Adaptive Security Event Visualization for Continuous Monitoring

Anatoly Yelizarov, Dennis Gamayunov

Lomonosov Moscow State University
Department of Computational Mathematics and Cybernetics
Information Systems Security Lab
{tolya, gamajun}@lvk.cs.msu.su

Abstract. The field of information security routinely produces the need for a security information and event management system operator who would be capable of durable and extensive (e.g., workday-long) monitoring of the system in his control with well-timed decision making in emergencies. The obvious concern is that such continuous exertion is bound to lead to the operator's increased fatigue, reduced attention span, and flawed decision making. This paper proposes methods of the visualization system's adaptation to these changes for improving the operator's efficiency in terms of speed and accuracy.

Keywords: Adaptive User Interfaces, Event Visualization, Information Security

1 Introduction

The problem of increasing the effectiveness of the "human-machine" system is tackled from two different aspects: by better preparing the human for work with a given machine (through training and building up experience) and by better *adapting the machine* to the specific objectives of the human. In the latter case, special attention is paid to the parts of the system where human directly interacts with machine or, in other words, the parts responsible for information reception and transmission. These parts go beyond the visualization system, screen, mouse, and keyboard, to include also the human himself as the one who perceives and interprets the information, and then interacts with the machine to convey his decisions. Differently put, there must be close attention to properties of the human as a "data channel".

One relevant example of such a "human-machine" system is exactly the "human operator and information security visualization system". Considering the ever increasing number of cyber-attacks, there is a growing need, on both corporate and government security levels, for the human operator of the attack detection system. Frequently, such an operator is tasked with *continuous* (through the entire working day) monitoring of the networks and services under his control. Besides, such an operator must make timely decisions in emergencies by performing typical actions that protect the monitored resources (e.g., blocking external hosts and subnets, running and shutting down services on the protected hosts and simple tweaking of the system's operation logic). Hereinafter such human operator will be referred to as "*the operator*".

The obvious concern is that workday-long exertion would result in increased operator fatigue, reduced attention span, and flawed decision making. Although we have not yet received sufficient supporting test data, further discussion proceeds from the assumption that *psychophysical condition of the operator varies over time* as the fatigue builds up, blunting his responses and worsening the perception quality of the same cognitive load level; though, these characteristics may change in the opposite direction as well (e.g., after a break or in the state of heightened alert).

As the operator is the only one responsible for making well-timed and correct decisions, it can be presumed that careful *consideration of the current psychophysical condition and individual cognitive features* of different operators would benefit the speed and accuracy of their decisions. However, the visualization systems now available (e.g., the monitoring modules in the security information and event management solutions from major vendors) provide versatile visualization tools designed for the "average" user. Whereas such systems do usually provide means for interface customization, they typically ignore current user individual abilities.

This paper discusses and proposes the methods to adapt the visualization system's functionality and presentation capabilities to the psychophysical condition of the operator; in par-

ticular, it focuses on ways in which the visualization system could improve the decision making process by taking into account changes of human-computer interaction (HCI) indicators according to operator's fatigue, reduced attention, etc. A further discussion narrows this focus to the informational security visualization systems and theirs operators, leaving the door open to generalization to any event-based visualization system and its continuously working user.

2 Background and Related Work

2.1 Information Security Visualization Systems

The hardware and software systems which support information security are commonly referred to as *security information and event management* (SIEM) solutions. A typical SIEM architecture consists of a security events collection module (that receives information from various sources, e.g., intrusion detection systems (IDS), intrusion prevention systems, firewalls, operating systems, databases, various applications, etc.), correlation and analysis module, database of security events, and monitoring module. The latter is generally divided into reporting module (to generate reports with information required for security administrators), and visualization module (responsible for displaying security events and status of network devices in real-time, visualization of past attacks, providing an interface for working with security event database, and providing opportunity for efficient management and resolution of incidents).

Of the SIEM architecture just described, this paper focuses solely on the visualization module (hereinafter such module is referred to as "*the visualization system*"). Visualization for information security is a relatively young domain that studies, designs new, and adapts common visualization techniques for information security data. These data are characterized by a large volume, lots of parameters and the need for real-time display. Comprehensive coverage of that topic, with emphasis on the visualization systems, is presented in surveys [16] and [10].

