A Task Centered Framework for Computer Security Data Visualization

Xiaoyuan Suo, Ying Zhu, and Scott Owen

Department of Computer Science
Georgia State University
xsuo@student.gsu.edu, yzhu@cs.gsu.edu, sowen@gsu.edu

Abstract. Most of the existing computer security visualization programs are data centered. However, some studies have shown that task centered visualization is perhaps more effective. To test this hypothesis, we have developed a new framework of visualization and apply it to computer security visualization. This framework provides a new way for users to interact with data set and potentially will provide new insights into how visualization can be better constructed to serve users' specific tasks.

Keywords: visualization, task, computer security.

1 Introduction

A fundamental question for visualization design is what makes visualizations effective? There have been different answers to this question. Some researchers take a more data-centric view and suggest that effectiveness depends on the accurate interpretation of presented data [1-3], or a matching between data structure and visual structure [4, 5]. However, a number of psychological studies have also shown that the effectiveness of visualization is task specific [6, 7].

In this paper, we propose a new task centered framework of visualization and apply it to computer security visualization. In our framework, a visualization system is optimized for specific tasks by mapping the task related parameters to the visual elements that have high accuracy, utility, and efficiency ratings.

Before visualizations are created, users specify tasks and their associated parameters. This process is essentially a task complexity analysis. Knowing the data parameters associated with a task helps users consciously control the complexity of the tasks and correlate task complexity and visualization complexity. This new framework provides a different way for users to interact with data set and potentially will provide new insights into how visualization can be better constructed to serve users' specific tasks.

2 Related Work

Many visualization designs have been proposed for computer security analysis. Noted examples include TNV [8], IDS RainStorm [9], PortVis [10], etc. Most of these designs, however, are prefabricated visualizations that cannot be easily reconfigured by users for different tasks. An implicit assumption is that users can use interaction techniques to customize data visualization for different tasks. While interaction is essential for making visualization usable, two important issues need to be addressed. First, for most existing visualization systems it is often not clear what specific tasks they are designed for. As a result, users may use the visualization for unintended tasks. Second, most existing visualization systems provide only low level interaction techniques, such as zooming, panning, that are restrained by the predefined visualization structure. They may be suitable for problem solving process with relatively stable procedure and task structure. However, many complex problem solving process are not so well defined. In many cases, problem solving is a process of searching in the solution space. This means that users may constantly testing different hypotheses and apply different strategies. The task structure may keep changing during the problem solving process. A new set of higher level interaction techniques are needed to support this dynamic problem solving method.

Some visualization systems, such as RUMINT [11], do provide a more configurable interface that allow users to assign parameters to different coordinate axes, or choose different types of diagrams. Outside the field of computer security visualization, Tableau Software is noted for its highly flexible and configurable interface that allows users to quickly construct different data visualizations. Another example is Many Eyes [12], a web site that allows different users to construct different visualizations of the same data set. Although these are powerful user interface techniques, users are still operating at the visualization level. But most end users would prefer to operate at a higher level of thinking – ask questions, test hypotheses, etc. For end users, constructing and configuring visualization is a secondary activity to their primary tasks. Again, we need a higher level interaction technique to help end users operate at the level of tasks.

The research presented in this paper is an attempt to address this issue. The central component of the proposed visualization framework is a task tree that is dynamically linked to data visualizations and data tables. Users operate by constructing and maintaining a task tree. A frame of data visualization is created automatically (or semi-automatically) for each task on the task tree, with the support of a visualization engine.

3 Overview of Task Centered Visualization Framework

We propose a Task-centered Visualization Design Architecture (TVDA). Figure 1 shows the main components of TVDA.

To construct a visualization, users start with a visual frame and then drag and drop visual structures into the view. They are assisted by a design-gallery style interface [13, 14] that contains multiple visual structures provided by a visualization engine.

For domain experts, a typical visual problem solving process takes the following steps:

- Open the data files or connect to the databases.
- Divide the work into multiple tasks. Create a hierarchical task tree.
- Associate data parameters with each task.

