To appear in *17th Annual Canadian Conference on Electrical and Computer Engineering (CCECE 2004)* at Niagra Falls, Canada (2 – 5 May 2004).

Improving Intrusion Detection Systems
Through Heuristic Evaluation

Andrew T. Zhou James Blustein Nur Zincir-Heywood
Faculty of Computer Science, Dalhousie University
{azhou, jamie, zincir}@cs.dal.ca

Abstract

This work is a report on efforts to improve the usability of intrusion detection systems. Specifically, we first conducted a worldwide survey of system administrators from different countries and economic sectors to understand the state of practice in security management with a particular focus on intrusion detection systems (IDSs). Then, based on these survey results and in depth interviews, we developed new heuristics to measure the effectiveness and efficiency of IDSs. The comparison of our refined heuristics and Nielsen's general heuristics on Snort, Snortsnarf and our proposed interface show that evaluators using our heuristics find significantly ($p<0.0002$) *more of the problems. Also, evaluations with both sets find fewer problems in our interface than in Snort or Snortsnarf.*

Keywords: *Human-Machine Interactions, Usability Engineering, Network Intrusion Detection*

1. Introduction

We loosely define intrusion detection systems (IDS) as a security management tool of computer network administrators who monitor systems/networks to detect inappropriate accesses. We contend that there are two main problems regarding the state of the art and the state of practice in intrusion detection systems: the underlying technique in detecting attacks, and the human interface to enable administrators to quickly and accurately detect and respond to attacks [1, 2]. Significant improvements can, in theory, be made to IDS by implementing better detection capabilities. However, experience has shown that even advanced technical solutions can fail when their user interfaces are not adapted to their users [14]. The importance of good interface is particularly important in real-time and security applications where users are likely to be stressed and errors can have serious consequences [4].

The results of a survey into the state of the practice in security management in a variety of companies and institutions worldwide clearly show that the state of the network security management is poor [16]. The situation is partly because of the lack of good tools for administrators. As well, IDS applications usually require more time than many network administrators have available to devote to their proper use. Some institutions, particularly in the military, hire full-time system administrators whose sole task is to monitor IDS. This is not an option for most companies. There is a need for improved IDS. One way to improve them is to supply more usable interfaces that will enable more efficient and effective use, and ultimately greater security.

We believe that evaluating applications is a step towards improving them. Heuristic evaluation is a very popular discount usability inspection method [9]. Some authors consider heuristics and guidelines to be identical [7]. Heuristic evaluations can detect up to 60% of the usability problems that an empirical user test would find [1, p.412]. However, before now there has not been a set of heuristics that is specifically created to evaluate security related applications. Our objective at this stage of the project is to generate heuristics of evaluating usability for this specific problem domain. Such methods are used to assess the quality of existing products and to identify needs that can be fulfilled by products.

In order to conduct the heuristic evaluation, we chose Snort as our candidate application. Snort is a simple but popular IDS. It is able to log and analyze traffic on IP networks. Because it is a command-line based application, we picked a Web-based application —– SnortSnarf — a user-interface front-end from Silicon Defense [12].

The remainder of the paper is organized as follows. First we review some previous work about using and developing heuristics. Then we discuss specifics of our methodology and the heuristics we propose. We also report on our proposed evaluation method. Finally, we draw conclusions and show directions for future work.

2. Method

2.1. Issues to address

How can we tell if software is well-designed? The major difference between evaluating IDS and evaluating applications of other domains derives from the types of users and the tasks those users need to undertake. General heuristics [9] can be applied to any software but they are focused on systems with clearly defined user tasks [8] whereas IDS users rarely perform well-defined tasks or have much time to choose a course of action.

An example of this is that most of the time, users of other applications can have clear objectives about what to do and where to go. On the other hand, IDS users often cannot accurately predict when, why, and how intrusions occur. A large part of the use of IDS is to quickly determine if an alert signal indicates an actual attack or is just a false alarm. When there are actual attacks, users must also plan and carry out responses. There is no certain routine to follow when facing incidents. IDS users must dynamically form and execute plans without complete information.

