Visual Analysis of Goal-Directed Network Defense Decisions

Chris Horn
Secure Decisions
a division of Applied Visions, Inc.
6 Bayview Ave
Northport, NY 11768
+1 631 759 3933
chris.horn@securedecisions.com

Anita D'Amico
Secure Decisions
a division of Applied Visions, Inc.
6 Bayview Ave
Northport, NY 11768
+1 631 759 3909
anita.damico@securedecisions.com

ABSTRACT
Security visualization has been focused largely on graphic representation of data and relationships between network activity, security sensor output, and attacker activity. Visual analysis tools have not been designed to facilitate the analysis of data related to defender activities and decisions. This paper reports on the initial effort of a research team to use visual analytics to support the modeling of the computer network defense (CND) decision process of an organization. We describe a tool to support the visual analysis of a hierarchical decision structure represented in a portable, file-based database. The tool visualizes and traces relationships between decision goals, sub-goals, decisions, information requirements, and data sources.

Categories and Subject Descriptors
D.2.1 [**Software Engineering**]: Requirements/Specifications – *Tools*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces – *Graphical user interfaces (GUI)*; J.7 [**Computer Applications**]: Computers in Other Systems; K.6.1 [**Computing Milieux**]: Project and People Management – *Systems analysis and design*

General Terms
Documentation, Security, Human Factors

Keywords
Visualization, visual analytics, information security, computer network defense (CND), decision model, decision aid, web-based tools, Protovis

1. INTRODUCTION
Security visualization has been focused largely on graphic representation of data and relationships between network activity [14, 17] security sensor output [2, 15], and attacker activity [6, 18, 22]. This reflects the preponderance of research on the detection, remediation, and prediction of activity by malicious actors. Far less research has been conducted on the activities and decision processes of network *defenders*.

The small but growing body of work on defender cognition and behavior [3, 5, 7, 10, 13, 16] often relies on meticulous observation of defenders while they are monitoring and analyzing network activity, in addition to interviews with defenders, their managers, and trainers. For security reasons, data collection is usually conducted offline, with hand-written notes and drawings that capture workflows and other information on large rolls of paper. Raw data collection is typically followed by transcription of notes into electronic form, and a manually-intensive extraction of key findings into Microsoft Excel spreadsheets and Microsoft Visio charts. These work products are then analyzed by the research team and again re-structured into forms for easier communication and collaboration among the research team and with the sponsor.

To date, there are no visual analysis tools designed to facilitate the analysis of data related to defender activities and decisions. As research on defender activities and decisions increases, so too will the need for visual tools to assist researchers in their analysis of defensive decisions, activities, and events.

This paper reports on the initial effort of a research team to use visual analytics to support the modeling of the computer network defense (CND) decision process of an organization.

Among the project's objectives was identification of: 1) CND decisions being made; 2) information needed to support these decisions; 3) data sources available to support the CND information requirements; 4) quality of data sources for supporting CND decisions; and 5) new technologies and data sources to support CND decision-making. The essence of our project was to provide a foundation for a security investment strategy: what technologies and enhanced data sources could potentially improve the CND decision capability?

To accomplish the project goals, we needed to build an understanding of how data is supplied to, and employed in, the decision making process. We adopted Endsley's Goal-Directed Task Analysis (GDTA) framework [1, 12] to structure the data collection and categorization. We related data sources to higher-level goals using a six layer model:

1. An overarching goal that orients operations (i.e., an "observed practice" mission statement)
2. Major goals that decision makers must achieve in pursuing the overarching goal
3. Sub-goals that incrementally contribute to accomplishing the respective major goal
4. CND decisions needed to accomplish the major goals and sub-goals
5. Situation Awareness (SA) information requirements needed to make decisions (the information needs of a decision)
6. Data sources that serve as the foundation for satisfying SA information requirements.