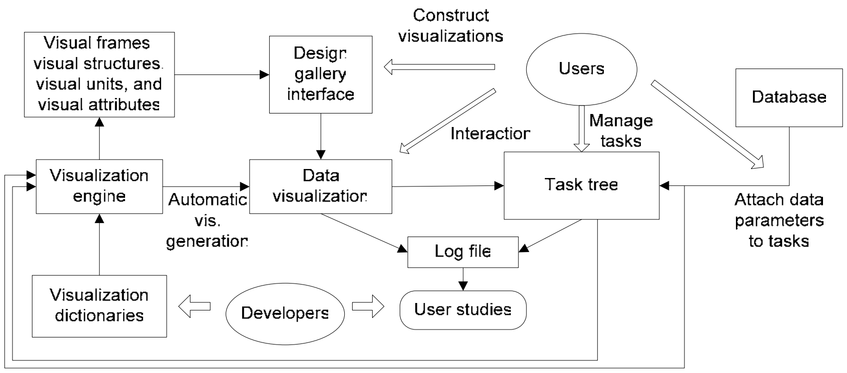


Fig. 1. Overview of the Task-centered Visualization Architecture

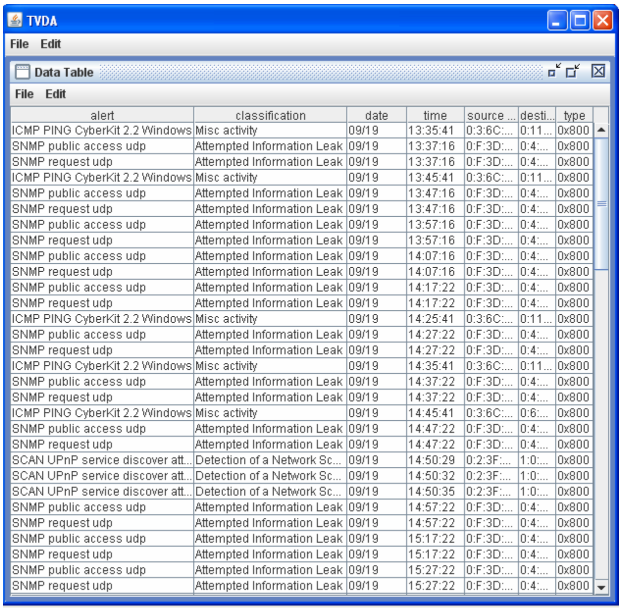
- For each task, construct a data-visualization. A visualization engine will automatically recommend multiple design choices, which are presented in a design-gallery style interface. The designs are selected and ranked based on their accuracy, utility, and efficiency scores in the visualization dictionaries.
- Explore the data visualization through interaction techniques.

Tree is an appropriate data structure for organizing and storing problem solving activities [15-17]. Each node on the task tree represents a specific task. For each task, users shall explicitly identify the type of this task, based on the task classification conducted in step 3.

For each task, users are required to explicitly identify the data parameters that are needed to perform the task. More specifically, these are the parameters that have to be kept in the working memory simultaneously in order to carry out the task. Based on the Relational Complexity Theory [18, 19], the complexity of the task is determined by the number of these parameters. The proposed visualization tool allows users to open data files, select parameters, and attach them to a task.

A task tree also has other benefits. First, the task tree itself can be seen as a visualization of the problem solving process, reducing the cognitive load by externalizing the task structure that would otherwise be stored in the working memory. Second, a task tree is essentially a visual language for describing a specific problem solving strategy and expertise [20, 21], which can be shared or reused.

