Visual Data Analysis for Detecting Flaws and Intruders in Computer Network Systems

Soon Tee Teoh[*] T.J. Jankun-Kelly[†] Kwan-Liu Ma[*] S. Felix Wu[*]

[*]Department of Computer Science
University of California, Davis

[†]Department of Computer Science and Engineering
Mississippi State University

Abstract

To ensure the normal operation of a large computer network system, the common practice is to constantly collect system logs and analyze the network activities for detecting anomalies. Most of the analysis methods in use today are highly automated due to the enormous size of the collected data. Conventional automated methods are largely based on statistical modeling, and some employ machine learning. In this paper, we show interactive visualization as an alternative and effective data exploration method for understanding the complex behaviors of computer network systems. We describe three log-file analysis applications, and demonstrate how the use of our visualization-centered tools can lead to the discovery of flaws and intruders in the network systems.

Keywords: information visualization, intrusion detection, visual data mining, network visualization, internet routing stability

1 Introduction

To keep computer and network systems secure and stable, it is necessary to collect vast amounts of data in order to analyze how the systems are performing dynamically. This is because no matter how rigorous the design process was for a particular system, many factors during run-time can compromise its performance. Likewise, even though network protocols may have strong theoretical bases, they may suffer security flaws and instability when actually deployed. Furthermore, most systems are not designed with perfect security. Intrusion detection and response are thus very important components of any computer system.

The above examples of run-time errors, protocol architecture weaknesses, and computer attacks are different examples of *flaws* and *intrusions*. Flaws refer to unintentional defects in the system design, human mistakes in the operation of the system, or malfunctions in the machinery. Intrusions refer to intentional illegitimate use of the system or malicious attacks on it. Whether intentional or not, such errors can cause severe damage to network systems. Collection and analysis of logs of actual activities are helpful in both detecting errors and analyzing them.

Although system logs are vital for understanding a running system, the analysis of logs to search for problems is a nontrivial task. This is because we need to discover new and unexpected knowledge to find hitherto unknown weaknesses and security flaws. The task of finding useful information by sifting through large amounts of data has spawned the field of *data mining*. Most data mining approaches are based on machine learning techniques, numerical analysis or statistical modeling. In any case, human interaction and visualization are used only minimally. Such automatic methods may miss some important features in the data.

In this paper, we describe solutions based on visual analytics, taking advantage of human intuition, visual pattern recognition and domain knowledge, as well as modern computers' processing power. We outline a visual exploration process for the analysis of logs. We also describe the principles behind using visual analytics in this application domain, and compare the visual analytics approach against existing alternative methods. We then present three examples of data analysis problems of detecting flaws and attacks, and show how the visual analytics process helps to discover new and useful knowledge in the data, leading to improvements in system security and stability. Lastly, we present updates to our previous visualization tools and show some new discoveries made by these tools. The new knowledge gained demonstrates the effectiveness of taking the visual analytics approach to uncover security and network protocol flaws.

2 Computer Security and Network Stability

The goals of computer security can be classified broadly into three categories: prevention, detection, and response. Intrusion detection plays a critical role in the security of most systems because prevention methods such as passwords and access control can be and are often compromised. Therefore, in the management of a robust system, it is necessary to incorporate some intrusion detection mechanism. Its purpose is to detect any unauthorized malicious use of the system that could compromise the availability, integrity or confidentiality of the resources. Analysis of the detected intrusion is also important so that corrective and/or punitive action may be taken. The typical approach to intrusion detection is to collect and analyze logs of user activity from which suspicious or anomalous behaviors can be scrutinized and analyzed to determine if the system has been attacked (see the "Data Mining and Visualization for Network Security" sidebar).

Network stability is closely related to computer security. Attacks over the Internet such as denial of service attacks can completely shut down the availability of the target website. In Section 4, we also discuss attacks on the Internet routing protocol itself, where an attacker can potentially usurp a router and gain access to TCP packets meant for other parties. Network administrators can also sometimes inadvertently make mistakes in router configurations, leading to serious problems in the network. In addition, a network protocol can have flaws, e.g. race conditions, that appear at run-time. Whether to detect architectural flaws, router errors, or network attacks, examining logs of the messages exchanged between the nodes in the network can help. This approach is thus the same as that taken for intrusion detection.

3 Using Visual Analytics

The guest editors of this special issue of CG&A describe *visual analytics* as "a contemporary and proven approach to combine the art of human intuition and the science of mathematical deduction to directly perceive patterns and derive knowledge and insight from

[*] {teoh,ma,wu}@cs.ucdavis.edu
[†] tjk@cse.msstate.edu

them.” This approach has been used in the past decade for various applications. For example, Ahlberg and Shneiderman [1] promote visual-based methods as a viable approach to information-seeking due to the ability of humans to recognize features in visual displays and recall related images to identify anomalies. According to Girardin [2], human perception can notice even unexpected features. From our experience, we believe that this is the biggest advantage offered by visual analytics compared to primarily algorithmic data mining methods.

Our work is based on the premise that valuable knowledge can be gleaned from large datasets; this is the premise behind the field of knowledge discovery. Most data mining methods are based on algorithms or statistical or mathematical models, often adapted from machine learning techniques. While these methods have been successful at efficiently performing tasks like classification, regression, clustering, detecting frequent itemsets and outliers, each method discovers only limited knowledge from the data.

Since visual data mining [5] is different in essence to automated methods, visual methods can discover valuable information that complements the knowledge found by more commonly-used statistical approaches. This is particularly useful in applications where the user is unsure about what needs to be discovered from the data, but only desires to explore the data to learn about the data. Many real-world problems fit into this category, such as the examples we discuss in Section 4.

The visual analytics process that we use is as follows. Starting with a large dataset, in our case a log file, an appropriate visual representation is designed, together with an intuitive interaction method. The user then interactively explores the data in order to extract knowledge. The critical part of this process is in the design of the visualization and interaction method. Good visual metaphors will bring out interesting features of the data in a way that makes sense. Note that for any application, different visual metaphors can be potentially used, and each can lead to different discoveries. Some of these discoveries lead to better understanding of the system. Other discoveries may reveal flaws and weaknesses in the system architecture, and some intruders may also be detected this way.

In the visual analytics process that we describe, there is a tight collaboration between human and computer. Human intelligence is

Data Mining and Visualization for Network Security

The use of data-mining methods for intrusion detection began with the realization that widely-used signature-based methods were too rigid to discover novel attacks. Such approaches differ in the data-mining methods they employ, as well as the data they analyze. Schultz et. al [13], for example, use Naive Bayes algorithms to detect malicious executables. Ghosh and Schwartzbard [4], on the other hand, use neural networks on the DARPA network connections dataset. Another problem with signature-based methods is the necessity for time-consuming human input. To solve this problem, Lee et al. [8] use data-mining methods to learn rules to accurately capture behavior from network connection and host session features.

Jiang et al. [6] present a pattern extraction algorithm and a method to compare the extracted patterns to find intra- and inter-pattern mismatches. An alarm is raised when there is significant deviation from normal behavior. In Mahoney and Chan’s work [9], association rules, which is another data mining technique, is applied to intrusion detection. The rich data mining techniques for finding frequent sequences are also helpful for computer security. For example, Michael [10] uses suffix trees to find frequently occurring sequences of system calls.