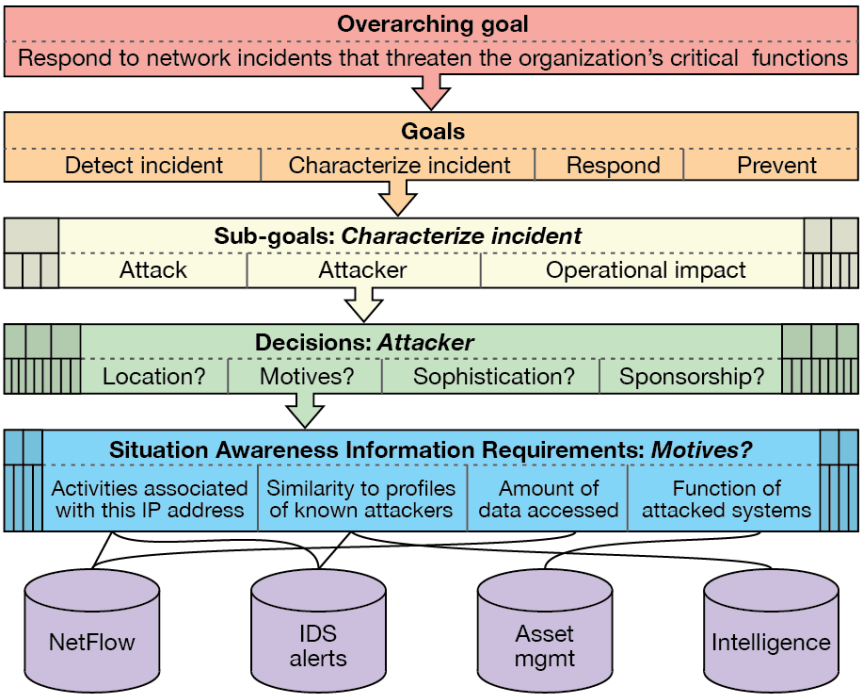

Figure 2. Sample slice through the six layer decision model

A sample slice through this six layer model, populated with notional information, is shown in Figure 2. At the top there is the one overarching goal that captures the mission of an organization from observations of its practices. This overarching goal can be decomposed into four goals that, here, represent all of the goals of an organization's work within the scope of analysis. Each of these goals is incrementally served by a set of sub-goals. The third row from the top shows the sub-goals for the "Characterize incident" goal; the sub-goals for the other goals are represented by the greyed-out boxes to the left and right. The remainder of the figure follows the same pattern, with the bottom-most row representing the data sources that are used to answer SA information requirements.

This paper addresses how we used visual analytics to structure the information contained in our model, analyze it, and communicate our findings.

2. RELATED WORK

The few published studies of work domain analyses and cognitive analyses of network defenders [3, 8, 13], and unpublished work we have had the opportunity to review or in which we participated, have relied on multi-page Visio-type drawings and extensive tables to capture and communicate findings. Cognitive task analyses in different, but equally complex, domains [11] use similar tabular and charting techniques. Endsley's GDTA has also been used to structure decisions across many domains [9]. Our inspection of related work did not yield any recommendations about the use of visual analytics in other GDTA studies.

3. MOTIVATION

Early in the project, our team was rapidly filling in the structure of the six-layered model. Data sources, SA information requirements, and CND decisions were being identified through a range of sources, including interviews with cyber defenders, reviews of standard operating procedure and policy documents, and observations of operations. At the same time, the team was working hard to synthesize and build internal consensus on the proper definition and organization of the higher-level goals under which decisions are grouped.

Faced with the challenge of collecting and sharing contributions from multiple team members, our team built a Microsoft Excel-based representation of the model. The spreadsheet was arranged as a horizontal tree; in the leftmost column A were major goals, column B contained sub-goals, column C contained decisions, and so on. Because major goals contain multiple sub-goals and sub-goals contain multiple decisions, the total number of rows devoted to a given goal is determined by how many data sources are employed by all of the SA information requirements for each decision, for each sub-goal under that goal. This simple fact meant that the spreadsheet quickly grew to 400+ rows.