Therefore, we believe that, if there is not a need to develop specialized heuristics for IDS then, there is at least an opportunity to adapt the general heuristics for systems like IDS.

2.2. Development of the IDS Heuristics

Baker et al. [2] and Mankoff et al. [8] have recently developed specialized heuristics for groupware and ambient displays respectively. We use the same methodology in developing heuristics for IDS: First we determined the primary goals of IDS. Next we modified the general heuristics to better suit what we know about the target user group (network security administrators) and the tasks they must perform with IDS. We checked that each heuristic reflected at least one issue that we detected in early stages and that no issues were left out. Finally we evaluated the new heuristics by comparing how successful they are at detecting problems in IDS applications. The final step was achieved by comparing against a set of general heuristics [9]. The details of our methodology follow.

Before defining a new set of proposed heuristics, we reviewed the characteristics of security management. Based on which we generated questionnaires for an online survey [16] of the state of the practice in security area. From the survey and anonymous interviews with several security administrators at multiple sites, we made a list of problems faced while using security management tools. Furthermore, we suggested solutions to those problems, and combined them with necessary and desirable (i.e. welcome but not strictly necessary) features from the survey results to form a checklist based on which we generated specific heuristics in six categories (see Table 2-1). Overlap between the general heuristics and ours is indicated in Tables 2-1 and 2-2 by shared superscripts.

In order to discover how effective IDS heuristics are in identifying usability problems, one of us evaluated the popular Snort application by using both the general and IDS heuristics. The results (see Tables 2-1 and 2-2 for details) indicate that IDS heuristics discovered 61 problems and general heuristics discovered 58 usability problems, respectively.

Except for the common part of both heuristic sets, IDS heuristics identified 14 violations in information navigation, which is typically a major feature of IDS applications. The general heuristics do not particularly address this issue. Information navigation may not be critical as in some domains as it is in intrusion detection. On the other hand, it is not obvious that problems like ‘Aesthetic and minimalist design’ are pressing issues in IDS applications (although they may be related to trust engendered by perceived reliability [3, 13]). Snort does not have operations that may cause problems related to these factors.

IDS Heuristic	Violations
Visibility of system and IDS status[1]	8
Consistency and standards[2]	8
Display of information	9
Information navigation	14
Flexibility and efficiency of use[3]	16
Help and documentation[4]	6
Total	*61*

Table 2-1: *IDS Heuristics Applied to Snort*

General Heuristics	Violations
Visibility of system status[1]	8
Match between system and the real world	3
User control and freedom	10
Consistency and standards[2]	8
Error Prevention	2
Recognition rather than recall	3
Flexibility and efficiency of use[3]	16
Aesthetic and minimalist design	0
Help users recognize, diagnose, and recover from errors	2
Help and documentation[4]	6
Total	*58*

Table 2-2: *General Heuristics Applied to Snort*

2.3. Validation method

Our heuristics are specifically designed to serve IDS usability evaluations. We believed that they would serve an important rôle for the design of IDS application. Although the comparison results of Snort evaluation indicated that IDS heuristics were capable of identifying usability problems, we wanted to receive feedbacks through a formal process. In actual practice, any heuristic set is applied by multiple unbiased evaluators and the results considered in aggregate. Therefore, we carried out an experiment to evaluate our IDS heuristics.

Participants There have been different views on what number of users is necessary to conduct an effective evaluation [11,15]. To be cautious we recruited 12 participants to use the heuristic sets to evaluate the interfaces. Because the rôles of expertise in the application domain and human factors are not entirely clear, we ensured that 5 of our evaluators were HCI experts and the others were network security experts.