In contrast, much less work has been done in applying visualization to computer security. These include Erbacher et al.’s [3] work using glyphs to visual intrusion detection data, Yurcik et al.’s [15] tool for visualizing network traffic, Girardin’s [5] packet-based visualization, and Tudumi [14], a visualization system designed to monitor and audit computer logs to help detect anomalous user activities.

Visualization-based data mining methods are also few in number. One example is PBC [1], a visual classification tool. Because classification can be applied to intrusion detection, this visualization tool is particularly relevant to computer and network security. Furthermore, this work also successfully incorporates visualization and machine learning techniques to improve data mining.

Various Internet and network visualization tools exist, such as the Internet Mapping Project [2], the H3Viewer [11], and Munzner et al.’s [12] work on visualizing the global topology of the MBone. These tools focus on displaying changes in reachability and topology. Labovitz et al.’s [7] paper on Internet routing instability also includes some simple visualization of the number of Internet routing changes.

References

[1] M. Ankerst, M. Ester, and H.-P. Kriegel. Towards an effective cooperation of the user and the computer for classification. In *Proceedings of the 6th International Conference on Knowledge Discovery and Data Mining (KDD ’00)*, 2000.

[2] B. Cheswick and H. B. and. Mapping and visualizing the internet. In *Proccedings of the 2000 USENIX Annual Techincal Conference*, 2000.

[3] R. F. Erbacher, K. L. Walker, and D. A. Fincke. Intrusion and misuse detection in large-scale systems. *IEEE Computer Graphics and Applications*, 22(1):38–48, January/February 2002.

[4] A. K. Ghosh and A. Schwartzbard. A study in using neural networks for anomaly and misuse detection. In *Proceedings of the 8th USENIX Security Symposium*, 1999.

[5] L. Girardin. An eye on network intruder-administrator shootouts. In *Proceedings of the Workshop on Intrusion Detection and Network Monitoring (ID’99)*, Berkeley, CA, USA, 1999. USENIX Assoc.

[6] N. Jiang, K. Hua, and S. Sheu. Considering both intra-pattern and inter-pattern anomalies for intrusion detection. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM’02)*, 2002.

[7] C. Labovitz, G. R. Malan, and F. Jahanian. Internet routing instability. *IEEE/ACM Transactions on Networking*, 6(5):515–528, October 1998.

[8] W. Lee, S. J. Stolfo, and K. W. Mok. A data mining framework for building intrusion detection models. In *Proceedings of the IEEE Symposium on Security and Privacy*, 1999.

[9] M. V. Mahoney and P. K. Chan. Learning models of network traffic for detecting novel attacks. In *Proceedings of the third IEEE International Conference on Data Mining (ICDM’03)*, 2003.

[10] C. C. Michael. Finding the vocabulary of program behavior data for anomaly detection. In *Proceedings DISCEX ’03*, 2003.

[11] T. Munzner. Exploring large graphs in 3d hyperbolic space. *IEEE Computer Graphics and Applications*, 18(4):18–23, July/August 1998.

[12] T. Munzner, E. Hoffman, K. Claffy, and B. Fenner. Visualizing the global topology of the mbone. In *Proceedings of the 1996 IEEE Symposium on Information Visualization*, pages 85–92, 1996.

[13] M. G. Schultz, E. Eskin, E. Zadok, and S. J. Stolfo. Data mining methods for detection of new malicious executables. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2001.

[14] T. Takada and H. Koike. Tudumi: Information visualization system for monitoring and auditing computer logs. In *Proceedings of the 6th International Conference on Information Visualization*, 2002.

[15] W. Yurcik, K. Lakkaraju, J. Barlow, and J. Rosendale. A prototype tool for visual data mining of network traffic for intrusion detection. In *Proceedings of the ICDM Workshop on Data Mining for Computer Security (DMSEC’03)*, 2003.

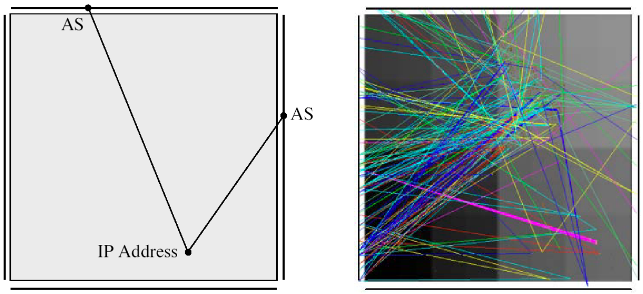

Figure 1: Visualization of OASCs. Left: Each event consists of a line connecting the affected IP prefix and ASes. Right: Actual data with color denoting event type.

used in two ways: First, in the design of the visualization and interaction techniques; and second, in the use of the interactive visualization to discover, analyze and draw conclusions from the data. On the other hand, computations are used to process and project the data onto the display, and to transform the data based on user input. In our Future Work section, we also mention incorporating more advanced machine learning techniques into the log file analysis systems.

4 Applications and Techniques

We present visual-based methods for detecting flaws and intrusions in networks. Each application examines a different problem domain and uses different visual metaphors. However, the common visual approach has led to useful insights in each case.

4.1 Anomalous Origin AS Changes

To manage packet traffic on the Internet, groups of hosts sharing portions of their IP addresses are partitioned into clusters of machines called *autonomous domains* (ASes). The problem of packet routing then simplifies to routing data between these larger entities (see "The Border Gateway Protocol" sidebar for more information). To maintain network stability, it is important that the AS associated with a group of IP addressed (their *origin AS*) actually corresponds to the router for that subnetwork. Thus, network analysts are interested in studying the dynamics of origin AS changes (OASCs) in order to distinguish normal behavior from faults or suspicious activity. Previously, we have presented a visual analysis tool to address this problem [8]. Here, we discuss an augmentation to the browsing portion of the tool and its application to a set of anomalous OASCs in 2000–2001.

The visual exploration of the OASC data is a two phase process. First, the user browses a sequence of visualizations summarizing the OASCs over time. Then, if an anomaly is discovered, the user can drill-down into the data to determine the type of anomaly and which ASes were involved. In the first stage, the user is essentially performing visual pattern matching, using the visual system to separate normal from abnormal behavior. This visual classification is based upon the rendering of OASCs for a given date. Figure 1 demonstrates our rendering method. Each IP address is mapped to a pixel using a quadtree decomposition, iteratively mapping pairs of bits from the 32-bit IP address. ASes are mapped along the four edges of the display area, and an OASC is represented by a colored line connecting the previous owner to the new owner of the AS (in O-type events, there is no previous owner). The color designates the type of change.