Working to read and synthesize data on a spreadsheet of this size is difficult. With such a small viewport relative to the size of the entire worksheet, scrolling creates a large short term memory load. Ensuring that changes were applied consistently throughout the sheet was time consuming and error prone. For example, early in the process we went through several iterations of selecting the

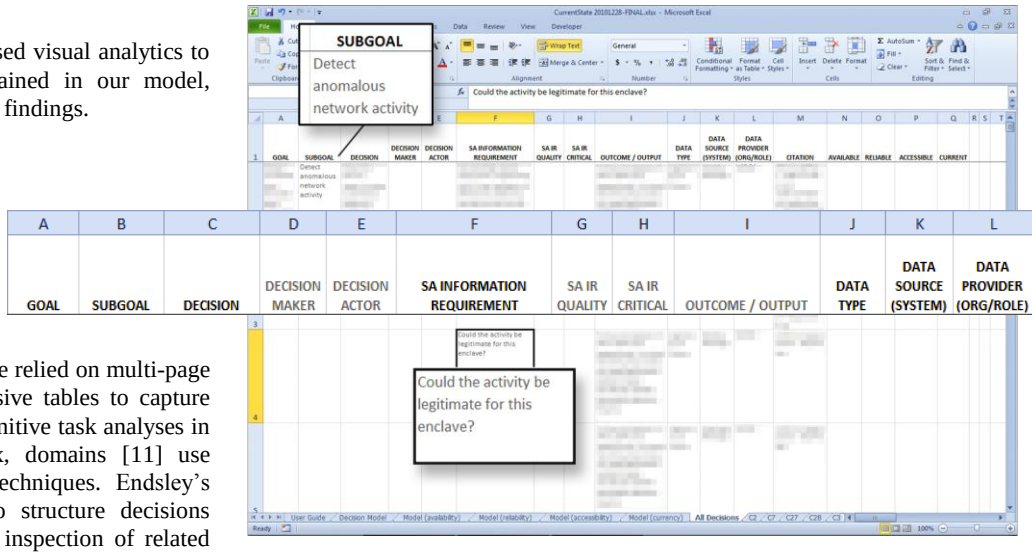

Figure 1. Excel spreadsheet used to record the decision model

best level of abstraction for representing data sources. As we redefined sources and changed their labels, the worksheet structure meant that someone had to search for all instances of a data source to change each of its entries.

Because team members often worked on the spreadsheet individually, there was a large need to be able to track and present changes. Just like with source code, changes included moves, edits, removals, and merges; sub-goals and their subordinate decisions were moved under a different goal, data sources were renamed or removed, and multiple decisions were merged into one. There was a need to not only be able to account for these changes, but to see them and present them back to the group for discussion and consensus.

In short, requirements emerged for a tool that would improve the readability, comprehensibility, and ease of communicating the substance of our decision model. We also needed an analytic capability to help us prioritize the contribution of data sources and system capabilities to the effective performance of CND; we needed to answer questions such as, "What are the most important decisions?", "What are the most common decisions?", and "On what data sources and systems do these decisions depend?".

To those who have worked collaboratively on similar types of projects, many of these challenges are familiar. In fact, prior to beginning work on the spreadsheet, we conducted a search for tools that could support the creation of a decision model. We considered entity-relationship and mind mapping software, business process modeling packages, and graph visualization tools – none offered significant advantages to a simple Excel-based approach. With the likely possibility that our sponsor would want a functional copy of the tool, however, we were forced to discount many of these tools due to the restrictive nature of the sponsor's computing environment; the tool could not require network access, administrator privileges, nor any supporting software (libraries, frameworks, etc.) that was not already part of the standard government desktop configuration. However, as the project progressed and we began trying to analyze the Excel-based decision model, the limitations of Excel forced us to reconsider our approach.

