Goodall, Ozok, Lutters, Rheingans, & Komlodi, 2005; Kandogan & Haber, 2005; Komlodi, Goodall, & Lutters, 2004; Yurcik et al., 2003). The complexity of the task coupled with the cognitively intensive processing of textual information can be overwhelming for engineers. Research on visualization tools has sought reduce this workload (Ball, Fink, & North, 2004; D'Amico & Kocka, 2005; Goodall, Lutters, Rheingans, & Komlodi, 2005; Goodall, Ozok et al., 2005; Komlodi et al., 2004; Lakkaraju, Bearavolu, & Yurcik, 2003; Yin, Yurcik, Li, Lakkaraju, & Abad, 2004), but few solutions have been successfully adopted by network security engineers.

We report results from a cognitive task analysis conducted through literature reviews and interviews with network security engineers at the National Center for Supercomputing Applications (NCSA) at the University of Illinois. We identify how the ID task as well as tools and resources used to perform the task cause cognitive overload. We then make recommendations of how visualization tools can be successfully integrated into ID to reduce this workload.

COGNITIVE TASK ANALYSIS

Intrusion detection is a very intensive task that poses high cognitive demands on network security engineers (Kandogan & Haber, 2005). Given the complexity of the task and the wide range of resources available to them, there are no detailed guidelines of how the task should be performed. Different institutions and individuals approach ID task differently, with their own set of tools, resources, and skills sets available to them, shaping an individual's approach to ID. Even within organizations, engineers may have diverse ways of carrying out ID. Despite the variety of approaches, we found common themes and similarities in how the task is performed by experts, reported in the literature as well as through interviews with network security engineers at the National Center for Supercomputing Applications at the University of Illinois.

The literature provided insights into the diverse roles of network security engineers, which provides for better design

of visualization tools for their needs. It also provided an overview of the ID task and many common themes in how it is performed. To understand the details of this task, we conducted in-house interviews with two network security engineers. Based on the findings in the literature, we asked further questions on how they conducted the ID tasks. The network security engineers were also asked to elaborate on the detailed process of ID and to ‘talk through’ an intrusion that needed to be processed.

Intrusion Detection Task: An Overview

From numerous interviews with network security engineers, Goodall et al. (2004b) classified the task of intrusion detection into three main phases: monitoring, analysis, and response. In the monitoring phase, engineers monitor a variety of systems and resources for suspicious activity that might indicate an intrusion. The analysis phase occurs when an IDS alert or other trigger event signals a potential intrusion on the network. When a real attack has been identified, the task moves into the response phase. The type of response depends on the attack. For example, the engineer might ask the owners of infected computers to clean their system using an anti-virus software or reconfigure firewalls, remove computers from the network, or, in severe cases, shutting down the entire network. This limited set of solutions in the response phase does not pose a high cognitive workload on the engineer.

Our interviews provided new insights to the various stages of intrusion detection not indicated by Goodall et al. (2004b). In the monitoring phase, we found that IDS, which generate alerts indicating potential intrusions, are key resources in the ID task. Unfortunately, the number of alerts that are triggered by the IDS can be overwhelming, especially with large heterogeneous networks, with a false alarm rate up to 99% (Julisch & Dacier, 2002). This adds to cognitive workload, as engineers must reduce the number of alerts to a manageable level while ensuring that true intrusions are not missed. Specifically, the monitoring phase requires allocation of attentional resources to monitor alerts and retrieve expert knowledge about the network to identify alerts indicating actual attacks.

Because IDS play such a key role early in the intrusion detection task, we found that active management of the IDS, or any alert-generating system, constitutes a substantial proportion of monitoring. This unique task can be viewed as a pre-processing phase before the monitoring phase. The management of these alerting systems requires the system to be updated based on knowledge of previous and recent attacks, expertise and knowledge of the network, and downloadable files provided by specific IDS software. This allows for better protection of the network and better filtering mechanisms for alerts. However, engineers can be overloaded with the amount of information to be integrated to maintain and update IDS at this phase.