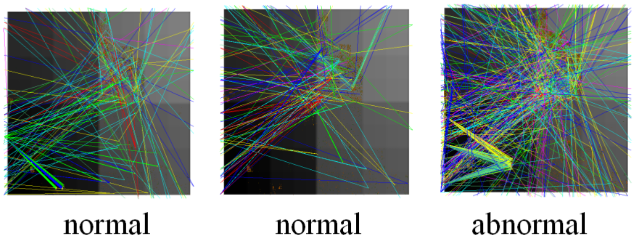

Figure 2: OASCs from three different dates. The first two are typical of normal OASCs. Looking at the display of different dates, the user gets a sense of what "normal" is. The right-most image, with its high concentration of lines from the same ASes, indicates an anomaly.

The original browsing tool was successful in separating normal from abnormal behavior in the OASC data (Figure 2). However, occlusion would occasionally mask some events. To rectify this issue, a new browser was developed. Instead of displaying a single depiction of the OASC events, one image for each change type and one displaying all the changes together are displayed. In addition, a depiction of the previous and next date's events are also shown. To manage screen real-estate, a focus+context radial layout is used [3]—the images are arranged in a circle about a larger focal image chosen from the others (Figure 3). The new layout solves the line-occlusion problem of the old browser by decomposing the event types while retaining the same mode of rapid, iterative exploration. A new date is chosen by selecting the previous or next date's image; in addition, one of the eight event type images can also be selected to become the new focus. To assist in analysis, the sectors of the circle are colored according to event type, with the combination image colored white.

A user can utilize the new event browser in several ways. For example, in our previous study, we noted several sequential CSM/CMS events; these indicate a misconfigured router and its subsequent correction. To determine how common these paired events are, a user could select either the CSM or CMS event as the focus and then explore the data (such as in Figure 3). Paired events would then be discovered by comparing the previous/next day's image for the given type (at the five or seven o'clock positions around the circle) against the image for its complementary type for the current date (at either 10 or 11 o'clock). This analysis would show that these events are very common, occurring almost once a month over the 480 days sampled.

As a further example, a user browsing the data could note some interesting behavior on March 31st, 2001 (Figure 4). The browser shows several coordinated CMS (light yellow), CMM (red), and H (blue) events. In addition, the image for April 1st (light gray) shows a similar pattern consisting of different events (CMS, CSM (cyan), and CSS (green)). It is important to note that this coordination is difficult to see in the combined image (center) and might have been missed without the other images. Using the browser, the user could then step through the subsequent dates to determine the length of the anomaly. The behavior is strongest in the first four days after March 31st, though smaller corrections continue afterwards. To find out which ASes were involved, the drill-down module described in our previous work would be used. A 3D representation of the events, using the quadtree of IP addresses as its base, AS identifiers for height, and colored cubes for events, shows two clusters of overlapping events (Figure 5), meaning that several OASCs are associated with the same IP address. Both of the clusters are in the same vertical plane, suggesting the same AS was involved, AS 703. After using the analysis tool, the user would conclude that AS 703 and 4740 claimed different portions of AS 17561's addresses (the H events), causing conflicting correcting events over the rest of day and the following days (the other events). This analysis demonstrates the strength of our visual analytic approach. For a user, noticing the correlated visual patterns without training is easy; a fully machine-based method would require a significant number of event signatures *ad hoc*.

The Border Gateway Protocol

Two of the applications discussed in this article analyze different aspects of packet routing on the Internet. It is thus useful to understand the mechanics of Internet routing. The Internet can be considered as a set of clusters, each cluster representing an organization's network. These *autonomous systems* (ASes) entirely manage any traffic within an AS cluster. To communicate between ASes, routers on the edge of an AS use the *border gateway protocol* (BGP) [6].

In BGP, each AS is assigned a unique identifier and an IP prefix mask. This mask identifies which subset of IP addresses correspond to hosts in the AS. For example, the IP prefix 128.120.0.0/16 means that every machine in the AS shares the same initial 16 bits 128.120. Using BGP, edge routers communicate network reachability information in order to properly transmit packets. These *BGP routes* consist of the destination IP prefix and a list of the ASes through which data will be routed to reach the destination. The BGP route "128.120.0.0/16: (7, 23, 92)," for example, means that packets for the IP prefix 128.120.0.0/16 would need to sequentially pass through AS 7 and AS 23 before reaching the AS representing their destination (AS 92). The AS responsible for an IP prefix is known as the prefix's *origin AS*.

Two aspects of BGP routing information are visualized in this article: Origin AS changes (OASCs) and BGP route changes. In the former, we depict the changes to origin ASes over time. An origin AS can change due to a change in the ownership of its IP prefix, valid network operation, network faults, or attacks. In the latter two cases, this change would force the delivery of packets to a wrong AS. An OASC event consists of the IP prefix affected, a list of ASes associated with the change (generally the new origin AS for the prefix), the date of the change, and the type of change. A change can either narrow the mask of addresses an AS already owns (a B-type) or another AS owns (an H-type), claim ownership of another AS's prefix (C-types), or clam ownership of an unowned prefix (O-types). The last two changes are further classified depending on whether a single AS or multiple ASes claim ownership of the prefix. As only one AS should claim ownership of a prefix, multiple origin AS conflicts indicate faults or attacks [7]. Some OASCs are complementary: A CMS event (a C-type change from multiple ASes to a single origin AS) could correct a CSM event (a C-type change from a single AS to multiple origin ASes). There are eight OASC types overall (OS, OM, CSM, CMS, CMM, CSS, H, and B); each is visualized in the tool described here.

In the second visualization, we examine BGP route dynamics. As a host's availability changes, either an AS along a route or the the origin AS could become unavailable. Whenever routing information changes, edge routers exchange *BGP announcements*. An announcement is either a new BGP route or a *withdrawal* event such as "128.120.0.0/16: WD;" in this example, the IP prefix 128.120.0.0/16 has become unavailable. From a sequence of BGP announcements, the behavior of Internet routing can be observed. Instabilities in routing can cause serious disruptions in network traffic. Previous studies have identified three forms of *instability events*: *slow convergence* to a stable path, *oscillations* between different paths, and *repeats* of the same path [1, 2, 3, 5]. The visualization discussed in this work aim to reduce the time to discover and analyze these events.

The routing data for this article was obtained from the Oregon Route Views server [4]. The data consists of BGP events over 480 days in 2000 and 2001.

References

[1] L. Gao and J. Rexford. Stable internet routing without global coordination. In *Proceedings of ACM SIGMETRICS*, pages 307–317, 2000.

[2] T. Griffin and G. Wilfong. An analysis of BGP convergence properties. In *Proceedings of ACM SIGCOMM*, pages 277–288, 1999.

[3] T. Griffin and G. Wilfong. A safe path vector protocol. In *Proceedings of IEEE INFOCOM*, pages 490–499, 2000.

[4] D. Myer. University of Oregon route views project, 2002. http://www.antc.uoregon.edu/route-views/.

[5] D. Pei, X. Zhao, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. Improving BGP convergence through consistency assertions. In *Proceedings of IEEE INFOCOM*, pages 902–911, 2002.

[6] Y. Rekhter and T. Li. A border gateway protocol 4 (BGP-4). Technical Report RFC 1771, The Internet Engineering Task Force, 1995.

[7] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang. An analysis of BGP multiple origin AS (MOAS) conflicts. In *SIGCOMM Internet Measurement Workshop*, 2001.