User controlled visualization construction is necessary for several reasons. First, complex problem solving is a dynamic process. In search for a solution, users need to test different hypotheses or different strategies. This means the task structure may be constantly changing, and a good visualization tool should allow users to dynamically reorganize visualizations to accommodate this change – because the effectiveness of visualization is task specific. Second, studies have shown that the effectiveness of visualizations depends on users' background and knowledge. Visualization is also a learned skill – as users become more experienced, their behavior for reading and constructing visualization may change [22]. Prefabricated visualizations combined with low level interactions – such as zooming, panning, and level-of-detail – are insufficient to address the individual differences. Third, self-constructed visualizations may



alert	classification	date	time	source ...	desti...	type
ICMP PING CyberKit 2.2 Windows	Misc activity	09/19	13:35:41	0:3:6C:...	0:11...	0x800
SNMP public access udp	Attempted Information Leak	09/19	13:37:16	0:F:3D:...	0:4:...	0x800
SNMP request udp	Attempted Information Leak	09/19	13:37:16	0:F:3D:...	0:4:...	0x800
ICMP PING CyberKit 2.2 Windows	Misc activity	09/19	13:45:41	0:3:6C:...	0:11...	0x800
SNMP public access udp	Attempted Information Leak	09/19	13:47:16	0:F:3D:...	0:4:...	0x800
SNMP request udp	Attempted Information Leak	09/19	13:47:16	0:F:3D:...	0:4:...	0x800
SNMP public access udp	Attempted Information Leak	09/19	13:57:16	0:F:3D:...	0:4:...	0x800
SNMP request udp	Attempted Information Leak	09/19	13:57:16	0:F:3D:...	0:4:...	0x800
SNMP public access udp	Attempted Information Leak	09/19	14:07:16	0:F:3D:...	0:4:...	0x800
SNMP request udp	Attempted Information Leak	09/19	14:07:16	0:F:3D:...	0:4:...	0x800
SNMP public access udp	Attempted Information Leak	09/19	14:17:22	0:F:3D:...	0:4:...	0x800
SNMP request udp	Attempted Information Leak	09/19	14:17:22	0:F:3D:...	0:4:...	0x800
ICMP PING CyberKit 2.2 Windows	Misc activity	09/19	14:25:41	0:3:6C:...	0:11...	0x800
SNMP public access udp	Attempted Information Leak	09/19	14:27:22	0:F:3D:...	0:4:...	0x800
SNMP request udp	Attempted Information Leak	09/19	14:27:22	0:F:3D:...	0:4:...	0x800
ICMP PING CyberKit 2.2 Windows	Misc activity	09/19	14:35:41	0:3:6C:...	0:11...	0x800
SNMP public access udp	Attempted Information Leak	09/19	14:37:22	0:F:3D:...	0:4:...	0x800
SNMP request udp	Attempted Information Leak	09/19	14:37:22	0:F:3D:...	0:4:...	0x800
ICMP PING CyberKit 2.2 Windows	Misc activity	09/19	14:45:41	0:3:6C:...	0:6:...	0x800
SNMP public access udp	Attempted Information Leak	09/19	14:47:22	0:F:3D:...	0:4:...	0x800
SNMP request udp	Attempted Information Leak	09/19	14:47:22	0:F:3D:...	0:4:...	0x800
SCAN UPnP service discover att...	Detection of a Network Sc...	09/19	14:50:29	0:2:3F:...	1:0:...	0x800
SCAN UPnP service discover att...	Detection of a Network Sc...	09/19	14:50:32	0:2:3F:...	1:0:...	0x800
SCAN UPnP service discover att...	Detection of a Network Sc...	09/19	14:50:35	0:2:3F:...	1:0:...	0x800
SNMP public access udp	Attempted Information Leak	09/19	14:57:22	0:F:3D:...	0:4:...	0x800
SNMP request udp	Attempted Information Leak	09/19	14:57:22	0:F:3D:...	0:4:...	0x800
SNMP public access udp	Attempted Information Leak	09/19	15:17:22	0:F:3D:...	0:4:...	0x800
SNMP request udp	Attempted Information Leak	09/19	15:17:22	0:F:3D:...	0:4:...	0x800
SNMP public access udp	Attempted Information Leak	09/19	15:27:22	0:F:3D:...	0:4:...	0x800
SNMP request udp	Attempted Information Leak	09/19	15:27:22	0:F:3D:...	0:4:...	0x800

Fig. 2. Data table

assist problem solving in ways different from prefabricated visualizations [6, 23]. Over time, these benefits will outweigh the initial learning curve.