2.2 Cognition in Human-Computer Interaction

Cognitive psychology and cognitive science are the major research areas addressing the mental processes such as attention, perception, and problem solving, which influence the operator's decision making efficiency. *Human-computer interaction* (HCI) studies and designs interaction between people and machine. The application of cognitive sciences to problems of HCI is called cognitive ergonomics [12]. HCI consists of factors associated both with the computer (such as equipment, performance, and software) and the human (such as training, experience, and individual differences) [15]. The term “individual differences” denotes the user's personal cognitive abilities rather than demographic description, though it may and does affect his cognitive abilities. Recent researches have confirmed that some of these individual *cognitive abilities*, such as working memory capacity [11],[20],[18], perceptual speed [20],[19], spatial ability [20], and locus of control [24], correlate with the user's performance as to speed and accuracy. The nature of human *visual perception* specific to visualization is covered in depth in [22]. Limitations of human cognition, and their impact on information security visualization are discussed in [3] by reviewing vulnerabilities of operator's visual perception and how they can be exploited in cyber-attacks.

2.3 User-Adaptive Visualization

Successful design of HCI starts from understanding the user, his tasks, and context of his actions [9]. An *adaptive interface* is generally defined as an interface that automatically varies its layout and elements adapting to the user's needs, task and context. Early adaptive interfaces could adapt only to their tasks or the data to be displayed, ignoring any information about the current user. With progress of research and technology, new user tracking techniques were developed — click-streams and eye gaze processing, physical and biomedical sensors, and user models, to name a few. Recent studies provide encouraging evidences that user's cognitive ability could be reliably detected in real time using eye gaze information [19] or [17].

One example of the adaptive interface is the *attentive user interface* that is a user interface aimed to support the user’s attention capacities [21]. Such an interface arises where the visualization system is blended with technologies allowing to track and infer priorities of user attention [14]. Current attentive interfaces are able to maximize the expected utility from the information user receive, thus increasing the efficacy of the HCI [8]. But the fundamental challenge

[7] of any new such attentive technique is reasoning about its interruption costs (e.g., periphery animation cause distraction from attention- and motion-intensive tasks [4]).

Another approach includes *adaptation to the user's experience*. For example, [19] presumes that in some cases non-experts may benefit from adaptive intervention (based on different effect on the label/legend access for different visualizations); or [1] that adapts the content of visualization in an educational system according to the user's domain knowledge. A *behavior-based adaptation* approach is presented in [6], where the system relies on the use of click-stream analysis to detect semantically meaningful patterns, and recommends a specific user support visualization for his current task.

3 Security Event Visualization

Further discussion is centered around the *security event visualization system we are developing* as a part of our research. Visualization, for our purposes here, is taken to mean a way of presenting information in the form of optical images only. Our other premise is that the mouse and keyboard are the only equipment we use to receive the HCI indicators from (more advanced mentioned tracking technologies such as eye gaze, physical and biomedical sensors are beyond this research). For simplicity, it can be presumed that visualization system receives messages from an IDS containing all *information to be displayed*. Any such message should include the following information on the detected event: the IP addresses of the target and source hosts; the type (e.g., scan, remote root attempt, denial-of-service (DoS), etc.) and severity level of the event (low, medium, high, or information messages), as well as its time of occurrence and links to related events. This latter piece of information is received from an IDS correlation module, and is essential for displaying complex [23] or multi-step attacks. All information received from IDS is stored in a separate database that operates with two main entities: hosts and attacks (every attack is unique and may be described recursively: it is either a single event or an array of attacks).