4.2 Routing Instability

The Internet is a complex distributed system running on a very large number of nodes using various protocols. The study of the operational behavior of the Internet is a fundamental and important part of the effort to make the Internet more stable, robust, and secure. In the past, monitoring and analyzing network behavior has been mainly done by browsing the raw data or looking at some simple plots of statistical analysis results. We have developed a suite of visualization techniques and formulated a sequence of steps utilizing them for improved understanding of Internet routing data, and this has been described in our previous work [7]. In this paper, we focus on the use of the system to detect and analyze a problem in the routing to the Google IP address. The discovery of this problem is mentioned in our previous work [7], but in this paper we describe further visual-aided analysis and investigation to find out the root cause of the observed problem.

The first indication of the problem is observed from the EventShrubs module, where each instability event is shown as a circle on a time-line. The circle has segments colored according to the instability type matching the event, such as *oscillation* and *repeat*. The size of the circle is determined by the number of BGP update messages included in the event. Obviously, the more the number of messages in a short period of time, the more unstable routing is, and the user's attention is immediately drawn to large circles. The height of each shrub has no inherent meaning; different heights are used simply to prevent shrubs from occluding one another. However, the presence of very tall shrubs does indicate that there are many different instability events around that period of time. Therefore, this provides an additional visual cue.

Figure 6 shows the EventShrubs visualization of the routing instability events of the Google IP address for the year 2001. You can see that a very severe instability event occurred around July of the year.

The next step naturally is to investigate what caused this problem. Using the browsing feature available in the visualization system, the user quickly locates the period of instability and looks at the text visualization of the update messages. This is shown in Figure 7.

To investigate even further, the user looks at the BGP update messages from a different peer. From the point of view of the obser-

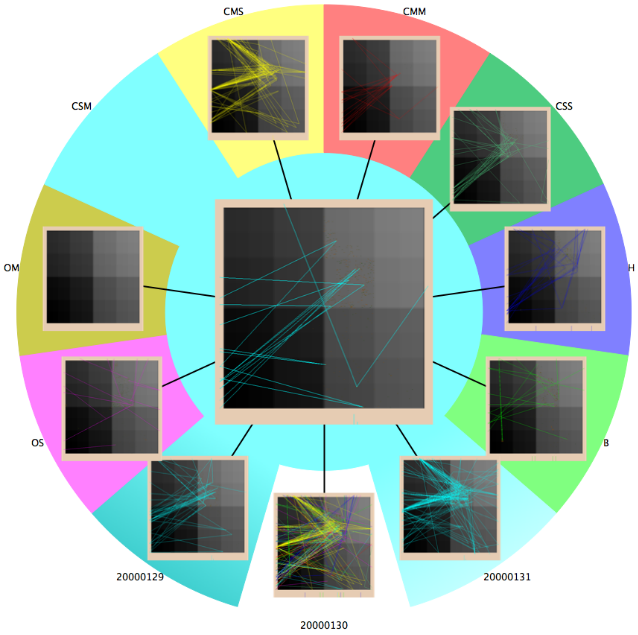

Figure 3: The new OASC browser. The eight change types are arranged radially about the center in addition to an image showing all events and images for the neighboring dates. User selection either changes the focus type or navigates through the dates. In this example, the CSM events for January 30th, 2000 are the focus.

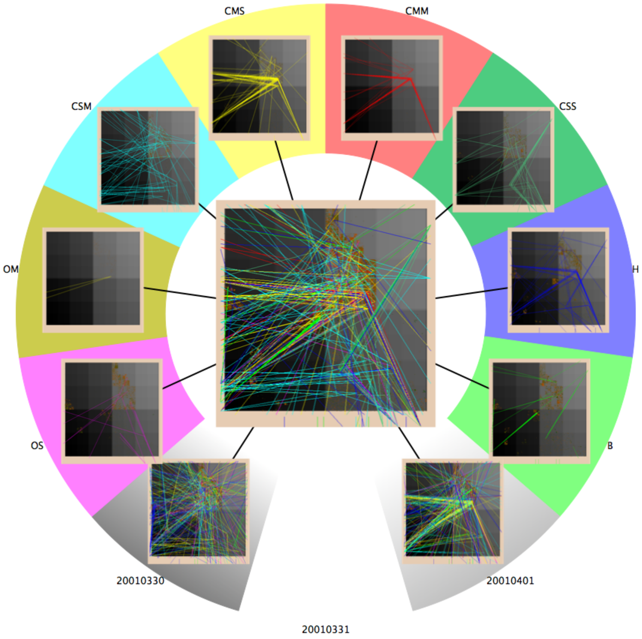

Figure 4: Correlated anomalous OASCs on March 31st, 2001.

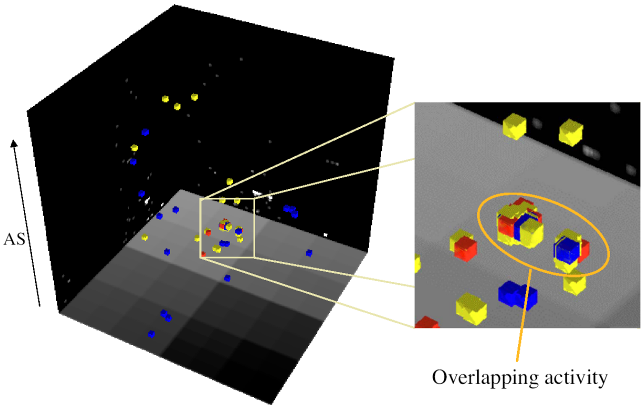

Figure 5: Further investigation into the March 31 anomaly. The two clusters involve AS 703, one of the ASes at fault.

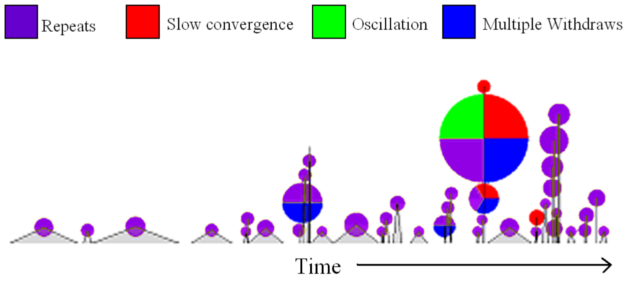

Figure 6: EventShrubs visualization of the instability periods of the route path to the Google IP address shows one instance of severe instability around July 2001.

vation point from which we're collecting the data, the observation point is connected to multiple peers, which are nodes that have a direct connection to the observation point. Each of these peers has a path to the destination IP address, in this case Google. The visual displays from Figures 6 and 7 show the messages from peer AS 2914. The user now looks at the messages from another peer, AS 3333. From AS 3333, there is no corresponding instability in this same time period. Instead, there is a long "Withdraw" covering this period of instability. A "Withdraw" message means that no path is available to the IP address. A visual comparison can be performed by telling the visualization program to display the messages from the two peers side-by-side (Figure 8). The most likely explanation is that the Google server was experiencing some major problems at that time, sporadically working and shutting down. This caused very frequent changes in its availability, leading to many BGP announcements and withdrawals. These messages are filtered out by an AS along the path from AS 3333 due to damping. The problems highlighted by the visual analysis are the problem on the Google server on that day, as well as the lack of damping along the path from AS 2914.

