Using Visual Analytics to Develop Situation Awareness in Network Intrusion Detection System

Olusegun Folorunso
Department of Computer Science
University of Agriculture Abeokuta, Ogun State, Nigeria
E-mail: folorunsolusegun@yahoo.com

Adio Taofiki Akinwale
Department of Computer Science
University of Agriculture Abeokuta, Ogun State, Nigeria
E-mail: atakinwale@yahoo.com

Aderonke Justina Ikuomola
Department of Computer Science
University of Agriculture Abeokuta, Ogun State, Nigeria
E-mail: deronikng@yahoo.com

Abstract

Network Intrusion Detection System (NIDS) is a security system that monitors the network traffic and analyzes activities for possible hostile attacks. A novel collaborative visual analytics application for cognitive overloaded site security officer (SSO) in the network intrusion detection environment is presented. The system was developed for site security officers who need to analyze heterogeneous, complex intrusion under time pressure, and then make predictions and time-critical decisions rapidly and correctly under a constant influx of intrusion alert/alarm. This purpose was achieved by designing system architecture of a Treemaps Visualization on NIDs. The Treemaps Network Intrusion Detection System was implemented using the Java platform. The results of an informal usability of the network system were evaluated by the security experts in the context of Endley's three levels of situation awareness. The proposed visualization tool has some economic advantages by aiding NID's SSO to dynamically discover intrusive zone which will reduce cost of manual analysis and high risks, efficient space utilization, interactivity, comprehension and esthetics.

Keywords: Information visualization, Network intrusion detection systems, Treemap algorithm, Hierarchical data

1. Introduction

The development of computer networking has changed the stand-alone pattern of computing, but it has also increased the risk and opportunity of network intrusion. The design of secure measures to prevent unauthorized accesses to resources and data of systems becomes a very important issue in the network security domain. Network security and intrusion detection systems are one of the key research areas in the networking era as the most difficult problem today is how to deal with and rely on the huge volume of information that flows across the network while many network attacks are being reported every day. At present, it is impossible to completely eliminate the occurrences of security events, and what security faculty can do is to try their best to discover intrusions and intrusion attempts so as to take effective measures to patch the vulnerabilities and restore systems. This brought about intrusion detection (ID) and intrusion detection system (IDS).

Intrusion is defined as any set of action that attempt to compromise the integrity, confidentiality or availability of system resources (Adetunmbi *et al*, 2006). Intrusion detection is defined to be the problem of identifying individuals who are using a computer system without authorization (i.e., crackers) and those who have legitimate access to the system but are exceeding their privileges (i.e., the insider threat). Intrusion detection systems (IDSs) are deployed to protect the computer infrastructures. The classical IDSs fall into two classes – anomaly based, and misuse based. An anomaly based IDS specify the normal behaviour of users or applications and consider any pattern falling outside the defined behaviour as an attack. A misuse based IDS specifies the signatures of attacks and parses audit files to detect any matches. The metrics for evaluating IDS are false alarms (false positives) and missed alarms (false negatives). Individual IDSs are often found to be unsatisfactory with respect to either or both of the metrics. For instance, anomaly based detection can generate many false positives since deviation from the specified normal behaviour is not necessarily an attack. Also, if the definition of normal behaviour is updated at runtime, an expert intruder can slowly change her behaviour to finally include it in the definition. This may give rise to a false negative. Misuse based detection can generate many missed alarms since for most practical open systems it is very difficult to define an exhaustive attack data base. IDSs are also classified as network-based or host-based in terms of source of data. The former collect raw network packets as the data source from the network and analyze for signs of intrusions Host-based IDS operates on information collected from within an individual computer system such as operating system audit trails, C2 audit logs, and System logs (Sundaram, 1996; Byunghae-Cha *et al,* 2005).

In recent years, the problem of network intrusion detection has attracted a lot of attention in the field of network security. Network intrusions carried out in various forms, such as worms, virus, spamming, Trojan horse, and many others, pose two major threats and damage on the victims. First, the intruders probe, gather, and deduce sensitive information about target hosts in an effort to gain unauthorized access to them and their networks. Second, the intruders inject unwanted packets into the target networks, aiming to disrupt the normal communications and services provided by the target networks. It is, therefore, critically important to implement effective network intrusion detection systems (NIDSs) to monitor the network and detect the intrusions in a timely manner (Huang et al., 2009).

One of the more interesting challenges for intrusion detection in a networked environment is to track users and objects (e.g., files) as they move across the network. For example, an intruder may use several different accounts on different machines during the course of an attack. Correlating data from several independent sources, including the network itself, can aid in recognizing this type of behavior and tracking an intruder to their source. In a networked environment, an intruder may often choose to employ the interconnectivity of the computers to hide his true identity and location.

Discovering intrusion in a network environment is costly, time consuming, and high risk activity. Over the years, researchers and designers have used many techniques to build different intrusion detection systems. Despite this, there have been one or more problems with present intrusion detection systems (IDS). Some major ones include (Anderson, 1980): high number of false positives and false negatives, and lack of efficiency.

The most important source of information for system administrator or site security officer (SSO) is the output of IDS, which automatically identify potential attacks and produce descriptive alerts. Due to the complicated nature of detecting actual intrusion, most current IDS place the burden of distinguishing an actual attack from a large set of false alerts on the system administrator or site security officer. This results to significant cognitive load. It is believe that this load may be mitigated using visual analytic which takes advantage of visualization, human factors and data analysis to amplify cognition using space-constrained “treemap”. Human factors (e.g. interaction, cognition, collaboration, presentation and dissemination) play a key role in the communication between human and computer, as well as in the decision process.