4. DESIGN & IMPLEMENTATION

After ruling out the feasibility of using Microsoft Office products to leverage our existing Excel file, we set out to build a completely standalone application that could operate in user space. With little budget for a large development effort, the team decided to employ the common web browser as a critical element of our tool. We were attracted to using web technology for its ability to support a graphical user interface with low implementation overhead.

There are three major components to our tool: a standalone web server, a file-based database, and web browser-based interfaces to edit, query, and visualize the data.

4.1 Standalone Web Server & Database

When we first started designing the tool, we thought that we may be able to write a tool that ran directly in the web browser. However, it quickly became apparent that it would be much easier and flexible to use the browser as it was designed: in conjunction with a web server.

In searching for a standalone web server, the team discovered that Python [20] comes bundled with a HTTP server module [17] that allows developers to create a simple web server with a few lines of code. Python also has modules that support working directly with SQLite [21], a library for creating self-contained, server-less,

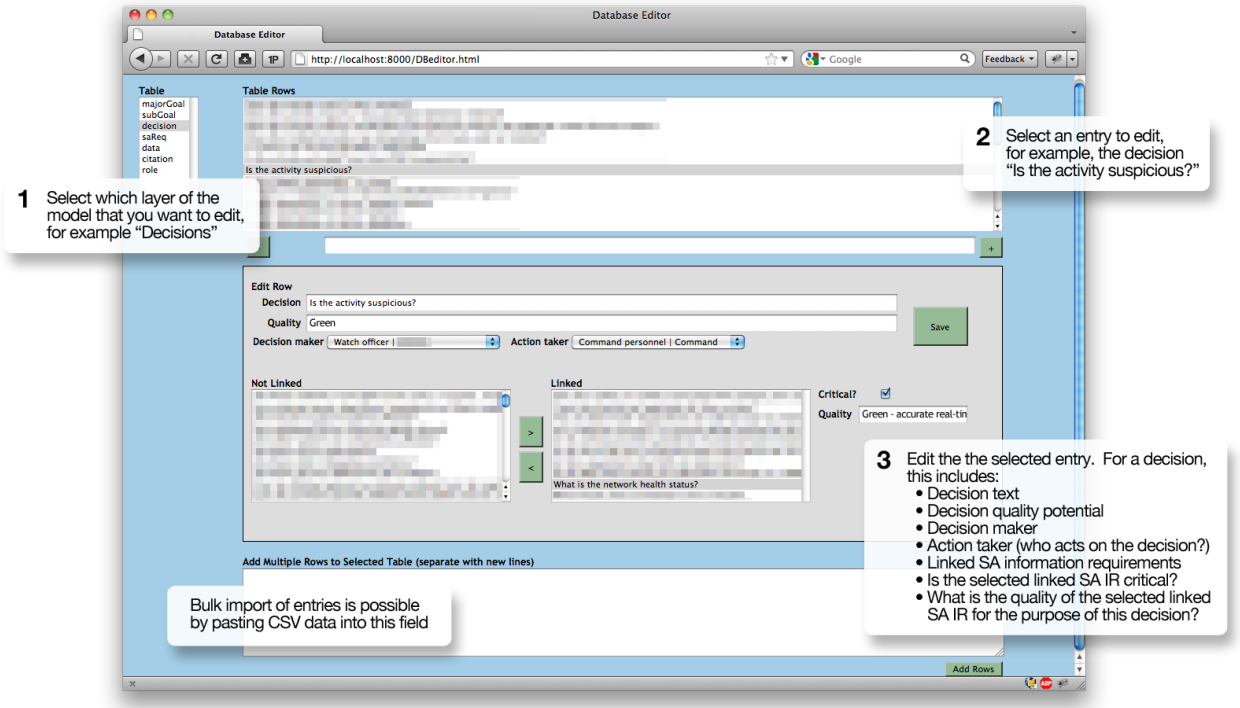

Figure 3. Web browser-based database editor

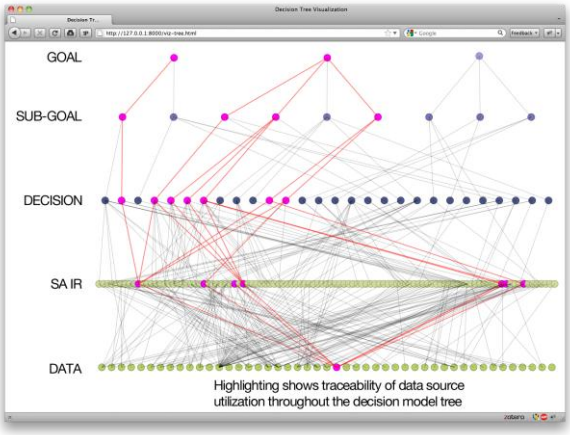

Figure 4. Interactive visualization of the decision model tree

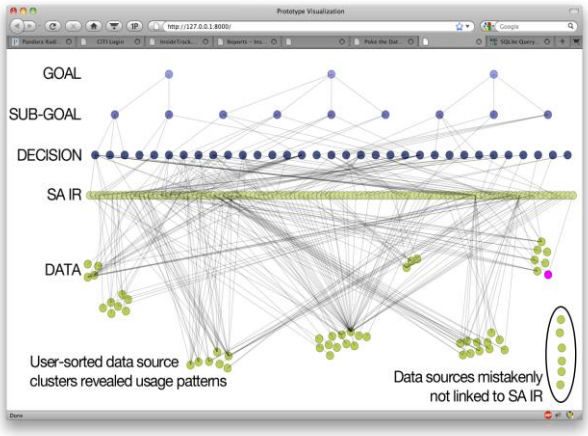

Figure 5. A user-sorted view of the decision model

relational databases. Furthermore, Python has been extended with software called py2exe [19] that allows its scripts to be converted into executable Windows programs.

By combining these technologies, we were able to rapidly create a lightweight web server and relational database back-end that runs as a standalone application from a folder residing anywhere on a user's computer. This design also supports basic collaboration between team members; if Alice wants to share her modifications to the decision model, she can simply email her database file to Bob, who can then work on it using his local copy of the tool.

4.2 Web Browser User Interface