4.3 Intrusion detection

Detecting intruders in a computer network system is often performed by the examination of logs of user activity, system calls or network connections. In our work, we allow the user to interactively explore logs so that intruders to computer network systems can be detected.

Intrusion detection can be treated as a form of the event classification problem. In a classification problem, each object is defined by its attribute values in a multi-dimensional space; furthermore each object belongs to one class among a set of classes. The task is to predict, for each object whose class is unknown, which class the object belongs to. Typically, a classification system is first trained with a set of data whose attribute values and classes are both known. Once the system has built a model based on the training, it is used to assign a class to each unclassified object. A classification-based intrusion detection system therefore takes samples of network connections labeled as "attack" or "normal". These samples are then used to construct a model so that any future connection can be compared with the model, and thus attacks can be detected. We describe how visualization is used not only for classification, but also for anomaly detection and cluster analysis, and how these tasks help in the intrusion detection effort.

4.3.1 Visual Classification

To display the data, we use PaintingClass [6], a user-directed visual classification program we previously designed. PaintingClass is based on decision trees, a popular and well-known classification approach. A decision tree classifier constructs a decision tree by recursively partitioning the dataset into disjoint subsets. One class is assigned to each leaf of the decision tree.

Interactive decision tree construction starts by visualizing the training set consisting of connections whose attribute values and class are known. In the root of the decision tree, every object in the training set is projected and displayed visually. Each non-terminal node in the decision tree is associated with a *projection*, which is a mapping from multi-dimensional space into two-dimensional display. The projection method used is Star Coordinates [4], where the position of a point representing a data object is given by ($\Sigma_{i=0}^{n}$ a_i x_i , $\Sigma_{i=0}^{n}$ a_i y_i), where a_i refers to the data objects attribute value in dimension i, and (x_i , y_i) are the screen coordinates of the endpoint of axis i. In other words, the position of a point is influenced by its attribute value in each dimension and the user-defined axis position of that dimension. By moving the axes in a Star Coordi-

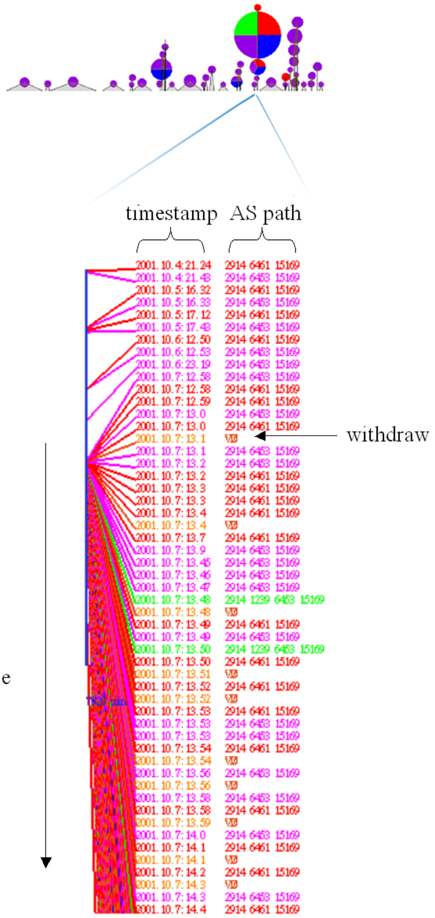

Figure 7: Visualization of the update messages from Peer AS-2914 during the Google instability event. Each update message is written horizontally across, with a line drawn to the vertical time-line to indicate the time the message was received.

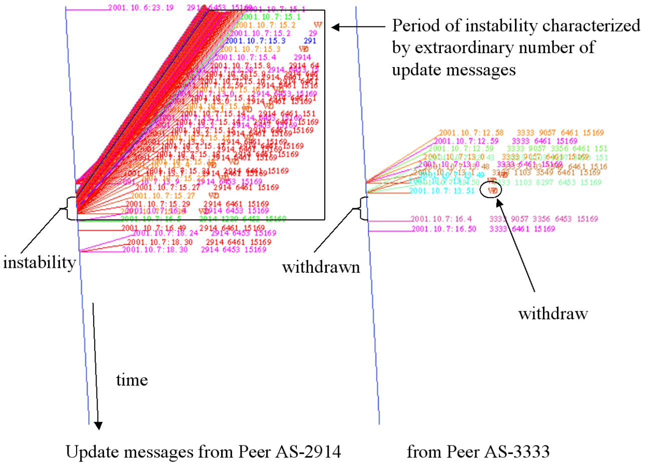

Figure 8: Visualization of the update messages from Peer AS-2914 together with Peer AS-3333 shows that the instability was filtered out by damping along the AS path from AS-3333 to the Origin AS. The lack of damping from AS-2914 is one of the problems discovered by the visual analysis.

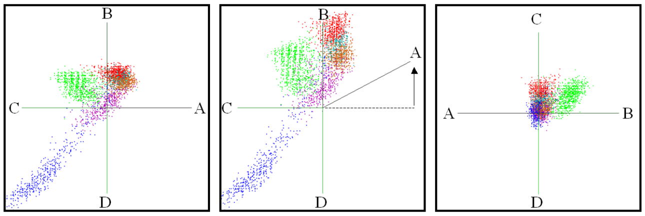

Figure 9: Left: Star Coordinates projection. The four dimensions used are represented as axes labeled A through D. Each data object is projected to one point on the display. Middle: The user moves the A axis with the mouse. The projected points move accordingly. Right: A projection with a different assignment of the axis yields a significantly different picture.

nates projection, the user creates a projection that best separates the data objects belonging to different classes. The position of each data object is therefore determined by the projection, and its color is assigned according to the class it belongs to (Figure 9).

Each projection is then partitioned by the user into *regions* by *painting*. Painting is performed by left clicking and dragging the mouse cursor over the screen. The path traveled by the cursor will be colored with the selected color. Next, for each region in the projection, the user can choose to re-project it, forming a new node. In other words, the user creates a projection for this new node in a way that best separates the data objects in the region leading to this node.

For every new node formed, the user has the option of partitioning its associated projection into regions. The user recursively creates new projections/nodes until a satisfactory decision tree has been constructed. Each projection thus corresponds to a non-terminal node in the decision tree, and each un-projected region thus corresponds to a terminal node. In this way, for each non-root node, only the objects projecting onto the chain of regions leading to the node are projected and displayed. PaintingClass displays the decision tree according to the schematic in Figure 10, and allows the user to switch the focus to different projections, move the Star Coordinates axes, paint regions, and build the decision tree.

In the classification step, each object to be classified is projected starting from the root of the decision tree, following the region-projection edges down to an un-projected region, which is a terminal node (ie. a leaf) of the decision tree. The class which has the most training set objects projecting to this terminal region is predicted for the object.