In addition to the traditional interaction techniques (e.g. zooming, panning, level-of-detail [24]), the TVDA also allows users to be able to merge two or more visual frames. This technique is designed to reduce the cognitive load of visual integration and inference by externalizing the mental transformations [25-28].

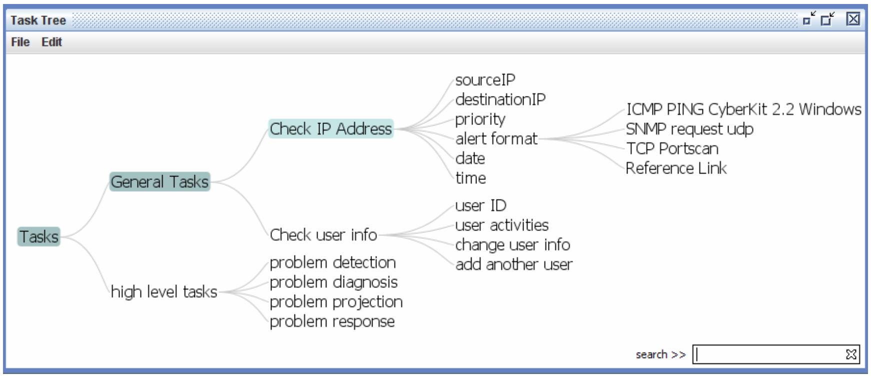


Fig. 3(a). Task tree

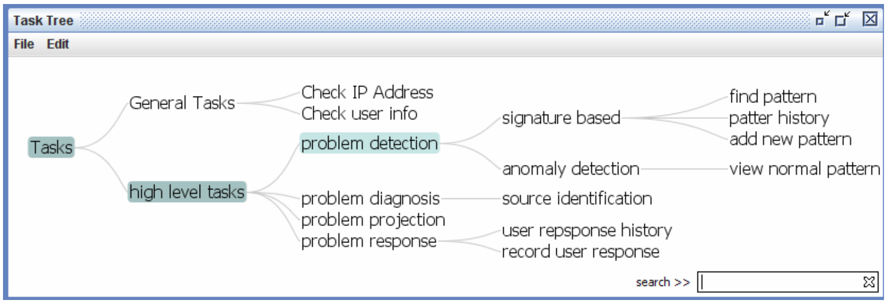


Fig. 3(b). High level task list

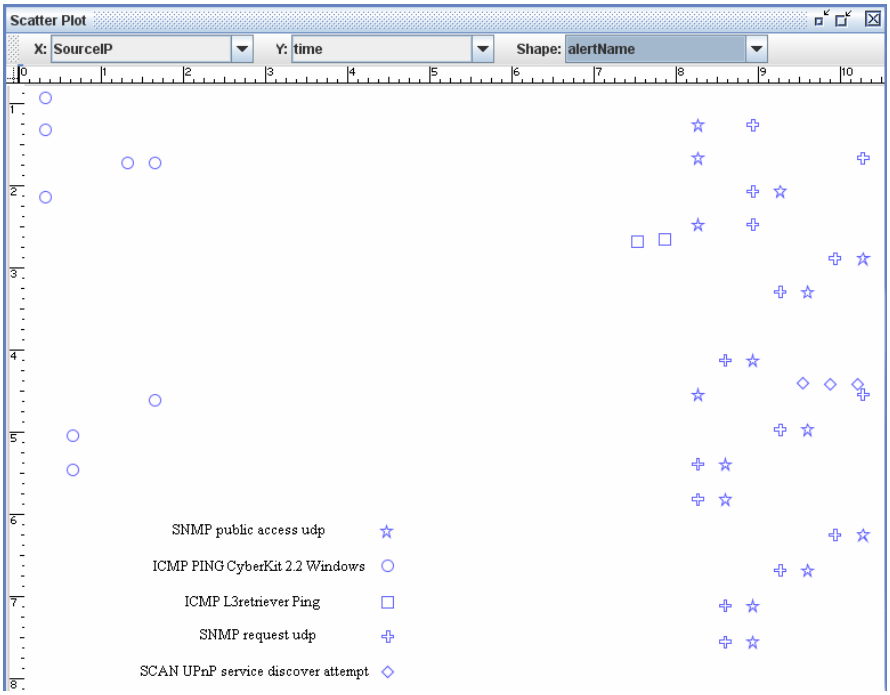


Fig. 4. Scatter plot. Different shapes represent different alert types. User can choose different parameters (Source IP, time, date, priority, alert name) for the X and Y axes.