Using a web server allows the rest of the tool to be written as a web application of server-side Python scripts and client-side HTML, CSS, and JavaScript. There are three distinct sections of the web browser-based user interface: a database editor, an interactive visualization, and a place to run SQL queries.

4.2.1 Database Editor

Users of our tool view and edit the decision model database through a single page user interface shown in Figure 3. The interface supports adding, removing, and editing items to each level of the hierarchy. Links between items, such as decisions and SA information requirements, are created by moving items between a "Not Linked" list and "Linked" list. For each link, there is often meta-data about that particular link that can be specified in the model; for example, an SA information requirement can be flagged as critical to its superordinate decision. This meta-data can be set by the user in the area immediately adjacent to where the selected linked item appears.

The bulk import of items was something that was very important early in the use of this tool, as we needed to translate our decision model from its Excel form to this tool's database. At the bottom of the screen, an input box allows users, in one step, to add multiple items to the selected level of the decision model.

4.2.2 Interactive Visualization

The primary motivation behind building this tool was a desire to visually explore our decision model. Our first sketch of a concept was a straightforward, tree representation of the decision model. We hoped to be able to gain insight into the question of "On what data sources and systems do these decisions depend?", as well as find patterns in the usage of data sources.

We implemented the tree-based concept using a custom layout in the Protovis [17] graphical toolkit. This layout defaults to the horizontally stratified graph seen in Figure 4. Along the top are major goals; the row below that contains sub-goals; the third row contains decisions, and so on. The pink highlighting appears when you click on a node in the graph; in this case, the data source in the bottom row was clicked, highlighting all of the SA information requirements (SA IR), decisions, sub-goals, and goals that incorporate information from that data source.