We applied PaintingClass to the KDD Cup '99 Intrusion Detection dataset. Each object in this dataset is a network connection. Each object is defined in 41-dimensional space, and belongs to one class among five possible classes: *normal*, *probe*, *DOS*, *U2R* and *R2L*. The *normal* class indicates that the connection is a harmless normal connection, whereas the other four classes are different types of attacks. There are 494,021 connections in the training set, and 311,029 in the test data. A detailed description of the dataset can be found at: http://kdd.ics.uci.edu/databases/kddcup99/task.html The KDD Cup '99 dataset is the only large-scale, publicly available data for evaluating intrusion detection tools.

We visualize the KDD Cup training set, where each connection is labeled as one of the five classes, and interactively construct a visual decision tree (Figure 11), through the painting of regions and specification of projections. This decision tree is then used to classify the test data, whose class labels are omitted; the results are discussed in Section 4.3.3.

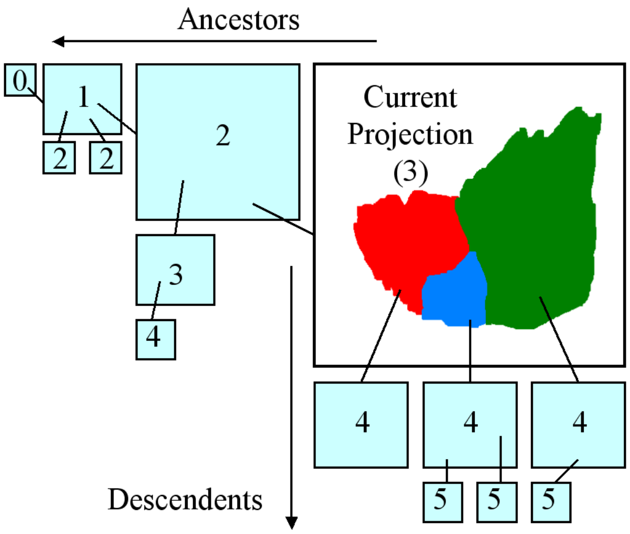

Figure 10: PaintingClass decision tree visualization layout. The current projection (ie. projection in focus) is drawn as the largest square in the upper right corner. In this figure, each projection is labeled by its distance to the root. In this example, the current projection has three regions, and each has been re-projected to a child projection.

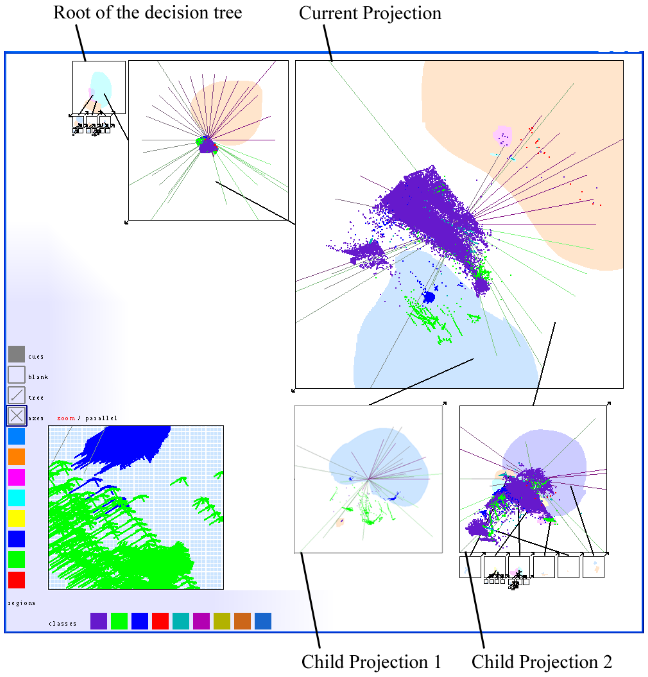

Figure 11: PaintingClass visualization of a decision tree constructed of the KDD Cup '99 Intrusion Detection dataset. In the current projection, the blue region is re-projected to Child Projection 1 and the white region is re-projected to Child Projection 2. In Child Projection 1, the user has adjusted the axes so that the objects of the blue and green classes are better separated than in the parent projection. The regions in Child Projection 1 are not re-projected, and so form the leaves of the decision tree.

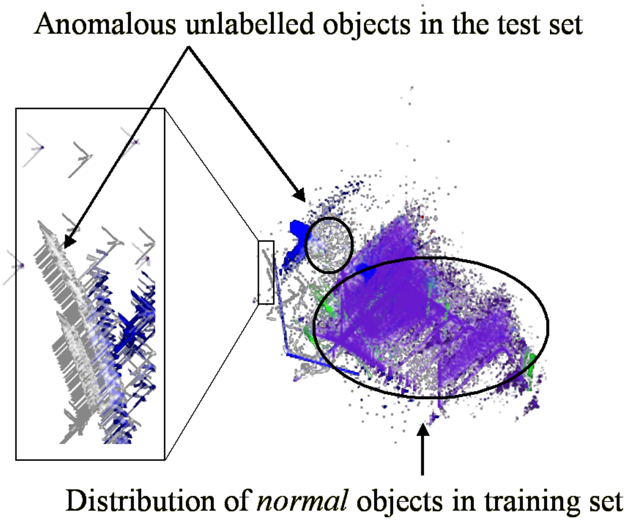

Figure 12: The normal objects in the training set are displayed as purple points. The grayscale objects belong to the un-classified test set. Two clusters of test set data are highlighted. The absence of normal data in the training set in these clusters indicate that they are intrusions.

4.3.2 Anomaly Detection

While standard decision tree classification is effective for intrusion detection, anomaly detection can complement such classification. Given examples of only normal activity, an anomaly detector creates a model of what constitutes normal activity. When it is subsequently given new unlabeled activity, the anomaly detector will compare it against the normal pattern and flag the new activity as an anomaly if it deviates from the normal pattern. The advantage of anomaly detection lies in discovering types of attacks that are previously unknown.

To use the visual intrusion detection system for anomaly detection, the connections to be classified are displayed together with the training set. For the training set, each object is colored according to the class it belongs to. The data to be classified are colored gray . For testing purposes, the objects in the test set are colored gray, as they would be if they were new connections to be classified as intrusion or normal. It is very important not to reveal the class labels of the test set objects, because we are simulating the visualization of as yet un-classified connections.

With the option of displaying objects from the test set (with class labels omitted) together with objects from the training set, the user can now identify regions where the density distribution in the two sets are different. In particular, the user is interested in finding regions where the training set has a low density of normal data while a high-density cluster exists in the test set. Such a region is considered an anomaly because the density distribution deviates from normal. An example is shown in Figure 12. Using PaintingClass, the user paints and labels such a region as one of the attack classes based on neighborhood information, and all test objects projecting to the region will be predicted to belong to the labeled class.

4.3.3 Accuracy

We used our visual intrusion detection system (IDS) to classify the KDD Cup ’99 test data to measure the accuracy of our approach. First, the visual IDS was used to visualize only the training data, and a decision tree classifier was built by the user. This corresponds to using PaintingClass as purely as a visual classification tool. Second, test data (with class labels omitted) was visualized together with training data, and a different decision tree was constructed by the user. This corresponds to using PaintingClass as both a classifier and an anomaly detector.