4 Implementation

We have built a prototype based on the proposed framework. This prototype is implemented with Java and uses the *prefuse* [29] – an open source interactive information visualization library.

Current system has three main windows: data table, task tree, and visualization.

1. Data table (Figure 2):
   When the data is loaded, the system will process the raw data (Snort etc.), extract the necessary information, and then display the relevant data in a table. The data table is intended for security experts to exam raw data and also to drag and drop parameters into the task tree.
2. Task tree (Figure 3):
   Tasks will be displayed in a tree format, in which each node represents a task. A task can be divided into sub-tasks. Each task is associated with a number of parameters, as explained earlier. Low level tasks are also available using menu bars on top of the main visualization (zooming, panning, etc.).
3. Visualization (Figure 4):
   When a task is selected in the task tree, the desired visual interface is automatically generated for that task. In this visualization, each different shape represents a different parameter. Figure 4 shows the visualization generated for the task "Check IP Address" and "problem detection". The different shapes on scatter plot represented different information extracted during a short period of time.

5 Conclusion and Future Work

We propose a task centered visualization design framework, in which tasks are explicitly identified and organized and visualizations are constructed for specific tasks and their related data parameters. The center piece of this framework is a task tree which dynamically links the raw data with automatically generated visualization. The task tree serves as a high level interaction technique that allows users to conduct problem solving naturally at the task level, while still giving end users flexible control over the visualization construction.

Much work needs to be done to realize the full potential of the proposed framework. Our future work includes developing a design gallery style visualization interface that allows users to compare and select from multiple visualizations that are automatically generated. A significant challenge is to develop a visualization engine that helps automatically generate visualizations given a task and its related parameters. The key is to codify the many design rules from the visualization research literature and to develop a systematic method to evaluate and optimize the visualization. Our previous work on visualization complexity analysis [30] can be used as the basis for the evaluation and optimization. Finally, we will develop an evaluation plan to test the effectiveness of the proposed framework, working with domain experts in the field of computer security.

References

[1] Mackinlay, J.: Automating the Design of Graphical Presentations of Relational Information. ACM Transactions on Graphics 5, 110–141 (1986)

[2] Cleveland, W.S., McGill, R.: Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. Journal of the American Statistical Association 79, 531–554 (1984)

[3] Cleveland, W.S., McGill, R.: Graphical Perception and Graphical Methods for Analyzing Scientific Data. Science 229, 828–833 (1985)

[4] Dastani, M.: The Role of Visual Perception in DataVisualization. Journal of Visual Languages and Computing 13, 601–622 (2002)

[5] Wattenberg, M., Fisher, D.: Analyzing perceptual organization in information graphics. Information Visualization 3, 123–133 (2004)

[6] Cox, R.: Representation construction, externalised cognition and individual differences. Learning and Instruction 9, 343–363 (1999)

[7] Freedman, E.G., Shah, P.: Toward a Model of Knowledge-Based Graph Comprehension. In: Hegarty, M., Meyer, B., Narayanan, N.H. (eds.) Diagrams 2002. LNCS (LNAI), vol. 2317, pp. 59–141. Springer, Heidelberg (2002)

[8] Goodall, J.R., Lutters, W.G., Rheingans, P., Komlodi, A.: Preserving the Big Picture: Visual Network Traffic Analysis with TNV. In: Workshop on Visualization for Computer Security, Minneapolis, MN, USA, pp. 47–54 (2005)

[9] Abdullah, K., Lee, C., Conti, G., Copeland, J.A., Stasko, J.: IDS RainStorm: Visualizing IDS Alarms. In: IEEE Symposium on Information Visualization's Workshop on Visualization for Computer Security (VizSEC) (2005)