3.1 Use Case Scenario

We will demonstrate our approaches using an exemplary scenario. Let us assume three simultaneous complex attacks detected within ≈10 seconds in the monitored network of ≈1000 hosts. The first of these is a distributed denial-of-service (DDoS) attack against certain network segment, the second is a DDoS attack against a certain host, and the third is a multi-step attack consisting of distributed scanning, root access to monitored host, and a DoS attack against another monitored host. The messages hitting the visualization system during this period fall into the following types: scan (low severity), remote root attempt (medium severity), successful remote root (high severity), DoS (medium severity). In the situation just described, the visualization system receives thousands of messages from the IDS, and the *operator has to come up with the following decisions* as quickly as possible: block the autonomous system and/or ranges of the IP addresses that appear to be involved in DDoS attacks and distributed scanning;reconfigure the captured host and transfer its services to another non-compromised host; and restore functionality of the hosts hit by the DDoS attacks if required.

We decided that the *highest priority data* for operator's decision making include the addresses of both the target host and the attacker, severity of the event, and links to related events. The time of occurrence and type of event were relegated to secondary parameters. The *time of event* for continuously working operator is not that important as he has to resolve first the most dangerous problem rather than the most recent one. As to the *type of event*, it is less important for decision making than the severity level, because the operator does not need any in-depth analysis of the situation; all he has to do is resolve the attack of the highest severity. On the other hand, it is the type of event that defines which decisions the operator can make in a specific situation — the factor taken into account by the visualization system when it prompts the operator to act (as shown in Figure 2).

3.2 Visualization

The traditional approach to visualization system design induces operator to switch among the modes, views and tabs in order to perceive all relevant nuances to making a decision (not to mention that the operator usually cannot interact directly with the visualization to implement his decision). We have assumed that the time taken to find a solution depends, among other

factors, on the number of "*search steps*" (or, in other words, on the number of switching context of perception). We therefore stepped aside from the conventional approach and made our visualization system display all its data within one OpenGL-widget. More specifically, we have built our system using Qt library components, with the visualization taking place *within a single QGLWidget*.

It can be considered [22] that the most separable dimensions for perception are color, elements of form (size and orientation), position in the 2D-space and simple motion. We presumed that the most natural, and therefore easiest-to-perceive, visualization of the monitored *network is a graph* with topology corresponding to the reality. Thus, the edges of the graph are consistent with data transmission channels and the vertices correspond to the hosts. That is, of the four most separable dimensions, the position in the 2D-space is set aside for host allocation.

We color-coded the *severity level* of event as it has only four values. Each vertex of the monitored network graph is colored in line with the most severe related event and highlighted with a color shade of the second most severe related event.

External hosts are displayed outside of the space occupied by the network graph. The entire external host space can be represented as a stripe with the X-axis corresponding to the two most significant bytes of an external IP-address (first two in the big-endian notation), and the Y-axis consistent with the two least significant bytes. The more external hosts from close subnets are involved in a certain attack, the greater is their display size (we believe that this feature helps to *identify botnets and distinguish false positive events on legitimate external users*). The stripe may be divided into up to 10 areas, corresponding to different attacks those can be binded to digit keys on a keyboard (see the Interaction section for details). The corners of that space are utilized to display the external hosts which are involved in several attacks.