This paper presents a visual analytics to develop situation awareness in Network Intrusion System by applying the hierarchical data visualization technique “Treemap”. This visualizes the hierarchical intrusive data from heterogeneous network by mapping leaf-nodes as square icons and non-leaf-nodes as rectangular border. The technique targets to represent all leaf-nodes of large scale hierarchical data in one display space and also utilize “focus-and-context” technique for proper monitoring of individual sites. This technique is useful not only for the overview of large-scale hierarchical data, but also discovery of intrusive information at local sites, and exploration of detailed attributes of intrusive workstations.

In the study, we experiment the visualization for the Activity-Relationship (AR) of a leading insurance company in Nigeria with eight zonal offices (non-leaf nodes) spread across Nigeria, and each branch has two sub-departments (leaf-nodes). All branches in each zone were networked together. The proposed technique uses all leaf-nodes of large scale hierarchical data in one display space and applies focus-and-context method for proper monitoring of individual sites.

The rest of this paper is presented as follows: related works in section 2; architecture in section 3; implementation procedure and evaluation of the design in section 4; user study in section 5; while section 6 discusses the conclusion and future works.

2. Related Works

The first generation of Intrusion Detection Systems captured network information/data into textual files. As these textual files grow, they become quite enormous and complex; making manual analysis cumbersome and unfeasible, usually resulting to undetected attacks and false alarms. Majority of the IDS used today are either rule-based or expert-system based. Their strengths depend largely on the ability of the security personnel that develops them. The former can only detect known attack types and the latter is prone to generation of false positive alarms (Adetunmbi et al., 2008). Several researchers have addressed the problem of false alarms and missed alarms with traditional IDSs which are classified as anomaly-based and misuse-based. Also, traditional IDSs often generate a very large number of alerts for practical attack scenarios. The alarms correspond to elementary goals of the attack being realized. This large volume of alarms makes it difficult for a system administrator or even an automated intrusion response system to take appropriate actions. To counteract this problem, several researchers have developed alert correlation methods to construct attack scenarios. One class of techniques combines alerts based on similarity of certain alert attributes (Apache, 2002), Valdes and Skinner (2001). For example, in Apache (2002), source and destination IP addresses and ports are used for determining similarity and graphs are drawn with links between related alerts. However, this class misses out on correlating a large set of related alerts. A second class of techniques by Cuppens and Ortalo (2000) and Dacier et al. (1996) uses training set data to determine relations between alerts.

In Cuppens and Ortalo (2000), attacks are characterized by pre-condition, post-condition, attacker actions, detection actions, and verification actions to determine if the attack succeeded. Knowing these attributes, they provide techniques to correlate alerts. However, the challenge remains to determine the attack characteristics. The most promising approach in alert correlation is demonstrated by Cuppens and Ortalo (2000) and Ning et al. (2002) which correlate alerts based on pre-conditions and post-conditions. Two alerts are correlated if the precondition of a later attack is satisfied by the post-condition of an earlier attack.

Information Visualization (IV) of IDS was needed to help site security officer in analyzing IDS outputs in order to spot attacks and possible attacks faster or hence deciding on how to respond to such attacks. Although Information Visualization seems like a natural choice for ID, however there has been little research into coupling the two, and only a single, informal user study (Yurcik, 2003) of the work of ID analysts. The limited number of efforts applying IV to the specific problem of ID usually lacked corresponding user studies to evaluate the need or effectiveness of the approach. Girardin and Brodbeck (1998) and Fortier and Shombert (2000) describe visualizations that use firewall log data to facilitate network profiling and log analysis. Erbacher et al. (2002) and Erbacher (2002) describe a 2D glyph-based visual overview of a single host and a small network, respectively. Nyarko, et al. (2002) uses input from existing IDS sensors and display that data using glyphs in 3D space. Solka et al., (2000) also utilizes an existing IDS for input and apply several known graphical techniques, such as parallel coordinate plots and circle plots, to visualize network traffic. All of these systems have made assumptions about the nature of ID work and the needs of security analysts without empirical support.

Wang and Lu (2007) developed an interactive wormhole detection and evaluation system to monitor attacks in large- scale wireless networks in real time. Aside from the research works carried out as mentioned above, IV of IDS relied on console-based IDS output usually in the form of tables.

A typical example of network log file is shown in table (1) where Network activities are captured as messages. It is often said that a picture is worth more than a thousand words. The ability to convey information in graphical/pictorial format improves communication and the assimilation of information. Research work has shown that ID specialist would prefer graphical representation of network information over text logs (Komlodi et al., 2004). While the IV systems applied to Intrusion Detection and other areas have been improved on the process of Intrusion Detection and reduced the cognitive load placed on the SSO, there are still short comings as SSO still need to analyze console tables as shown in table(1).

3. Design Methodology

3.1 System Architecture of a Treemaps Visualization on Network Intrusion Detections

Treemap-NIDS implementation is based on a Two-layered architecture. Figure (1) illustrates this two-layered architecture.

The Network Layer (Layer 1):

This layer is composed of computers (probes) at the various LANs (linked together as a WAN) which monitor the network and collects data on suspicious network activities. Data captured in this layer is routed in real time and stored in Text logs, XML Files and database (RDBMS) such as Oracle.

The Analysis Layer (Layer 2)

In this layer, data captured in the first layer in (XML, txt and RDBMS) are collated and processed by an Active Database that improves the speed at which data is analyzed. This layer extends the traditional expert system model with facilities for inferencing directly on heterogeneous database tables and monitoring events through exploiting database triggers such as deleting, updating and inserting. This layer presents the User Interface implementation of NIDS visualization using treemap.

Treemap Algorithm

Treemap algorithm is to draw treemap and to track cursor movement in the tree. The algorithms may be applied to any tree, regardless of its branching degree. Both algorithms appear on the figures 2 and 3 below. The basic drawing algorithm produces a series of nested boxes representing the structure of the tree. The cursor tracking algorithm facilitates interactive feedback about the tree. Every point in the drawing corresponds to node in the tree. While the current tracking point (from a mouse or touch screen input device) is in a node, the node is selected and information about it is displayed (Card et al ,1999), as shown in table 1.