Table 1: Cost score of visual IDS compared with top 5 (out of 24) KDD Cup ’99 Entries. The lower the cost the better.

Visual classification and anomaly detection	0.2087
KDD Cup First Place	0.2331
KDD Cup Second Place	0.2356
KDD Cup Third Place	0.2367
KDD Cup Fourth Place	0.2411
KDD Cup Fifth Place	0.2414

The scoring system used to judge the entries of the contest is described at http://www.cs.ucsd.edu/users/elkan/clresults.html. It is a cost-based system. The objective of a classifier is to minimize the cost. The results are also presented in the same website.

Table 1 shows the cost score of the top 5 entries to the contest. Visualizing only the training data, using the visual IDS incurred a cost of 0.2551, which would place it in the middle among all 24 entries. When anomaly detection was incorporated into the visual IDS, the cost improved to 0.2087. This is significantly better than the winning entry of the contest, considering that the threshold for statistical significance used in judging the contest is 0.0060. Comparing our present results with the actual participants of a past contest certainly does not indicate the superiority of our method. Our intention is merely to show that visual classification and anomaly detection does indeed work.

4.3.4 From Patterns to Knowledge

What is perhaps the most important contribution of visual intrusion detection is that besides being very accurate, some new patterns were observed during interactive use of the visualization system. These patterns indicate important properties of the data, and were unlikely to have been discovered by automatic classification methods.

Some patterns observed in the visualization of the data caught the attention of the user. For example, Figure 13 shows an L-shaped cluster of denial-of-service (DOS) attack connections (blue). Such a pattern leads the user to question whether it represents a cluster in a higher-dimension space.

We created an interactive visual mechanism to allow the user to analyze such patterns in higher-dimensional space. Once the user observes a pattern, the user paints on the pattern. The user then specifies the class to be examined, and clicks on the “Correlate” button. The program makes a list of all objects belonging to the specified Class and projecting to the painted region. From this list, the program finds the minimum and maximum values of the screen x and y coordinates. The coordinate that has the larger range (maximum - minimum) is called the *ScreenCoordinate*. The program then displays a table of plots, one plot for each dimension in the original attribute space. In each plot, each object is displayed as a point whose y coordinate is determined by its attribute value in that dimension. The x-coordinate is determined by the ScreenCoordinate.

Figure 13 shows an example of a table of plots. The user observes that most points follow a trend in each plot. For example, in some dimensions, the attribute values of the objects are constant, in others, they vary linearly with ScreenCoordinate. The user thus hypothesizes that the trends observed describe a cluster, and that the points which are anomalous do not belong to the cluster. The user paints on these points and clicks on the “Remove” button. The objects represented by these points are removed, and only the remaining points are labeled as belonging to this cluster.

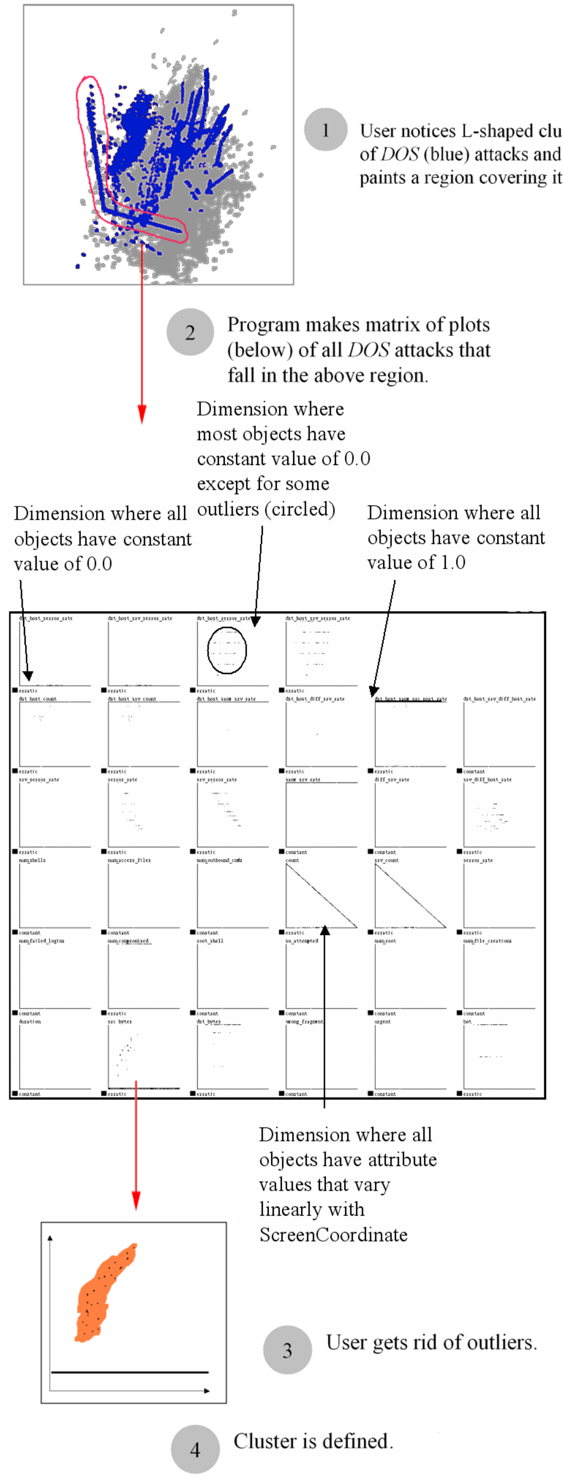

Figure 13: The visual cluster definition process. Top: DOS attack connections are shown in blue, all other connections are colored gray. A big L-shaped cluster is observed. The user paints a region covering this L-shape. Middle: Table of plots of individual dimensions against *ScreenCoordinate*. The user notices many dimensions that are mostly constant except for a few outliers. Bottom: These outliers are then be removed by painting. The remaining points are considered to belong to the cluster being defined.

In this way, clusters of attack types can be defined precisely. Each constant dimension d_i results in an equation $d_i = c$. Each pair of linearly varying dimensions d_i and d_j results in an equation d_i = k × d_j + c. For example, the L-shaped DOS cluster shown in Figure 13 can be defined as a 25-dimension cluster. This cluster contains the entire L-shaped cluster, which consists of 280,795 objects, with 3,649 false positives. (A false positive is a non-DOS object wrongly classified as a DOS object.)

Note that if additional dimensions such as *srv_count* and *count* are taken into consideration, the two parts of the L-shape each becomes a separate cluster. This is clearly visible by painting the entire L-shape and viewing each dimension in the matrix of plots. The lower-dimensional description is therefore a generalization of the two higher-dimensional clusters, and represents common characteristics of the two clusters. This "signature" thus can be used to detect new variants of this attack type. In our use of the visual tools, lower-dimensional generalizations of other clusters were also found.