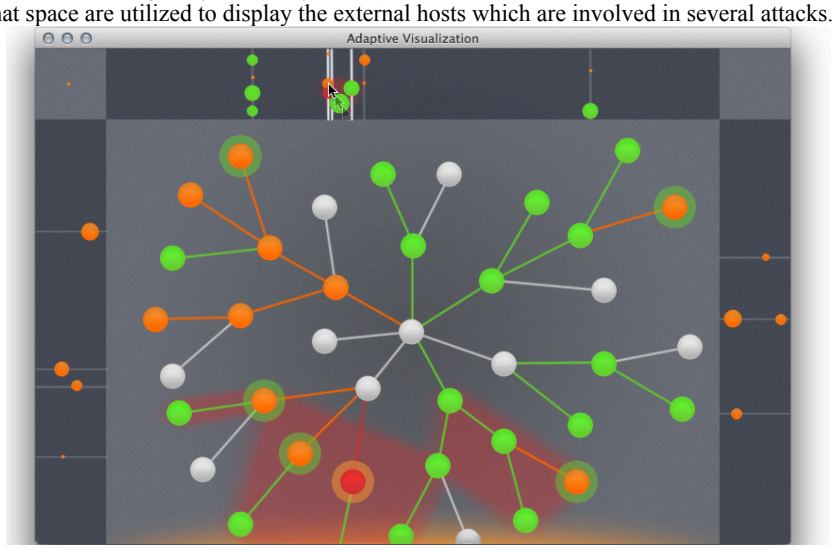

Fig. 1. Visualization system displaying the use case scenario

We decided not to explicitly present such entity as an *event* (in contrast to common event log systems representing each event as a row in their tables), firstly, because of their huge number, and, secondly, because the perception of an event as such has no effect on the sequence of actions by the operator (again, the operator should react not to the event itself, but to a breach in security likely to be resulting from the complex attack consisting of several events).

So too we decided not to explicitly display the *connections among related events* within the same complex attack in order to avoid occlusion due to the large amount of links. Instead, we highlight all hosts involved in a certain attack on operator's interactions with the system.

All events are displayed in the visualization system in *real-time*. New events displayed either through animation of the target host when it falls within the field of vision, or by highlighting the screen border beyond which the target host is hidden (as shown in Figure 1). The

latter technique engages the operator's spatial reasoning abilities and prevents his amazement at seeing a few new compromised hosts as he navigates through the network map. The animation fades with time, letting the operator perceive the order of recent events.

3.3 Interaction

Owing to the human *pre-attentive color processing* and the fact that 2D-location and color form the most separable dimension pair [22], the operator may easily perceive the hosts that require his attention. All he has to do is just select, e.g., the red hosts located in a certain area in the network map of his current interest. Generally speaking, the *highest priority breach may not be the severest attack* but the attack that targets the specific network segment (e.g., where most valuable hosts are located). Once he has determined the highest priority problem facing him, the operator moves the mouse pointer over the host of interest. As soon as this host is hovered, a *pie menu* [2] appears around it displaying the severest events related to that host (this technique stems from Fitts's law [13]).

After hovering the menu item of concern, every host related to the attack holding the chosen event becomes highlighted. The highlighting changes with hovering another element of the pie menu. Oftentimes the operator needs to compare host groups related to different attacks. Our visualization system enables operator to save up to ten selections, simply by holding the desired digit key on the keyboard while the hosts of interests are being highlighted. It will be possible to recall the saved group later by pressing the appropriate key. If the operator clicks on the selected event type, the system prompts him to make a decision based on the type of event.

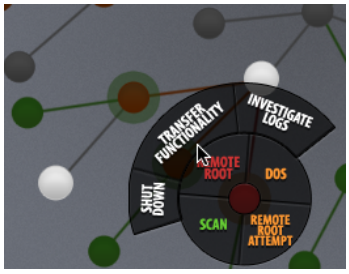

Fig. 2. Interaction with the pie menu

4 Adaptation for Continuous Monitoring

In our visualization system we have implemented the following four adaptation methods: adaptation to current task, adaptation to operator's fatigue level, adaptation to operator's cognitive overload, and adaptation to operator's experience level.

4.1 Task-Centered Adaptation

Generally speaking, any visualization system enables the operator to take only those decisions which have been predefined. It can be presumed that the operator implements his decisions by choosing which *predefined command sequence* to run with what parameters, e.g., which host (as parameter) to block (as predefined command sequence).

According to the object of action, all possible decisions may be divided into two categories — those targeting internal or external hosts. Of these, the *operations with the internal hosts* are performed via direct interaction with the network graph vertices. For example, the operator can transfer functionality of the compromised host to another host in a couple of clicks, with the system highlighting the hosts which are acceptable for the transfer (as shown in Figure 2).

In their turn, the operations with external hosts can be performed in either indirect or direct way. The *indirect operations with external hosts* are performed via the described pie menu around the internal host with an event related to the external hosts of interest (e.g., the operator can block all external hosts those have scanned certain internal host in one click). The *direct operations with external hosts* are performed via direct manipulations with the space of an external hosts. When the operator decides to block a certain group of external hosts, he just hovers one of them and the system automatically highlights proximate external hosts related to the hovered one; if the operator still needs to extend the group, he can simply hold the mouse but-

ton and move the pointer towards other external hosts, thus expanding the scope of highlighting (as shown in Figure 1).