[10] McPherson, J., Ma, K.-L., Krystosk, P., Bartoletti, T., Christensen, M.: PortVis: a tool for port-based detection of security events. In: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security, ACM Press, Washington (2004)

[11] Conti, G.: Security Data Visualization: Graphical Techniques for Network Analysis. No Starch Press (2007)

[12] Viegas, F.B., Wattenberg, M., Ham, F.v., Kriss, J., McKeon, M.: Many Eyes: A Site for Visualization at Internet Scale. In: Proceedings of the IEEE Symposium on Information Visualization (2007)

[13] Marks, J., Andalman, B., Beardsley, P.A., Freeman, W., Gibson, S., Hodgins, J., Kang, T.: Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation. In: Proceedings of ACM SIGGRAPH Conference (1997)

[14] Terry, M.: Set-Based User Interface, in PhD Thesis, School of Computing, Georgia Institute of Technology, Atlanta, Georgia (2005)

[15] Bratko, I.: PROLOG Programming for Artificial Intelligence, 2nd edn. Addison-Wesley Longman Publishing Co., Inc., Amsterdam (1990)

[16] Pain, H., Bundy, A.: What stories should we tell novice PROLOG programmers? In: Artificial intelligence programming environments, pp. 119–130. John Wiley & Sons, New York (1987)

[17] Simmons, R., Apfelbaum, D.: A Task Description Language for Robot Control. In: Proceedings of the 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems, Victoria, B.C., Canada (1998)

[18] Halford, G.S., Baker, R., McCredden, J.E., Bain, J.D.: How Many Variables Can Humans Process? Psychological Science 16, 70–76 (2005)

[19] Halford, G.S., Wilson, W.H., Phillips, S.: Processing capacity defined by relational complexity: Implications for comparative, developmental, and cognitive psychology. Behavioral and Brain Sciences 21, 803–865 (1998)

[20] Casner, S., Bonar, J.: Using the expert's diagram as a specification of expertise. In: Proceedings of IEEE Symposium on Visual Languages (1988)

[21] Davies, J., Goel, A.K.: Transfer of problem-solving strategy using Covlan. Journal of Visual Languages and Computing 18, 149–164 (2007)

[22] Petre, M., Green, T.R.G.: Learning to Read Graphics: Some Evidence that 'Seeing' an Information Display is an Acquired Skill. Journal of Visual Languages and Computing 4, 55–70 (1993)

[23] Cox, R., Brna, P.: Supporting the use of external representation in problem solving: the need for flexible learning environments. Journal of Artificial Intelligence in Education 6, 239–302 (1995)

[24] Shneiderman, B.: The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In: Proceedings of the IEEE Conference on Visual Languages. IEEE, Los Alamitos (1996)

[25] Ratwani, R.M., Trafton, J.G.: Making Graphical Inferences: A Hierarchical Framework. In: Proceedings of the Annual Meeting of the Cognitive Science Society (CogSci) (2004)

[26] Ratwani, R.M., Trafton, J.G., Boehm-Davis, D.A.: Thinking Graphically: Extracting Local and Global Information. In: Proceedings of the Annual Meeting of Cognitive Science Society (2003)

[27] Trafton, J.G., Kirschenbaum, S.S., Tsui, T.L., Miyamoto, R.T., Ballas, J.A., Raymond, P.D.: Turning pictures into numbers: extracting and generating information from complex visualizations. International Journal of Human-Computer Studies 53, 827–850 (2000)

[28] Trafton, J.G., Trickett, S.B.: A New Model of Graph and Visualization Usage. In: Proceedings of the Annual Meeting of Cognitive Science Society (2001)

[29] Heer, J., Card, S.K., Landay, J.A.: Prefuse: A Toolkit for Interactive Information Visualization. In: Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI) (2005)

[30] Suo, X., Zhu, Y., Owen, G.S.: Measuring the Complexity of Visualization Design. In: Proceedings of the 2007 Workshop on Visualization for Computer Security (VizSEC) (2007)