The highlighting was implemented to help users visually trace the linkages between each level. In this example, a type of report (data source) is seen to be important to a quarter of documented decisions and fully half of the CND sub-goals.

Users can hover their mouse cursor over a vertex, or node, to cause a "tool-tip" label containing that node's description to appear next to the selected item. While hiding these labels is not ideal for easy interpretation of the graph, in practice it is less of an impediment than one would expect and does serve to greatly reduce visual clutter.

The real power of this visualization lies in its interactivity. Figure 5 shows a view of the decision model after it was sorted by one of our team members. Here, we were trying to identify patterns in data source usage in the decision model. One specific goal we had in mind was to identify highly relied-upon data sources – under the theory that one type of high-yield investment would be to increase the efficiency or such sources. For example, if a technology improvement could save 20 minutes each time a data source was consulted, seeking the most highly used data sources would maximize the benefit of that investment.

In fact, we observed such a class of data sources after applying the manual sort shown in Figure 5. The center cluster of data sources was linked to virtually all SA information requirements. Despite their ubiquitous usage, the generic nature of these data sources meant that they had been somewhat overlooked as an investment target; this visualization fostered a group discussion and provided a strong picture of how much the CND decision process depended on them.

Luckily, due to the way data were imported into the tool, SA information requirements were mostly organized under their corresponding decisions, facilitating visual analysis.

The visual sorting employed by this user also revealed data quality problems. For example, the group of vertices on the far right of Figure 5 is data sources that were present in the database, but not linked to any information requirements. Identifying such data sources in the editor view is all-but-impossible; here they clearly stand out.

4.2.3 SQL Query Interface

In addition to the visual exploration of the data, it proved necessary to delve more deeply and explicitly into the model using direct queries.

Such queries were useful for extracting answers to questions that relied upon the meta-data for each node in the model. For example, Figure 5 depicts a group of three data sources that are only lightly used in the CND decision process. This pattern reflects the limited utility of that group due to low data quality; a point that was raised by many interviewees. The importance of these data sources, however, is understated by this view. Only by considering the criticality of those data to their SA information requirements and by other meta-data can the true effect of this low quality be seen.

Due to time limitations, we were unable to build an interface that abstracted away the need to write SQL, so instead we met the need with a simple text-based interface that can pass SQL to the database and receive back formatted data structures. This data can then be visually analyzed, or cleaned in a text editor for further analysis in Excel.

Despite the inelegance and high burden on the user of this access functionality, it is hard to understate its power relative to using only a simple spreadsheet. In minutes, a reasonably skilled author of SQL queries can accurately answer questions that require aggregation across multiple levels of the decision model.

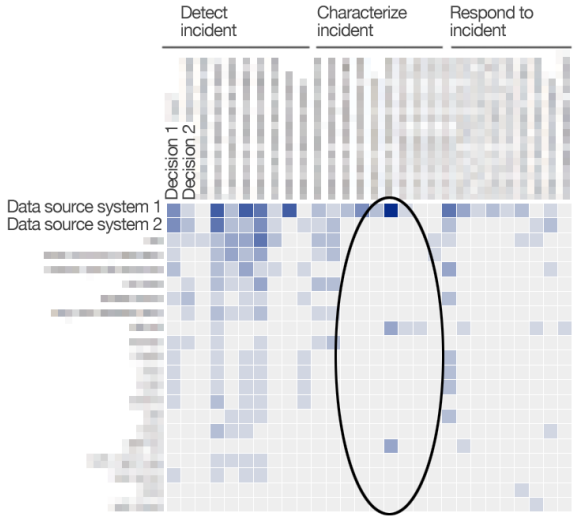

Figure 6. Data source usage frequency by decision

Answering such questions would take 10 to 15 times longer to complete by hand. Sample questions we asked were "How many decisions does this data source support?", and "How many times are data sources used by each decision?".