4.2 Adaptation to Fatigue Level

The operator's psychophysical condition is taken in this paper to mean the visualization system's notion about the operator's psychophysical state based on following collected HCI data: the *number of actions per various time intervals*, *response speed* (measured by how fast the operator responds to a newly appeared menu or recently displayed event), and *accuracy* in hitting interface control elements (according to Fitts's model [5]).

The system monitors *keyboard keystrokes* and mouse actions. The latter include *mouse button presses* and *mouse movements*. E.g., the interface element is considered to be hit accurately if the mouse pointer was stopped after entering the space occupied by the element and before leaving that space. Mouse movement tracking relies on the standard Qt mechanisms and is possible by enabling the `QWidget::mouseTracking` property.

As fatigue builds up, the visualization system notices changes in the operator's interaction characteristics and adjusts the following visualization parameters: *increases size* of the control elements, *lowers saturation* of the color palette, *diminishes animation* intensity, and automatically *leads the mouse pointer* to the most significant displayed object in the nearby area.

4.3 Adaptation to Cognitive Overload

Since in practice a cognitive overload has a direct bearing on the level of fatigue, we utilize the same HCI data as mentioned above. Besides that, the visualization system is able to estimate a current *cognitive load* by the number of displayed hosts, events and control elements. Sometimes the operator may need an extra time to make a decision, not because of fatigue, but because of the vast amount of displayed information he has to absorb.

The visualization system measures the time of the operator's continuous engagement with it and estimates a likely fatigue level, recognizing the situations of cognitive overload. This done, the visualization system decreases the cognitive load by adjusting the following visualization parameters: by *zooming* the network graph (so there is fewer displayed elements), by *increasing transparency* level of insignificant hosts (e.g., those uninvolved in the most severe attacks, or the ones tagged as low-priority in the network topology setup), by *decreasing the pie menu elements*, and by *aggregating information about a subnetwork on some host* on a higher network hierarchy level and then hiding that subnetwork.

4.4 Adaptation to Experience Level

The operator's experience level is taken in this paper to be a metric of his skills in handling the system's functionality and interface. The visualization system measures that level by analyzing such collected HCI data as the *keyboard shortcuts usage rate* and *time interval between appearance of an assistance tooltip and the operator's response action*. According to these data, the system varies both the cognitive load and its level of assistance.

The visualization system has a predefined base of average operator *activity patterns* (e.g., once there is a new event, the operator hovers the related hosts in one minute if he has no other decisions to make). Whenever the current operator demonstrates a deviation from that pattern, his experience level is considered to be increased (or decreased). Thus, when the system estimates the level to be low, it provides *assistance* in the form of tooltip that points to the element of interest according to the current situation.

5 Conclusions

In this paper we have presented a security event visualization system able to recognize changes in the indicators of its interaction with the operator and adapt its functionality and presentation capabilities to those changes to bring about an improved decision making performance as to speed and accuracy. We have presented methods of the visualization system's adaptation to the current task, as well as the operator's fatigue level, cognitive overload, and experience level. We believe that these adaptation methods can go beyond security visualization to be generalized to any event visualization system and its continuously working operator. As this paper is a part of ongoing research that is on its early stage, we are not ready yet to present any evaluation numbers, but this is our next step.