Furthermore, our interviews revealed that the analysis phase can be further divided into two subtasks: (1) determining the cause of the alert and (2) deciding whether further investigation is warranted. Engineers use the alert message along with other resources to determine the cause and context

in which the alert was generated. The context information can provide clues to whether the alert is true. Gathering such clues is key to the decision subtask. Making a decision on an alert is one of the most difficult and cognitively challenging tasks in ID. Engineers need to rely on retaining pertinent information about the alert in working memory, retrieving their expert knowledge from long term memory, and integrating these pieces of information to make a decision on the ‘trueness’ of an alert.

Pre-processing, Monitoring, Analysis: A Detailed Look

Both the literature and our interviews indicate that the monitoring and analysis phases are the most cognitively intensive parts of intrusion detection. Network security engineers need situation awareness, experience, and background knowledge to identify a true attack on the network. In this section, we provide a more detailed look into the stages of intrusion detection as indicated by Goodall et al. (2004b) and the additional insights indicated in the previous overview section.

Pre-processing. In the pre-processing phase, the engineer needs to configure the system that generates alerts. This is crucial in intrusion detection as the alerts provide a starting point for investigations. Because of this, engineers would like to minimize the number of false alarms without missing any true alerts. In order to do this, engineers complete the following subtasks:

(1) Creating alerts. Alerts are generated when the system matches a signature of an attack with the network data. This can be used using IDS or scripts written by network security engineers on network data (i.e. system logs, Netflows). In either case, engineers can control the content of the alert. For example, they can consider various data in addition to the alert message and use the alert content as a preliminary means of triaging the events. In many cases, engineers can dismiss an alert as a false alarm with the network data that is provided along with their background knowledge and expertise.

(2) Filtering alerts. To reduce the amount of alerts, network security engineers create basic scripts and/or configure the IDS to filter the alerts. Engineers typically tailor their filters based upon the specific behaviors and intricacies of the internal network. For example, an engineer might write a filter to exclude the IDS alerts that are generated due to excessive connections being made for web servers. By using filters, the amount of alerts that are generated is reduced, thus reducing the amount of alerts that must be examined, and potentially reducing the percentage of false alarms.

The pre-processing phase does not need to be completed for all intrusion detection tasks. However, information of new attack signatures, signature updates for the IDS, intrusions, and experience with the network might initiate this phase. While this can be cognitively taxing, the resulting detailed and filtered alerts can make the use of engineers’ time more efficient by reducing the amount of alerts that are seen.

Monitoring and analysis. From our interviews, we found that it is difficult to make the distinction between the monitoring and analysis phases as defined by Goodall et al. (2004b). Specifically, the interviewed engineers indicated they simultaneously monitored and analyzed the alerts to decide whether

further investigation is needed. Because of this, we redefine the monitoring and analysis phases in more detail. For the monitoring phase, the main task is detecting a potential “true” alert. Specifically, engineers sift through the alerts that are generated and make initial judgments on the “trueness” of the alert by prioritizing the alert. This can be a very arduous and tedious task. In a given day, the network security engineers we interviewed stated that only 0.1% to 0.5% of the alerts signifying an actual incident. The analysis phase has been redefined into the following subtasks:

Determining the cause of the alert. Network security engineers typically look at the cause of the alert by looking at the alert message generated by the IDS and/or any of the resources that provide more detail about the alert. Finding the cause of the alert enables network security engineers to understand the context in which the alert is generated, providing the engineer with more clues of whether the alert is true.

Deciding further investigation is necessary. This is one of the most difficult steps in intrusion detection. Engineers must use their expertise and knowledge and their current knowledge of the attack to form a decision on whether further investigation is needed. The initial assessment in the monitoring phase and the determination of the cause of the alert form their initial decision. In some cases, this information is enough to warrant a confident decision. In other cases, engineers need to gather more information about the alert. In these cases, the decision subtask can be divided into three consecutive subtasks: (1) investigate and identify the individual computers in the network; (2) create hypotheses about the activity between the computers (legitimate action or an attack); and (3) investigate the hypotheses based on past experience and current information on network intrusions available.

Monitoring and Analysis: An Example

To illustrate these subtasks in detail, the network security engineer we interviewed walked us through an ID task. Because the engineer had already configured the IDS and filters of alerts for his specific needs, we do not include the pre-processing phase and the subsequent subtasks.