Drawing algorithm

The treemap can be drawn during one pre-order pass through the tree in O(n) time, assuming that node properties (weight, name, etc.) have previously been computed or assigned. The drawing algorithm proceeds as follows in C-language specification:

(i) The node draws itself within its rectangular bounds according to its display properties (weight, colour, borders, etc.).

(ii) The node sets new bounds and drawing properties for each of its children, and recursively sends each child a drawing command. The bounds of a node's children form either a vertical or horizontal partitioning of the display space allocated to the node.

Tracking Algorithm

The path from the root of the tree to the node associated with a given point in the display can be found in time proportional to the depth of the node.

In our implementation, when a node draws itself it stores its bounding box in an instance variable. Every point in the tree-map corresponds to a node in the hierarchy; in addition every node is contained in the bounding box of the root node. Recall that each node's bounding box completely encloses the bounding node containing a given point thus involves only a simple descent through one path in the tree, until the smallest enclosing bounding box is found.

4. T-NIDS Implementation

Treemap algorithm in figure 2 and 3 respectfully was developed using the Java Platform. Using the Java Platform for developing the application provided the following benefits:

(i) Platform independence

(ii) As an Object Oriented language, Java gives the programmer the ability to perform system abstraction to a high degree

Traffic Treat Assessment-TTA (Probes)

Analyzing network traffic data is the basis for which any possible network compromise can be detected, hence there is a need to scrutinize network traffic in order to identify/isolate any case of intrusion. Traffic Treat Assessment is a method used to unearth intrusions, relying primarily on session data (Bejitch, 2005). The indications of suspicious or malicious traffic are usually in the form of records of network sessions within an enterprise or organization (a Leading Assurance Company in Nigeria in this case). Any activity that does not conform to the traffic patterns of normal business activity may indicate compromise.

4.1 Experimental Results

We visualized the activity-relationship (AR) of a leading Insurance company in Nigeria with eight zonal offices (non-leaf nodes) spread across Nigeria, with each branch having two sub-departments (leaf-nodes); namely, Assurance and Pensure. The network layer probes each sub network and sniff network traffic and forwarded packets in real time to a Central Oracle Database. The analysis layer identified intrusion and reports them to the NIDS SSO who monitors the heterogeneous networks at a glance (See figure 6). Figure (4) displays T-NIDS of eight insurance companies with their branches.

As shown in figure 4, the screenshot shows the top node indicating collected data for the selected eight branches of a reputable Assurance Company in Nigeria. The right pane of the application shows details of data on each node, such as the Source IP, Destination IP etc. By double-clicking on a node we could drill down to get detailed information about a particular node as shown in figure (4) using (Focus –and- context) technique. For instance, in figure 4 click a node called Lagos branch of the assurance company, we can view detail on data collected from Lagos office (LANs). This is depicted in figure 5. Similarly, figure 6 displays possible node dependent properties using different colours and their possible compromises.

Analysis of traffic data as shown in figure (6) reveals the basis for the isolation of specific sessions as possible intrusions/compromise on the network. This analysis is explained as follows:

Session Data from client host **LA-PC8 (Lagos-Personal Computer nr 8)** reveals an unusual LDAP (Light-Weight Directory Access Protocol) traffic outbound from organization.

ABJ-PC1 (Abuja - Personal Computer nr 1): shows unknown traffic initiator. Unlikely connections initiated by NAT gateway from port 22 TCP to port 1073.

ABJ-PC2 (Abuja - Personal Computer nr 2): shows unknown traffic initiator. Unlikely connections initiated by NAT gateway from port 22 TCP to port 32774.

PH-PC1 (Port – Harcourt - Personal Computer nr 1): shows request for a gif image located on an HTTP server the use of port number 3003 instead of port 80 to communicate with a HTTP server looks suspicious.

PH-PC5 (Port - Harcourt - Personal Computer nr 5): non-business related web surfing indicating possible compromise.

PH-PC7 (Port – Harcourt - Personal Computer nr 7): an analysis of this session data reveals the inability to identify server host with the IP address.

KD-PC1 (Kaduna - Personal Computer nr 1): analysis of this data reveals port number offering web-mail service that accepts usernames and passwords in the clear (without encrypting them).

KA-PC6 (Kaduna - Personal Computer nr 6): Numerous DNS IP addresses suspected, users might be creating DNS servers. Indications of poor network structured at that location (LAN).

KA-PC7 (Kaduna - Personal Computer nr 7): possible host compromise, clients accessing arbitrary SMTP services.

5. Informal Usability Evaluation: A Network Intrusion Detection System

5.1 Case Study

The nature of the IDS console and the high cost of it and SSO time precluded formal usability testing of intrusion. Intrusion alerts were evaluated via informal interviews with SSO who had used the system in the network environment. In order to perform this analysis, user comments were evaluated from interviews on five suggested areas for evaluation of visual analytics systems proposed by Scholtz (2006): situation awareness, collaboration, interaction, creativity, and utility, with particular emphasis on situation awareness. Although this analysis is necessarily subjective, it is considered appropriate for operational environments where the user cannot take time out from critical work to answer usability questions. Endsley defined situation awareness (SA) as “the perception of elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future” (Endsley, 1995; Endsley and Garland, 2000; Endsley, 1988). Endsley’s model defines three stages in the formation of situation awareness: *perception* (level 1 SA), or basic monitoring and cue detection of elements in the environment; *comprehension* (level 2 SA), the ability to interpret this information correctly and combine it with other information; and *projection* (level 3 SA), the ability to predict what happens next. The lack of situation awareness is said to be a major cause of human error and accidents in many domains (Hartel (1991) and Israel (2003)), especially those with high intrusion and potentially serious consequences of false alert.