4.2.4 Flexibility

One final advantage to having the model represented in a database was that it meant that the team could pull out interesting data sets and manually visualize them using other means, including other Protovis layouts.

Figure 6 contains a matrix that shows the usage frequency of data sources across decisions. The shading of each cell indicates how many times a given data source system can be employed by a decision (since the SA information requirements under a decision can each rely on the same data source system).

The matrix was developed to directly show the contribution of data sources to decisions, following the discovery of the highly linked cluster of data sources shown in Figure 5. It proved to be somewhat useful for judging the relative importance of data sources to different decisions in the CND decision process – its primary shortcoming is that it doesn't convey any information from the meta-data that characterizes each data source in the context of a specific SA information requirement (e.g. criticality).

The matrix is also useful to rapidly identify which data source systems are employed in the resolution of any given CND decision. By sighting down columns, one can quickly see the data sources that can be employed in a decision. By charting all of this data in one chart, the chance of seeing patterns increases. For example, the middle area has much fewer filled boxes than the left edge; this reveals a dearth of data source systems relating to incident characterization when compared to incident detection.

The y-axis sorting on overall data source usage frequency also allows a person to quickly determine the most and least relied-upon data sources. This, in turn, helps identify data sources underutilized in the decision process, possibly due to technical or political obstacles.

5. CONCLUSIONS

The tool described in this paper provided an effective and efficient means of representing and analyzing the hierarchical decision model we built to model the computer network defense decision process of an organization.

The tool was effective, ultimately, because it helped us shape a higher fidelity understanding of our client's CND decision process than was possible using a spreadsheet representation. We used its visual analytic capability to help us answer the question "On what data sources and systems do these decisions depend?", discovered useful patterns in how data sources are employed by CND decisions, and provided insight into which SA information requirements supported multiple sub-goals and goals.

It was efficient because it saved time. A rapid development pace was enabled by the availability of high quality web-related technologies. The database structure allowed us to painlessly alter model nodes (e.g., a data source) in one place and the interactive tree visualization allowed us to quickly discover and correct previously unnoticed data entry and coding errors. Additionally, the database editor was used by a member of the administrative staff to define the bulk of the model's links and corresponding meta-data in one working day after only 30 minutes of training.

This is not to say that the tool was without its flaws; several shortcomings stand out:

First, although the database editor proved accessible to users, the workflow for setting per-link meta-data is repetitive and tedious. Furthermore, the database editor doesn't afford smooth navigation up the decision model structure; for example, if you are working on a decision, it is not possible to see under which sub-goal that decision falls. When revising links and verifying data entry, this limitation imposes a significant burden on the user.

Second, as was previously mentioned, the lack of labels on the interactive tree view impairs analysis of the model. If the opportunity arises, it would be nice to explore incorporating labels on the vertex itself, possibly with a control to disable the display of labels if it is not always a desirable feature.

Third, the lack of automated sorting in the interactive tree view made visual analysis of the model more difficult. In retrospect, we were lucky that the SA information requirements were roughly located beneath the decisions they supported. The layout algorithm should be enhanced to attempt to minimize the sum of incident edge lengths.

Fourth, the current visualizations do not convey the full depth of information stored in the database. It would have been nice to have time to explore more sophisticated views that convey more of the meta-data information for each level of the decision model.

6. ACKNOWLEDGMENTS

This work would not have been possible without the cooperation of the CND analysts who provided us with access to their facilities and insight into their processes.

7. REFERENCES

[1] A Survey of Cognitive Engineering Methods and Uses: *http://mentalmodels.mitre.org/cog_eng/ce_methods_I.htm*. Accessed: 2011-04-04.

[2] Abdullah, K., Lee, C.P., Conti, G., Copeland, J.A. and Stasko, J. 2005. IDS rainStorm: visualizing IDS alarms. (Oct. 2005), 1- 10.