The first task was (1) to *detect a potential true alert*: The network security engineer was alerted to this alert from a message in their e-mail inbox. He had configured their filter so that he would receive high priority alerts in his e-mail inbox. Next the engineer had to (2) *determine the cause of the alert*. The IDS alert message was written in the body of the e-mail stating that machine A was being port scanned X times by machine B; both A and B were IP addresses. The next task, (3) *deciding further investigation is necessary*, actually involved several additional steps:

(3a) *Investigate the machines*: by simply looking at the IP address, our engineer was able to determine that machine A was a local machine and was scanning machine B. He looked up information online of machine B and was able to determine that it was a university machine used to run Condor jobs. He then investigated the local machine A (i.e. what group the machine belonged to, the type of machine, and the OS running on the machine). He found that the group using machine A uses Condor, too. (3b) *Create a hypothesis*: from the information

gained about machines A and B, the engineer was beginning to suspect that this was not a port scan, or an actual intrusion, but rather a gridFTP. (3c) *Test the hypothesis*: from experience, the engineer knew that the characteristics of a port scan are a high packet count, low byte count, and different flags set because of different TCP states. The characteristics of a gridFTP are a high packet count, high byte count, and a continuous TCP session. The engineer looked into the Netflow data between machines A and B and found the network traffic matched the characteristics of the gridFTP and not a port scan. Although he was confident, the engineer continued to verify his hypothesis by looking at the flags in the Netflow data: the alert was a false alarm and there was no intrusion associated with the alert.

RESOURCES AND TOOLS

Network security engineers have many resources and tools at their disposal, each of which is vital in the task of ID but only provide a limited scope of information (Yurcik et al., 2003). Hence, information integration is left to the engineers, who use both extensive background knowledge and numerous different textual tools and resources to accomplish this task. All of the above contribute to the cognitive complexity of the engineers’ jobs.

Background Knowledge

To be effective in the task of intrusion detection, network security engineers need to be armed with vast amounts of background information. This information serves as a baseline of network activity that is ‘normal’ and a standard for comparison if an intrusion is suspected. Our literature review and interviews imply that the two key sources of information are (1) the internal network, including the environment and (2) knowledge about the user base (Goodall et al., 2004b). Information about the internal network and the environment include subnet allocations, locations of critical assets on the network, network diagrams, allocated machines, etc. User-base information includes list of the user base, primary objective of the user base, understanding how other people use your network. Such knowledge is critical at each phase of ID: configuring the IDS (Goodall et al., 2004a, 2004b) in the pre-processing phase, determining what alerts to monitor closely in the monitoring phase, distinguishing between true and false alerts in the analysis phase, and determining the best course of action in the response phase.

Engineers’ experience helps them to organize, prioritize, and retain critical information about the environment, apply their knowledge, and continually gain deeper insight and adjust their understanding about the network. This knowledge comes at a price, as retrieval of this information can be difficult, especially with time-critical tasks as with ID.

Textual Resources

Many of the tools available to network security engineers present specific pieces of information in textual form

Research by Komlodi et al. (2004) and Goodall et al. have investigated the requirements of visualization tools for the monitoring and analysis phases. For monitoring, the tools should support pattern and anomaly recognition and provide an overview of the network and alerts (Goodall, 2005). These tools require little interaction and focused attention from the users (Komlodi et al., 2004) but rather alert users to areas that need attention, support pattern and anomaly recognition, and provide flexibility and customizability by the user (Komlodi et al., 2004). For analysis, researchers suggest tools that present data from multiple points of view (Goodall, 2005). These tools can aid engineers in identifying attacks while understanding the context in which they occur.

RECOMMENDATIONS

Two key issues that have not been addressed in the visualization usability literature are (1) potential overload of resources and (2) concurrent use of textual data. Visualization tools have merely added to the plethora of existing resources that engineers need to search and sift through each day. Many visualization tools only provide a limited scope of information but much functionality that can be easily overlooked. To address this problem, we recommend creating simple and tightly integrated visualization tools that incorporate a variety of resources to be used across subtasks. For example, the map interface might be used in the pre-processing phase, to help gain understanding and expertise, as well as in monitoring by providing an overview of where alerts are generated. This shifts tools from very specific functionality and uses to more general usability. Visualization tools with general functionality can also serve as ‘organizers’ for traditional textual tools.