5.2 Procedure

The subjects of the study include 10 volunteers (two females and eight males) who are research staff in Department of Computer Science of University of Agriculture, Abeokuta, Nigeria. Half of the subjects have network and security background while the other half do not. Most of the subjects do not have graphics and visualization backgrounds. All the ten volunteers have normal vision and are not colour blind. We concluded a training session before the experiment, since most of our subjects are not familiar with network intrusion detection system. The experiment is performed right after the training session.

5.3 Stimuli

The stimuli of the study are 52 independently generated network intrusion detection scenarios visualized by the proposed treemap approach. We visualized the activity-relationship (AR) of a leading insurance company in Nigeria with eight zonal offices (non-leaf nodes) spread across Nigeria, with each branch having two sub-departments (leaf-nodes); namely, Assurance and Pensure. The network layer probes each sub network and sniff network traffic and forwarded packets in real time to a central active database. The analysis layer identifies intrusion and reports them to the SSO who monitors the heterogeneous networks in a portrait using a recursive partitioning method to display intrusion at different sub-departments in each zone.

5.4 Analysis of Users' Comments

The users' comments are annotated according to the three levels of Situation Awareness (SA). These, as well as examples of each category and how it illustrates the level of SA are described below:

(i) Perception: Situation Awareness Level 1

Level 1 SA, perception, the most basic of the levels of situation awareness, is typically the easiest for a user to develop and a system to support, and often it is taken for granted. However, there were certain cases in our operating environment where it became clear that our system was fostering previously undeveloped perception of objects. This was demonstrated by the following comments:

"T-NIDS allows you to pay attention to intrusion on the networks. Before, the use of T-NIDS it was just something that happened within a twinkle of an eye. It was so difficult for the SSO to pay attention to the volume of intrusion that comes in." "With the new interface, with the intrusive activities popping up in the interface, it reminds you to pay attention; you know there are certain intrusion alert you need to respond to."

(ii) Comprehension: Situation Awareness Level 2

At level 2, the operator (SSO) is synthesizing information about the intrusion to become aware of the big picture: understanding the meaning of the information they are gathering about the intrusion and to use it to further their own goals.

Examples of this are: "They've actually been looking at the information generated by the intrusion detection system, this information is important, because they wanted to know, what the intrusion is, the source, the target, the operator were excited, nobody knows about this intrusive activities, what can we find out."

"When they see the intrusion, they become interested in learning more about the source, start looking at the other intrusion information ... they want to know more about this source."

(iii) Projection: Situation Awareness Level 3

Finally, at level 3 SA, the user can accurately predict what it going to happen next. Comments from interviews indicate that T-NIDS was quite successful in fostering this final level of situation awareness. "The T-NIDS allows the SSO to know how to respond to attacks". "Having a clear view of how the intrusion takes place in the network and how much time needed to respond to the various attack."

(iv) Evaluation Areas: Collaboration, Interaction, Creativity, Utility

Scholtz suggests four other evaluation areas for visual analytic systems: collaboration, interaction, creativity, and utility.

Collaboration

User feedback indicated that T-NIDS enhanced collaboration. Several people remarked on the fact that the tool systematized user's comments. "

Interaction

T-NIDS greatly enhanced the user's ability to interact with the information on intrusion; this also facilitated their interest and attention, and increased their levels of situation awareness. "T-NIDS gives the SSO information that either they couldn't or was too difficult to access because of the large volume of intrusion that comes in everyday. Now, the SSOs' are much more interested in exploring the techniques.

Utility

Over and over again, users commented on how useful the program was. Perhaps this reveals how much a program without clear interface can add to stress and cognitive load. The SSOs also commented on how T-NIDS improved everyone's efficiency. "Having a visual view of the intrusive activities has been extremely useful. Our main remark is that taking intrusive information is very hard and time consuming". "During the experiment, we had really good efficiency; and T-NIDS helped a lot by reducing the stress faced by the SSO." "The quick visualization of intrusions are extremely useful."

Creativity: Discovery and Decision-Making

Creativity is not normally a characteristic one would consider in developing software. However, given the challenging nature of the observing intrusion, creative solutions to problems are frequently necessary. Thus, we categorized effective decision making and discovery under creativity. Users comments indicated that Information Visualization facilitated decision-making. T-NIDS has been very useful during the experiment. The graphical view of the intrusions generated by the intrusion detection system eased the decision of respondents to intrusion on the network or to pay extra observation on the source of the intrusion.

5.5 Results and Discussions

In general, T-NIDS received very positive user feedback. In all five of Scholtz's areas of evaluation, T-NIDS received favourable reviews. In taking the users' comments as a whole we found evidence that T-NIDS has improved SSOs' situation awareness, increased overall efficiency and assisted SSOs to make collaborative decisions. Selective presentation of information, the ability to interact with the information and the use of visualizations were all effective in increasing situation awareness in this time-critical computing application.

6. Conclusions and Future Work

This paper introduced an application of visual analytics system developed for network intrusion detection system collaboratively operating a large-scale hierarchical data visualization technique. In the application, we considered T-NIDS as the best option yet in the area of network intrusion detection. Other merits of the System include dynamic addition of new network nodes that can be visualized at runtime without restarting the system. Additionally, we showed the effectiveness of a simplifying visualization in increasing situation awareness for users needing to synthesize large amounts of intrusive data and make critical decisions under time pressure. This confirms previous studies by Chittarro et al. (2007), Kim et al. (2007) and Aragon and Hearst (2005) on the value of simplifying visualizations in interfaces designed for cognitive overloaded users.

Furthermore, we have demonstrated that the cultivation and maintenance of user's situation awareness, a concept first studied in the field of aviation and cockpit management, is crucial in other fields where time-critical decisions involving large, dynamic data sets must be made.