Task Participants were asked to evaluate two web applications — SnortSnarf and SnortReader — using both the general and our IDS heuristic sets. (SnortSnarf is an interface application to Snort developed by Silicon Defense [12]; SnortReader is also an interface application to Snort that we created as an early attempt to implement solutions to

problems that we found in use of IDS application.) Every participant completed the evaluation individually, and was allowed to spend as much time as needed to finish the test. Before the evaluation, each one of them received a package through e-mail. The package included a brief introduction to our project and instructions on how to conduct the evaluation. Separately, we conducted a review using Nielsen & Molich's methodology [10] to generate a master list of issues and their severities: Severe issues — ratings of 4–5 on a five-point scale — were problems that we felt might substantively discourage users from using IDS; Moderate issues — ratings of 2–3 — were issues that might decrease the speed or accuracy of identifying intrusions; All other issues were considered minor (e.g. problems caused by the misunderstandings about how applications worked or restrictions imposed by the WWW interface.) We identified and rated 32 separate issues (see Table 3-1) for the distribution of ratings).

3. Results

Seventeen issues in SnortSnarf were identified by the general heuristics and 19 issues by our IDS heuristics. Both sets identified 10 problems in our SnortReader.

Severity of Problems Found	Count of Problems Found in IDS UIs		Detection Rate By Heuristic Set	
	SnortSnarf	SnortReader	General	IDS
1 (low)	1	8	89%	77%
2	8	5	77%	69%
3	4	—	75%	100%
4	6	—	67%	100%
5 (high)	—	—	—	—

Table 3-1: The number of known issues at every severity rating for both interfaces

Result 1: *Our specific heuristics found more of the problems than the general set did.*

Twelve participants identified 23 problems in total by using IDS heuristic set when 17 problems were discovered by using the general heuristics set. A one-tailed sign-test shows that this overall result is highly significant ($p<0.0002$, df=11). The improvement is especially noticeable with the moderately to highly severe problems.

Figure 3-1 shows the increase in percentage of known issues found as the number of evaluators increased. This increase is to be expected but it is not clear at what point the number of issues should be expected to level out.

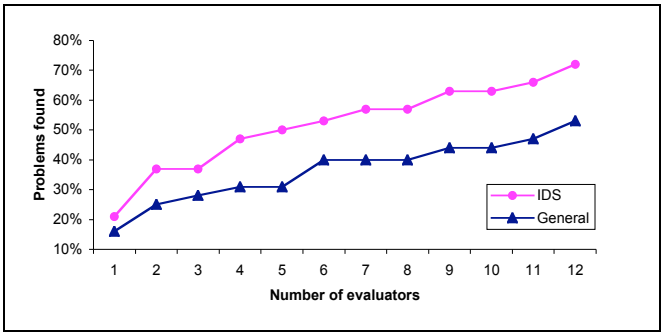

Figure 3-1: A comparison of percentage of issues found by evaluators of both interfaces using each set of heuristics

Result 2: *Evaluators found fewer problems in our novel interface than in SnortSnarf.*

Evaluators found 16 of 19 known problems with SnortSnarf and 11 of 13 with SnortReader. This is evidence, although not conclusive, that the interface we developed is better suited for use with Snort than SnortSnarf is. Figure 3-2 shows the increase in numbers of known issues found as the number of evaluators increases and the number of issues found by evaluators in the two interfaces.

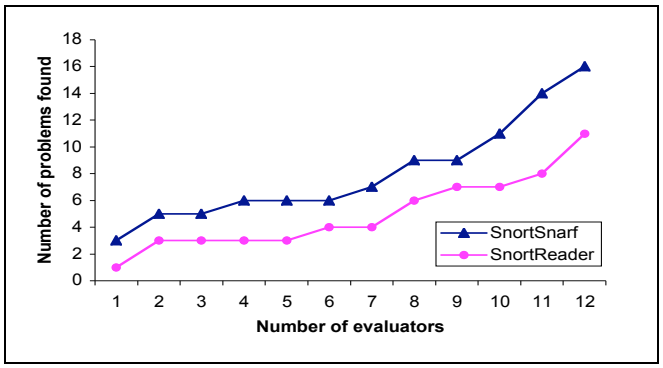

Figure 3-2: A comparison of problems found by evaluators in two interfaces

4. Conclusions

In this work, we developed a new set of IDS heuristics to improve security through better usability. We examined our heuristics using a formal method based on Nielsen and Molich's method and input from a survey on the state of the practice of security management. In our experiment, our heuristics identified significantly more usability problems in IDS than general heuristics did. Our approach to improve IDS usability is shown to be effective.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canadian Foundation for Innovation (CFI), the cooperation of Thor Solutions, Inc. and the Telecom Applications Research Alliance (TARA).