Our research suggests that network security engineers will continue to use the textual resources despite advances in data visualization. Textual resources are often rich with detailed information critical to understanding the context of the attack, whereas visualization tools tend to present an overview of the data. Even in the cases where the details of the resources are provided in the visualization tool, the engineers would still regard their use difficult and inefficient. Many tools require much interaction with the interface to navigate through detailed information. Familiarity with textual tools, however, allows engineers to quickly navigate and filter through data to find desired information, providing more flexibility, efficiency, and customizability than current visualization tools. Our recommendation is hence to integrate visualization tools with current-use textual tools, specifically by combining the ‘shell’ window of the textual tool—commonly used to filter through textual data—with visualization tools and integrating the functionality between them. This way engineers can use visualization for an overview and pattern identification but have detailed textual data available to them at all times.

ACKNOWLEDGMENTS

This research was supported in part by a grant from the National Center for Advanced Secure Systems Research (N00014-03-1-0765). The technical monitor was Ralph Wachter, Office of Naval Research. The views presented in this paper are those of the authors and do not necessarily represent official National Center for Advanced Secure Systems Research positions.

REFERENCES

- Ball, R., Fink, G. A., & North, C. (2004, Oct. 29). Home-centric visualization of network traffic for security administration. *VizSEC/DMSEC*, 55-64.
- D'Amico, A., & Kocka, M. (2005, Oct. 26). Information assurance visualizations for specific stages of situational awareness and intended uses: Lessons learned. *VizSEC*, 107-112.
- Goodall, J. R. (2005, June 15-17). User requirements and design of a visualization for intrusion detection analysis. *IEEE Workshop on Information Assurance and Security*, 394-401.
- Goodall, J., Lutters, W., & Komlodi, A. (2004a). I know my network: collaboration and expertise in intrusion detection. *CSCW*, 342-345.
- Goodall, J., Lutters, W., & Komlodi, A. (2004b, Aug 6-8). The work of intrusion detection: Rethinking the role of security analysts. *AMCIS*, 1421-1427.
- Goodall, J. R., Lutters, W. G., Rheingans, P., & Komlodi, A. (2005, Oct 26). Preserving the big picture: Visual network traffic analysis with TNV. *VizSEC*, 47-54.
- Goodall, J. R., Ozok, A. A., Lutters, W. G., Rheingans, P., & Komlodi, A. (2005, Apr 2-7). A user-centered approach to visualizing network traffic for intrusion detection. *Extended Abstracts CHI*, 1403-1406.
- Julisch, K., & Dacier, M. (2002, Jul 23-26). Mining intrusion detection alarms for actionable knowledge. *KDD*, 366-375.
- Kandogan, E., & Haber, E. (2005). Security administration tools and practices. In L. Cranon & S. Garfinkel (Eds.), *Security and Usability: Designing Secure Systems that People Can Use* (pp. 357-376). Beijing: O'Reilly.
- Killcrece, G., Kossakowski, K., Ruefle, R., & Zajicek, M. (2003). *State of the Practice of Computer Security Response Teams (CSIRTs)* (No. CMU/SEI-2003-TR-001): Carnegie Mellon Software Engineering Institute (SEI).
- Komlodi, A., Goodall, J. R., & Lutters, W. G. (2004, Apr 24-29). An information visualization framework for intrusion detection. *Ext. Abstr. CHI*, 1743.
- Lakkaraju, K., Bearavolu, R., & Yurcik, W. (2003). NVisionIP - A traffic visualization tool for security analysis of large and complex networks. *International Multiconference on Measurement, Modeling, and Evaluation of Computer-Communications Systems*.
- Yin, X., Yurcik, W., Li, Y., Lakkaraju, K., & Abad, C. (2004, Apr 15-17). VisFlowConnect: Providing security situational awareness by visualizing network traffic flows. *IPCCC*, 601-607.
- Yurcik, W., Barlow, J., & 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*.