We are sure that this system will help SSO in their day to day work, and hope that users of this technique will seek to improve on the system either as Independent System Developers.

T-NIDS is currently undergoing a lot of improvement, to incorporate a lot of features implemented in other Treemap visualization systems as illustrated in Figures 4-6 below, such as filtering criteria (e.g. drill down capacity to visualize isolated network nodes). Other features such as the use of colour and size setting to illustrate actual intrusion and false alerts, and the size of network activity respectively were also investigated.

Also dynamic addition of network nodes to the analysis is a key feature for future implementations. This will enable SSO to automatically plug in data streaming from LANs into the visualization software.

References

Adetunmbi A..O., Zhiwei S., Zhongzhi S., and Adewale O.S. (2006). Network Anomalous Intrusion Detection using Fuzzy-Bayes, in IFIP International Federation for Information Processing, 228, Intelligent Information Processing III, Eds. Shi Z., Shimohara K., Feng, D., (Boston: Springer) 525 – 530.

Adetunmbi A. O., Falaki S. O., Adewale O. S. and Alese B. K. (2008). Network Intrusion Detection Based on Rough Set and K-Nearest Neighbour. International Journal of Computing and ICT Research, 2(1). [Online] Available: http://ijcir.org/volume2-number1/article7.pdf

Anderson, J.P.(1980). Computer Security threat monitoring and surveillance, Technical report,James P. Anderson co. Box 42, Fort Washington.

Apache (2002). "Apache Chunk Buffer Overflow Attack". [Online] Available: http://httpd.apache.org/info/security_bulletin_20020617.txt

Aragon C. and Hearst M. (2005). "Improving Aviation Safety with Information Visualization: A Flight Simulation Study," CHI 2005: ACM Conference on Human Factors in Computing Systems, Portland, OR.

Bejtlich R. (2005). Extrusion Detection, *Security Monitoring for internal intrusions*, Addison Wesley.

Bruls D.M., Huizing K. and Wijk J.J. (2000). Squarified Treemaps, Proceedings of Data Visualization, 33-42.

Byunghae-Cha, K.P. And Jaittyun, S. (2005). Neural Networks Techniques for Host anomaly Intrusion Detection using Fixed Pattern Transformation. ICCSA 2005, LNCS 3481, 254-263

Card S.K., Mackinlay J.D. and Shneiderman B.(1999.) Readings in Information Visualization using Vision to Think, 152-159

Chittaro I., Zuliani F. and Carchietti F. (2007). Mobile Devices in emergency Medical Services: User Evaluation of a PDA-Based Interface for Ambulance Run Reporting. In: Loffler, J. and Klann M (eds.) Mobile Response, Berlin/Heidelberg: Springer, 19-28.

Cuppens F. and Ortalo R. (2000). “LAMBDA: A Language to Model a Database for Detection of Attacks”, In Proceedings of the Third International Workshop on the Recent Advances in Intrusion Detection (RAID’2000), Toulouse, France.

Dacier M, Deswarte Y and Kaâniche M. (1996). “Models and Tools for Quantitative Assessment of Operational Security”, in 12th International Information Security Conference (IFIP/SEC'96), (S.K. Katsikas and D. Gritzalis, Eds.), 177-186, Chapman & Hall, Samos (Greece).

Endsley M. R. (1995). "Toward a theory of situation awareness in dynamic systems," *Human Factors*, 37, 32-64.

Endsley M. R. (1988). "Situation awareness global assessment technique (SAGAT)". Proceedings of the National Aerospace and Electronics Conference.

Endsley M. R. and Garland D. J.(2000). *Situation Awareness Analysis and Measurement*: Lawrence Erlbaum Associates.

Erbacher R. F. (2002) Glyph-based generic network visualization. *Proc SPIE Conference on Visualization and Data Analysis*, 228-237.

Erbacher R., Walker, K. and Frincke D. (2002). Intrusion and misuse detection in large-scale systems. *IEEE Computer Graphics & Applications*, 1, 38-48.

Fortier, S.C. and Shombert, L.A. (2000). Network profiling and data visualization. *Proc. IEEE Systems Man and Cybernetics Society Information Assurance and Security Workshop*, 136-142.

Girardin, L. and Brodbeck, D. (1998). A visual approach for monitoring logs. *Proc. Systems Administration Conference (LISA)*, 299-308.

Hartel C., Smith K. and Prince C. (1991). "Defining aircrew coordination: searching mishaps for meaning," 6th Int'l Symposium on Aviation Psychology, Columbus, OH.

Israel R. (2003). Mercator's Projection. http://www.math.ubc.ca/~israel/m103/mercator/mercator.html, accessed 2008.

Huang C., Chang R.K.C. and Huang P. (2009). Signal Processing Applications in Network Intrusion Detection Systems. EURASIP Journal on Advances in signal Processing.

Kim S., Jang Y., Mellema A, Ebert D., and Collins T. (2007). "Visual Analytics on Mobile Devices for Emergency Response," IEEE Symposium on Visual Analytics Science and Technology, Sacramento, CA.

Komlodi A., Goodall J. R. and Lutters W.G. (2004). *Information Visualization Framework for Intrusion Detection* CHI 2004, Vienna, Austria, 1743-1746.

Ning P., Cui Y., and Reeves D.S. (2002). “Constructing Attack Scenarios through Correlation of Intrusion Alerts”, In Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS 2002), Washington D.C., 245-254.

Nyarko K., Tanya C., Scott C., and Ladeji-Osias K. (2002). Network intrusion visualization with NIVA, an intrusion detection visual analyzer with haptic integration. *Proc. 10th Symposium on Haptic interfaces for Virtual Environment and Teleoperator Systems*, 277-284.