References

1. Brusilovsky, P., Ahn, J. W., Dumitriu, T., & Yudelson, M. (2006, July). Adaptive Knowledge-Based Visualization for Accessing Educational Examples. In Information Visualization, 2006. IV 2006. Tenth International Conference on (pp. 142–150). IEEE.
2. Callahan, J., Hopkins, D., Weiser, M., & Shneiderman, B. (1988, May). An Empirical Comparison of Pie vs. Linear Menus. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 95–100). ACM.
3. Conti, G., Ahamad, M., & Stasko, J. (2005, July). Attacking Information Visualization System Usability Overloading and Deceiving the Human. In Proceedings of the 2005 Symposium on Usable Privacy and Security (pp. 89–100). ACM.
4. Dutta, S., McCrickard, D. S., Deshmukh, S., & Jouenne, V. (2002, June). Evaluating Benefits and Distractions of Animated Secondary Displays for Attention-Centric Primary Tasks. In 2002 International Conference on Imaging Science, Systems, and Technology (CISST'02), Las Vegas, NV.
5. Fitts P.M. (1954, June). The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. In Journal of Experimental Psychology, 47(6), 381–391.
6. Gotz, D., & Wen, Z. (2009, February). Behavior-Driven Visualization Recommendation. In Proceedings of the 14th International Conference on Intelligent User Interfaces (pp. 315–324). ACM.
7. Horvitz, E., Jacobs, A., & Hovel, D. (1999, July). Attention-Sensitive Alerting. In Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (pp. 305–313). Morgan Kaufmann Publishers Inc..
8. Huberman, B. A., & Wu, F. (2008). The Economics of Attention: Maximizing User Value in Information-Rich Environments. Advances in Complex Systems, 11(04), 487-496.
9. Karat, J., Karat, C. M., & Ukelson, J. (2000). Affordances, Motivation, and the Design of User Interfaces. Communications of the ACM, 43(8), 49-51.
10. Kohlhammer J., et al. (2011). Visual Analytic Representation of Large Datasets for Enhancing Network Security: D1.1 Analysis of Current Practices.
11. Lohse, G. L. (1997). The Role of Working Memory on Graphical Information Processing. Behaviour & Information Technology, 16(6), 297-308.
12. Long, J. (1989). Cognitive Ergonomics and Human Computer Interaction (Vol. 1). Cambridge University Press.
13. MacKenzie, I. S. (1992). Fitts' Law as a Research and Design Tool in Human-Computer Interaction. Human-Computer Interaction, 7(1), 91-139.
14. McCrickard, D. S., & Chewar, C. M. (2006). Designing Attention-Centric Notification Systems: Five HCI Challenges. Cognitive Systems: Human Cognitive Models in Systems Design, 67-89.
15. Merchant, S. (2002). Customizing the Human-Computer Interface to Compensate for Individual Cognitive Attitude: An Exploratory Study. Informing Science, 1043-1049.
16. Shiravi, H., Shiravi, A., & Ghorbani, A. A. (2012). A Survey of Visualization Systems for Network Security. Visualization and Computer Graphics, IEEE Transactions on, 18(8), 1313-1329.
17. Steichen, B., Carenini, G., & Conati, C. (2013). User-Adaptive Information Visualization — Using Eye Gaze Data to Infer Visualization Tasks and User Cognitive Abilities. In Int. Conf. on Intelligent User Interfaces.
18. Toker, D., Conati, C., Carenini, G., & Haraty, M. (2012). Towards Adaptive Information Visualization: On the Influence of User Characteristics. In User Modeling, Adaptation, and Personalization (pp. 274–285). Springer Berlin Heidelberg.
19. Toker, D., Conati, C., Steichen, B., & Carenini, G. (2013). Individual User Characteristics and Information Visualization: Connecting the Dots through Eye Tracking. In Proc. of the ACM SIGCHI Conference on Human Factors in Computing Systems,(CHI 2013).
20. Velez, M. C., Silver, D., & Tremaine, M. (2005, October). Understanding Visualization Through Spatial Ability Differences. In Visualization, 2005. VIS 05. IEEE (pp. 511–518). IEEE.
21. Vertegaal, R., Shell, J. S., Chen, D., & Mamuji, A. (2006). Designing for Augmented Attention: Towards a Framework for Attentive User Interfaces. Computers in Human Behavior, 22(4), 771-789.
22. Ware, C. (2012). Information Visualization: Perception for Design. Morgan Kaufmann Pub.
23. Yelizarov, A., & Gamayunov, D. (2009, October). Visualization of Complex Attacks and State of Attacked Network. In Visualization for Cyber Security, 2009. VizSec 2009. 6th International Workshop on (pp. 1–9). IEEE.
24. Ziemkiewicz, C., Crouser, R. J., Yauilla, A. R., Su, S. L., Ribarsky, W., & Chang, R. (2011, October). How Locus of Control Influences Compatibility With Visualization Style. In Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on (pp. 81–90). IEEE.