5 Future Work

In each of the three applications presented, even though we have made some key discoveries from the visualization, there is other useful information in the datasets to be uncovered. For example, many OASC events remain to be analyzed. Although our visual representations have revealed some patterns, additional algorithmic filtering and processing may reveal other patterns, such as the correlation between different days, or temporal patterns pertaining to specific ASes or IP prefixes. Such patterns, if exist, are not easily observable using current visualization. Similarly, machine learning algorithms can be applied to routing instability analysis to generate better definitions of instabilities. They can also be applied to intrusion detection to guide the user in finding "good" projections. We are thus working on how to efficiently combine interactive visual methods with algorithmic methods to make further analyses and discoveries.

The cluster definitions found in Section 4.3.4 can be analyzed to extract characteristics of different attack types. We also begin to look at the problem of real-time BGP monitoring, as well as real-time deployment of the visual IDS.

6 Conclusions

We have demonstrated the usefulness of visual analytics in discovering flaws and intrusions from exploring logs. First, router misconfiguration was discovered by visualizing Origin AS changes. Second, lack of damping and unstable network availability was discovered by visualizing and analyzing routing update messages. Third, visualization was used not only to successfully classify connections as attack or normal, but also to reveal patterns and signatures in the attacks.

All three applications require the design of appropriate visualization tools and interaction techniques. Because of the difference in the nature of the data, the task, and the desired knowledge, the visual metaphors used in each of the applications are very different from one another. In fact, different visualization tools are used in the same application.

It is important to point out is that this knowledge is not easily obtainable through conventional statistical or algorithmic analyses. The use of visual analytics in exploring logs is particularly effective because humans are good at discovering unexpected patterns, whereas automatic methods are better suited to performing a specific well-defined task. For example, the observation of clusters such as the example shown in Figure 13 is completely unexpected.

Before visualization, we were expecting to see attack clusters having Gaussian-like distribution. Only after visual analysis did we find that many of these clusters have clearly-defined characteristics in the form of $d_i = c$. Without prior knowledge, standard clustering algorithms such as k-means would not have discovered such cluster characteristics.

Another advantage of visual analytics is in the analysis of the data to find the cause or explanation for an observed phenomenon. The example shown in this paper is the explanation found for the Google BGP routing instability. Visual analytics is advantageous when the question is open-ended, meaning that the range of possible explanations is unknown beforehand. If the possible explanations could be enumerated exhaustively, then automated methods can easily test the signatures of each of the possibilities. However, in cases where this is not possible, such as in our Google example, then it can be useful for the user to visually explore the data, look at the patterns, and use domain knowledge to help think of the explanation.

One characteristic of information visualization is that there is no inherent visual spatial coordinates of the data being visualized. Therefore, the mapping from data attributes to visual attributes is unrestricted. Different mappings can possibly be used. The key is to find an appropriate mapping that has the potential to lead to discoveries. The success in finding patterns in our three examples shows that the mappings we have chosen make sense. However, we would not be surprised to find that other mappings can lead to more discoveries.

The process of visual analytics is a continuous loop between interactive visualization and knowledge discovery (and perhaps even back to data collection/preparation, but not in our example applications). Starting with a dataset and analysis goals, a visualization is designed. From the interactive exploration of the dataset, the user gains previously unknown knowledge about the data. With this new knowledge, the user may be prompted to ask further questions, necessitating further analysis using existing visual tools, or requiring new tools to be developed. This loop continues until the user is satisfied with the information acquired. Our example applications illustrate this process very well, and this is shown in Figure 14. This figure shows two examples, one from intrusion detection, the other from routing instability. Each example has two iterations of the visual analytics loop.

It is not our intention to replace conventional machine learning methods with visual data mining. Just as different machine learning methods can be used together to complement one another in discovering different knowledge in particular datasets, the addition of visualization to the suite of data mining methods employed can help discover patterns missed by other methods. Therefore, the full power of visual analytics for log-file analysis remains to be exploited.

7 Acknowledgments

This work has been sponsored in part by the U.S. National Science Foundation under contracts ACI 9983641 (PECASE award), ACI 0222991, and ACI 0220147 (ITR). The authors would also like to thank Ke Zhang for preparing some of the data files.

References

[1] C. Ahlberg and B. Shneiderman. Visual information seeking: Tight coupling of dynamic query filters with starfield displays. In *Proceedings CHI'94: Human Factors in Computing Systems*, pages 313–317, 1994.

[2] L. Girardin. An eye on network intruder-administrator shootouts. In *Proceedings of the Workshop on Intrusion Detection and Network Monitoring (ID'99)*, Berkeley, CA, USA, 1999. USENIX Assoc.

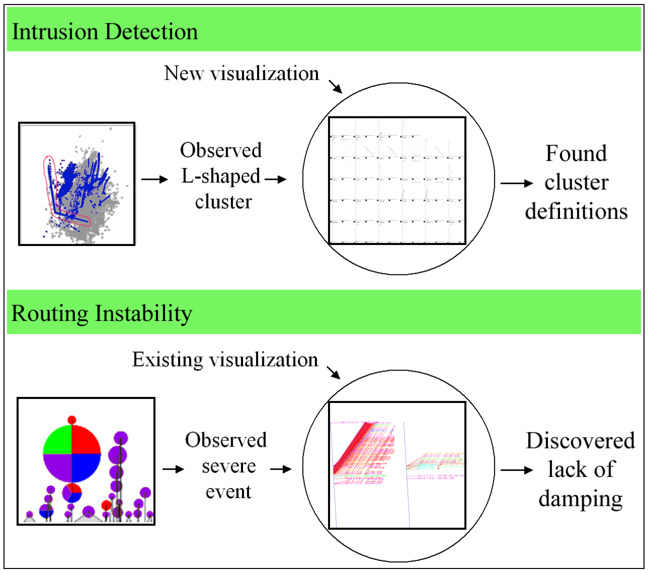

Figure 14: Two examples of two iterations of the visual analytics loop. Discovery of knowledge via one visualization can require the design of new visualization (top) or the re-use of existing visualization (bottom) for more in-depth investigation.

[3] T. J. Jankun-Kelly and K.-L. Ma. MoireGraphs: Radial focus+context visualization and interaction for graphs with visual nodes. In *Proceedings of the 2003 IEEE Symposium on Information Visualization*, pages 59–66, 2003.

[4] E. Kandogan. Visualizing multi-dimensional clusters, trends, and outliers using star coordinates. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 107–116, 2001.

[5] D. A. Keim. Information visualization and visual data mining. *IEEE Transactions on Visualization and Computer Graphics*, 7(1):100–107, January–March 2002.

[6] S. T. Teoh and K.-L. Ma. PaintingClass: Interactive construction, visualization and exploration of decision trees. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 667–672, 2003.

[7] S. T. Teoh, K.-L. Ma, and S. F. Wu. Visual exploration process for the analysis of internet routing data. In *Proceedings of the IEEE Conference on Visualization 2003*, pages 523–530, 2003.

[8] S. T. Teoh, K.-L. Ma, S. F. Wu, and X. Zhao. Case study: Interactive visualization for internet security. In *Proceedings of the IEEE Visualization Conference 2002*, pages 505–508, 2002.