Scholtz J. (2006). "Beyond Usability: Evaluation Aspects of Visual Analytic Environments," IEEE Symposium on Visual Analytics Science and Technology, Baltimore, MD.

Snapp S, Breatano J, Dias G.V, Goan T.L, Herberlein L.T, Che-Lin H., Levitt K.N, Mukherjee B., Smaha S.E., Grance T., Teal D.M and Mansur D. (1999). DIDS- (Distributed Intrusion Detection system)-Motivation, Architecture, and An Early Prototype.

Solka J.L., Marchette, D.J. and Wallet B.C. (2000). Statistical Visualization Methods in Intrusion Detection. *Computing Science and Statistics*.

Sundaram, A. (1996). An Introduction to Intrusion detection. [Online] Available: ftp.cerias.purdue.edu/pub/doc/intrusion_detection/Intrusion-Detection-Intro.ps.Z.

Valdes A and Skinner K. (2001). “Probabilistic alert correlation”, In Proc. of the 4th Int'l Symposium on Recent Advances in Intrusion Detection (RAID 2001), 54-68.

Wang W. and Lu A. (2007).Interactive wormhole detection and evaluation, Information Visualization (6), 3-17

Yurcik W., Barlow J., Lakkaraju K., and Haberman M. (2003). Two visual computer network security monitoring tools incorporating operator interface requirements. *CHI Workshop on HCI and Security Systems*.

Table 1. A typical network information log – Komlodi et al (2004)

Message Signature	Classification	Priority	Date	Time	Source IP	Source Port	Destination IP	Destination Port	Protocol