[3] Albert, C., Dorofee, A.J., Killcrece, G., Ruefle, R. and Zajicek, M. 2004. *Defining Incident Management Processes for CSIRTs: A Work in Progress*. Technical Report #CMU/SEI-2004-TR-015. Carnegie Mellon University (CMU) Software Engineering Institute (SEI).

[4] BaseHTTPServer — Basic HTTP server: *http://docs.python.org/library/basehttpserver.html*. Accessed: 2011-04-04.

[5] Biros, M.D.P. and Eppich, C.T. 2001. Human Element Key to Intrusion Detection. *Signal*.

[6] Chu, M., Ingols, K., Lippmann, R., Webster, S. and Boyer, S. 2010. Visualizing attack graphs, reachability, and trust relationships with NAVIGATOR. *Proceedings of the Seventh International Symposium on Visualization for Cyber Security* (2010), 22–33.

[7] D'Amico, A. and Whitley, K. 2008. The Real Work of Computer Network Defense Analysts. *VizSEC 2007*. J.R. Goodall, G. Conti, and K.-L. Ma, eds. Springer Berlin Heidelberg. 19-37.

[8] 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. *Human Factors and Ergonomics Society Annual Meeting Proceedings* (2005), 229–233.

[9] Gheisari, M., Irizarry, J. and Horn, D.B. 2010. Situation Awareness Approach to Construction Safety Management Improvement. *Procs 26th Annual ARCOM Conference* (Leeds, UK, Sep. 2010), 311-318.

[10] Goodall, J.R., Lutters, W.G. and Komlodi, A. 2004. I Know My Network: Collaboration And Expertise. *In Proc. of CSCW 2004*. 2004, (2004), 342-345.

[11] Jamieson, G.A., Miller, C.A., Ho, W.H. and Vicente, K.J. 2007. Integrating Task- and Work Domain-Based Work Analyses in Ecological Interface Design: A Process Control Case Study. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*. 37, 6 (Nov. 2007), 887-905.

[12] Jones, D.G. and Endsley, M.R. 2005. Goal Directed Task Analysis. *Protocols for Cognitive Task Analysis*.

[13] Killcrece, G., Kossakowski, K.-P., Ruefle, R. and Zajicek, M. 2003. *State of the Practice of Computer Security Incident Response Teams (CSIRTs)*. Technical Report #CMU/SEI-2003-TR-001. Carnegie Mellon University (CMU) Software Engineering Institute (SEI).

[14] Lee, C.P. and Copeland, J.A. 2006. Flowtag: a collaborative attack-analysis, reporting, and sharing tool for security researchers. *Proceedings of the 3rd international workshop on Visualization for computer security* (New York, NY, USA, 2006), 103–108.

[15] Marty, R. 2008. *Applied Security Visualization*. Addison-Wesley Professional.

[16] McCloskey, M.J., Stanard, T.W. and Armstrong, A.A. 2001. *A cognitive analysis of the information security domain*. Technical Report #ARFL-HE-WP-TR-2001-0041. Klein Associates Inc.

[17] McPherson, J., Ma, K.-L., Krystosk, P., Bartoletti, T. and Christensen, M. 2004. PortVis: a tool for port-based detection of security events. *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security* (New York, NY, USA, 2004), 73–81.

[18] O'Hare, S., Noel, S. and Prole, K. 2008. A Graph-Theoretic Visualization Approach to Network Risk Analysis. *Proceedings of the 5th international workshop on Visualization for Computer Security* (Berlin, Heidelberg, 2008), 60–67.

[19] py2exe.org: *http://www.py2exe.org/*. Accessed: 2011-04-04.

[20] Python Programming Language: *http://www.python.org/*. Accessed: 2011-04-04.

[21] SQLite Home Page: *http://www.sqlite.org/*. Accessed: 2011-04-04.

[22] Visualizing a Security Attack on a VOIP Honeypot Server - information aesthetics: *http://infosthetics.com/archives/2011/03/visualizing_a_voip_security_attack.html*. Accessed: 2011-04-04.