REFERENCES

[1] Stephen D. Amstrong, William C. Brewer, and Richard K. Steinberg. Usability testing. In Samuel G. Charlton & Thomas G. O'Brien, (eds), *Handbook of Human Factors Testing and Evaluation*, Second edition, Chapter 18, pp. 403–432. Lawrence Erlbaum, ISBN 0-8058-3291-2. 2002.

[2] Kevin Baker, Saul Greenberg, and Carl Gutwin. Empirical development of a heuristic evaluation methodology for shared workspace groupware. *Proceedings of the 2002 ACM conference on Computer supported cooperative work*. New Orleans, LA, USA. pp. 96–105. <URL: http://doi.acm.org/10.1145/587078.587093>.

[3] Laurie Brady and Christine Phillips. Aesthetics and usability; a look at color and balance. *Usability News*, 5.1, <URL:http://psychology.wichita.edu/surl/usabilitynews/51/aesthetics.htm>, 2003. Accessed 2004-02-01.

[4] Andrew Dillon. Beyond usability: Process, outcome and affect in human computer interactions. *Canadian Journal of Information Science*, 24(1):57–69, 2001.

[5] Richard A. Kemmerer and Giovanni Vigna. Intruder detection: A brief history and overview. *IEEE Computer*, 35(4): 27–30, 2002.

[6] Richard A. Kemmerer and Giovanni. Vigna. Intrusion detection: A brief history and overview. *IEEE Security & Privacy*, 2002. <URL:http:// www.computer.org/security/supplement1/htm/>.

[7] Clayton Lewis & John Rieman. *Task-Centered User Interface Design*. <URL:ftp://ftp.cs.solorado.edu/pub/cs/distribs/clewis/HCI-Design-Book/>. Accessed 2004-02-01, 1994.

[8] Jennifer Mankoff, Anind K. Dey, Gary Hsieh, Julie Kientz, Scott Lederer, and Morgan Ames. In Victoria Bellotti, Thomas Erickson, Gilbert Cockton, and Panu Korhonen (eds.) *Proceedings of the conference on Human factors in computing systems* (ACM CHI 2003), pp. 169–176, 2003. <URL: http://doi.acm.org/10.1145/642611.642642>

[9] Jakob Nielsen. Heuristic evaluation. [webpage]. <URL:http://www.useit.com/papers/heuristic/heuristic_list.html>, 2003. Accessed 2004-02-01.

[10] Jakob Nielsen and, Rolf Molich, Heuristic evaluation of user interfaces, *Proceedings of the SIGCHI conference on Human factors in computing systems: Empowering people*, pp.249-256, April 01-05, 1990, Seattle, WA. <URL: http://doi.acm.org/10.1145/97243.97281>.

[11] Jakob Nielsen and T. K. Landauer. A mathematical model of the finding of usability problems. *Proc. ACM INTERCHI*, pp. 206–213, 1993. <URL: http://doi.acm.org/10.1145/169059.169166>.

[12] SnortSnarf. Silicon Defense. <URL:http://www.silicondefense.com/software/snortsnarf>.

[13] C. W. Turner. The online experience and consumer's Perceptions of E-commerce Security. *Proc. Human Factors and Ergonomics Society*, pp.1246–1250, 2002.

[14] A. Whitten and J. D. Tygar, Why Johnny Can't Encrypt: A Usability Case Study of PGP 5.0. *Proc. USENIX Security*, 1999.

[15] A. Woolrych and G. Cockton. Why and when five test users aren't enough. *Proc. IHM-HCI 2001*, vol. 2, pp.105–108, 2001.

[16] Andrew Zhou, James Blustein, and Nur Zincir-Heywood. The state of network security management: Issues and directions. Technical Report CS-2003-06. Dalhousie U. Faculty of Computer Science, 2003. <URL:http://www.cs.dal.ca/research/techreports/2003/CS-2003-06.html>.