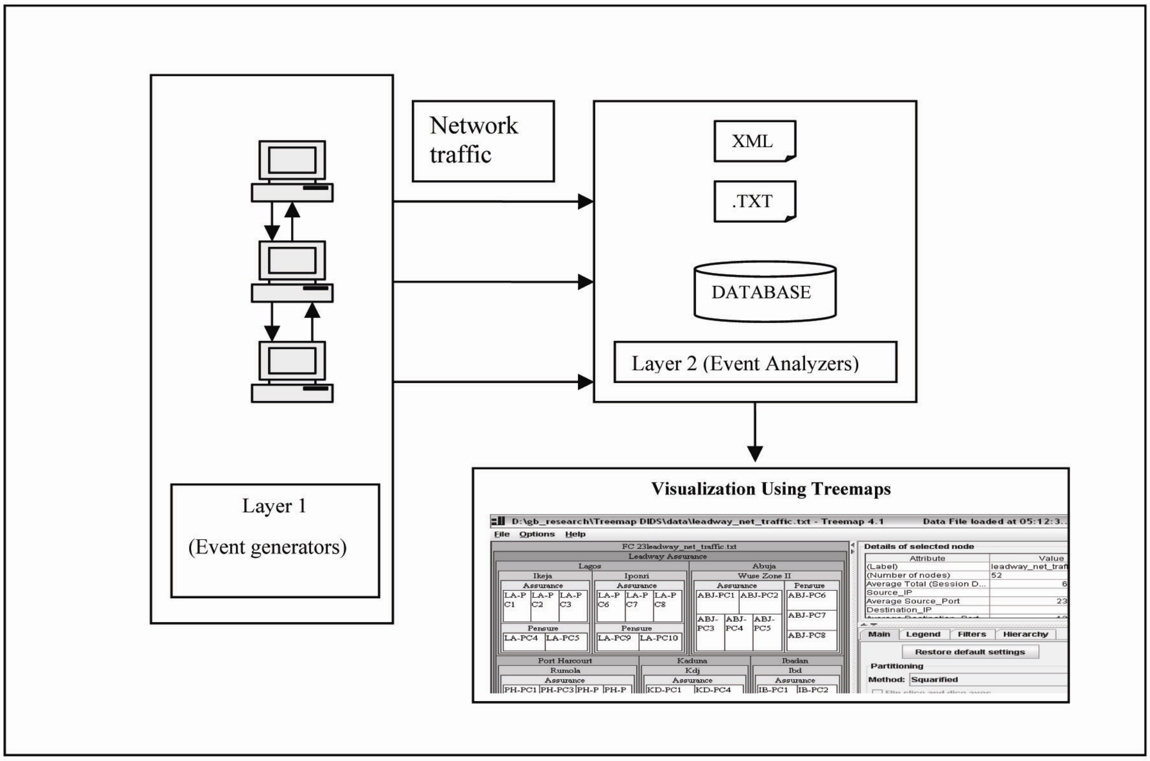


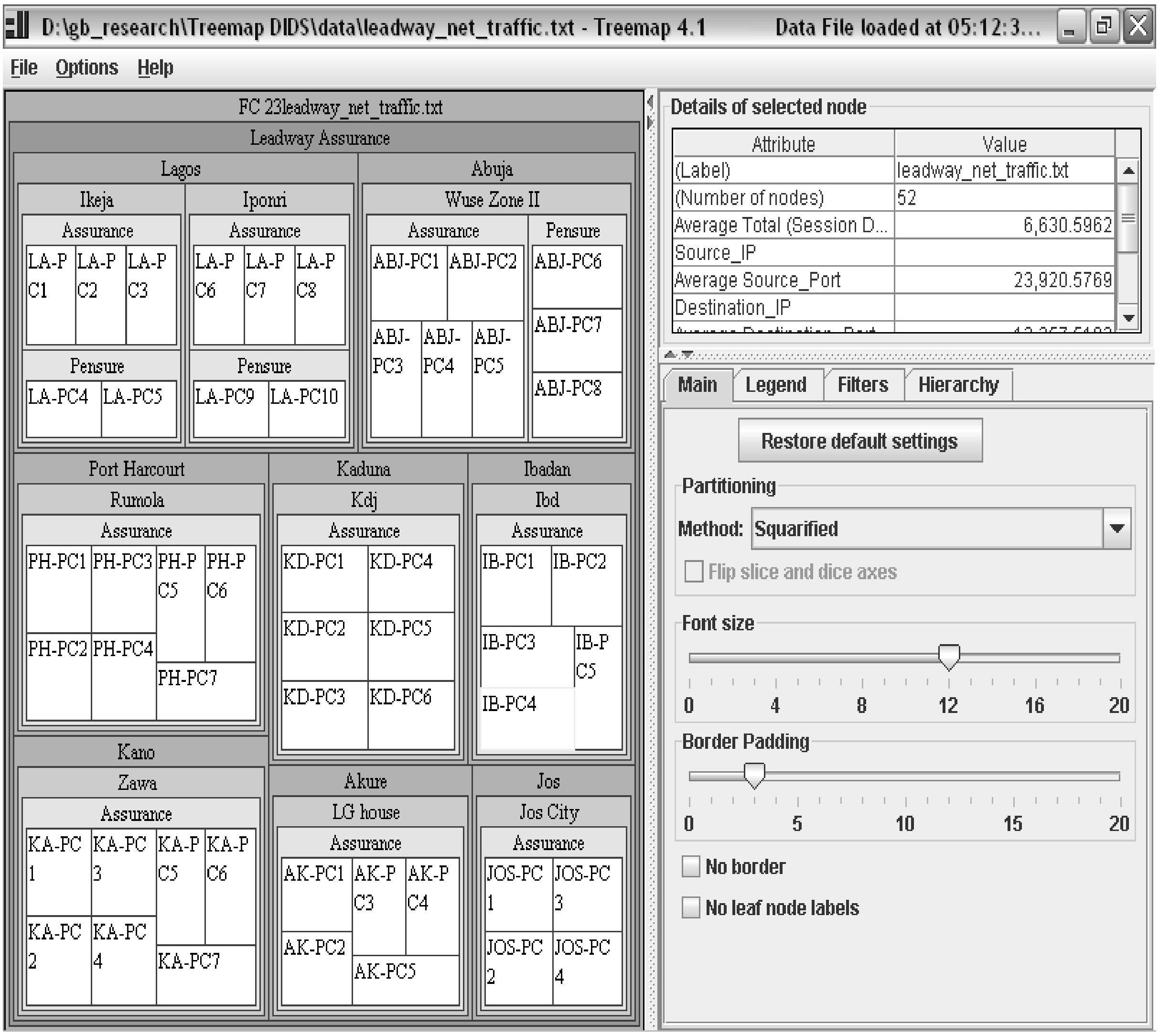
Figure 1. System Architecture of Treemaps Visualization on NIDs

```
DrawTree() The node get a message to draw itself
{donesize=0 ;
PaintDisplayRectangle();
          Switch (myOrientation) {
Case HORIZONTAL;
      StartSide=myBounds.left;
Case VERTICAL;
        startSide=myBounds.top;
}
If (myNodeType==Internal) {
        ForEach(childNode) Do {
childNode->SetBounds(startSide, doneSize, myOrientation);
ChildNode->SetVisual();
ChildNode->DrawTree ();
}}
Findpath(node)
SetBounds(startSide, doneSize, parentOrientation)
{ doneSize=doneSize+mySize ;
Switch (parentOrientation) {
Case HORIZONTAL;
myOrientation=VERTICAL;
endSide=parentWidth*doneSize/ parentSize;
SetMyRect(startSide+offset,
parentBounds.top+offset,
parentBounds.left+endside-offset,
parentBounds.bottom-offset,
parentBounds.bottom-offset);
startSide=parentBounds.left+endSide;
case VERTICAL;
myOrientation=HORIZONTAL;
endSide= parentHeight*doneSize /parentSize;
SetThisRect(parentBounds.left+offset,
startSide+offset,
ParentBounds.right-offset,
parentBounds.top+endside-offset);
startSide=parentBounds.top+endSide;
}}
} Findpath(node)
```

Figure 2. Drawing Algorithm

```
FindPath(point thePoint)
{ if node encloses the Point then
        Foreach     child of thisNode do {
           Path=FindPath(thePoint);
                If (path!= NULL) then
                         Return(Insert List(thisNode, path));
}return(NULL);
}
```

Figure 3. Tracking Algorithm



Details of selected node

Attribute	Value
(Label)	leadway_net_traffic.txt
(Number of nodes)	52
Average Total (Session D...	6,630.5962
Source_IP	
Average Source_Port	23,920.5769
Destination_IP	

Partitioning

Method: Squarified

□ Flip slice and dice axes

Font size

0 4 8 12 16 20

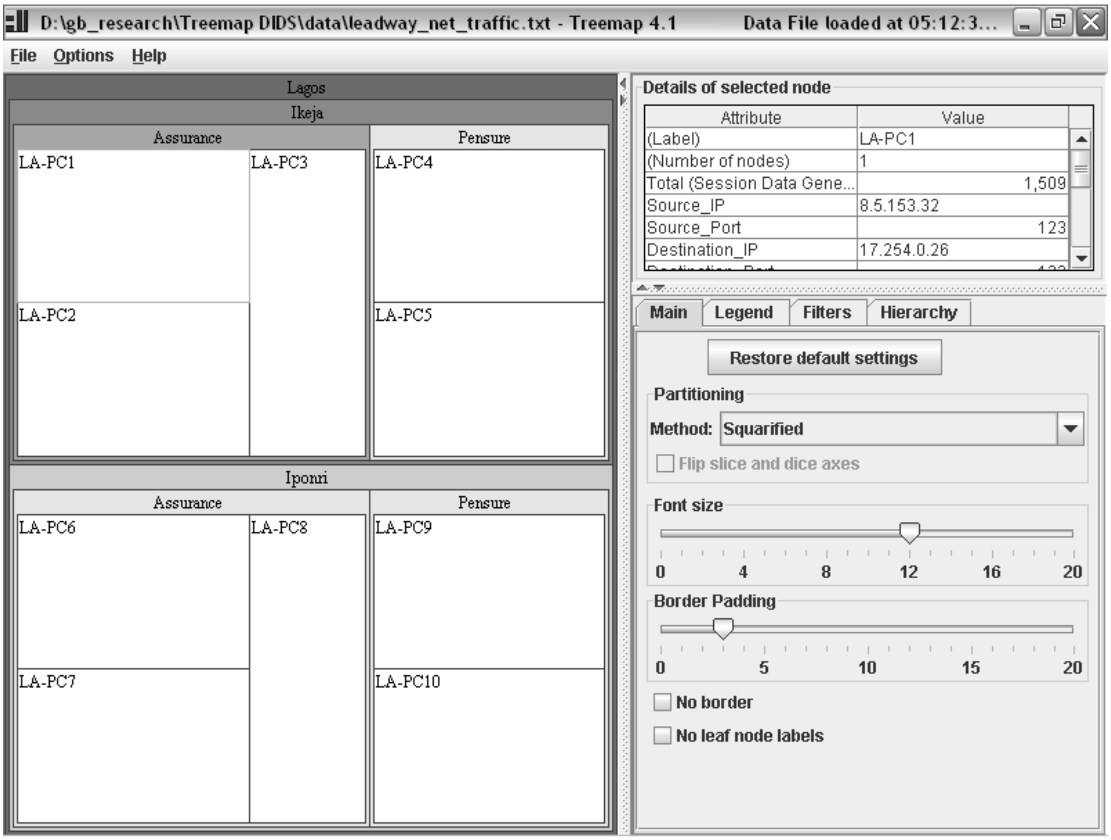
Border Padding

0 5 10 15 20

□ No border

□ No leaf node labels

Figure 4. Treemap NIDS illustrated

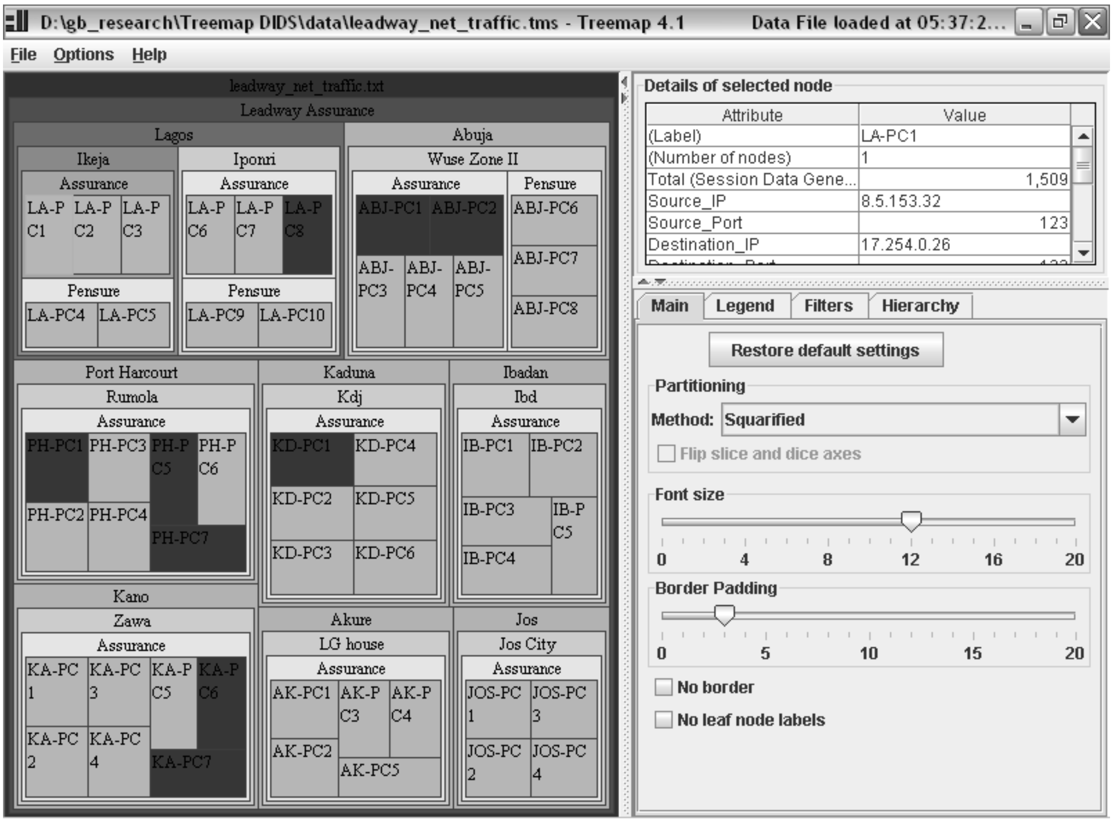


Details of selected node

Attribute	Value
(Label)	LA-PC1
(Number of nodes)	1
Total (Session Data Gene...	1,509
Source_IP	8.5.153.32
Source_Port	123
Destination_IP	17.254.0.26

Partitioning
Method: Squarified
□ Flip slice and dice axes
Font size
Border Padding
□ No border
□ No leaf node labels

Figure 5. Expanded node for the Lagos LANs



Details of selected node

Attribute	Value
(Label)	LA-PC1
(Number of nodes)	1
Total (Session Data Gene...	1,509
Source_IP	8.5.153.32
Source_Port	123
Destination_IP	17.254.0.26

Partitioning
Method: Squarified
□ Flip slice and dice axes
Font size
Border Padding
□ No border
□ No leaf node labels

Figure 6. Visualizing Odd Traffic