Crucial Nodes Centric Visual Monitoring and Analysis of Computer Networks

Hanchen Song
National University of Defense Technology, China
Email: songhanchen@gmail.com

Chris W. Muelder
University of California, Davis, USA
Email: muelder@cs.ucdavis.edu

Kwan-Liu Ma
University of California, Davis, USA
ma@cs.ucdavis.edu

***Abstract*—Monitoring of computer network events is essential in uncertain and time varying situations. Several techniques and tools have been developed to reveal useful patterns from raw network data sets. Challenges for network monitoring include processing massive data, spotting unknown patterns, and interactive analysis for deeper reasoning. Generally, computers in intranets are categorized into crucial nodes or not depending on their roles. We address the issue of network events monitoring by focusing on crucial network nodes, and we present visualization approaches for crucial nodes monitoring and analysis. Contributions of this paper include an efficient categorization and exchange mechanism for multiple streaming data, a comprehensive interactive visualization system with coordinated views, and an intuitive radial visualization which fuses firewall data and IDS data inherently for crucial node monitoring. In our study using the IEEE VAST Challenge 2011 dataset, we found two kinds of anomalies.**

I. Introduction

In network security, traffic monitoring, analysis, and anomaly detection are critically important tasks for the early prevention of and timely response to malicious activities, such as attacks, intrusions, and worms [26]. In a typical computer network infrastructure, network monitoring usually relies on network security facilities such as firewalls and intrusion detection systems (IDS). A firewall is a service or set of devices designed to block unauthorized access while permitting authorized communications. But sometimes it is possible for an attack to come into a network via a channel authorized by the firewalls. So many firewalls are instrumented with logging mechanisms that collect records of all communications for subsequent analysis. An IDS, on the other hand, is a device or software application that actively monitors networks and/or system activities to detect suspicious behaviors. While there are several kinds of IDS, the most effective ones are based on a set of predefined rules or criteria. However, these methods will fail to detect attacks that are unknown to these predefined rules, and are also prone to false positives [2], [23].

Neither automated system is perfect, and even their combination will not always be able to defend against new, unknown attacks. So security experts must monitor the network traffic through the analysis of the complex log files from the firewall and IDS. These log files are produced in real-time, and quickly accumulate to an extreme scale, which poses a great challenge for security experts to analyze [16]. Additionally, when a network is found to be under attack, a short response time is critical.

Visualization provides a much more efficient means for analyzing large log files, which makes it an effective method for network traffic monitoring and analysis. Existing approaches often focus on analyzing a single aspect of security data. Many works analyze just the traffic patterns of a single computer or at the border firewall of a network [8], [10], [17]. But real-world networks are generally quite complex, so it is important to consider interconnectivity and not just the border between the internal network and the rest of the internet.

For instance, Figure 1 shows a typical corporation network infrastructure including several network segments, firewalls, and an IDS. Such networks often consist of tens to hundreds of computers. But not all computers are equivalent; some are critical to the entire organization (e.g. the web-server), while others are not.

We have designed an interactive visualization system which utilizes both firewall logs and IDS alerts to facilitate the analytical reasoning of suspicious network activities with respect to the crucial infrastructure nodes. Our system works in real-time, by streaming data directly from the sensors to the visualization, and efficiently categorizing and fusing the information from the firewall and IDS, which allows users to visually analyze data from multiple sources simultaneously and instantly. The visual interface consists of several linked views to help users explore the data from multiple perspectives, including radial summaries of crucial nodes' statuses, a matrix-based summary of network segment's activity, a parallel coordinates view showing the activity between segments, and a timeline of statistics of firewall and IDS logs.

The contributions of the work include:

- A highly efficient data categorization and exchange mechanism for multiple log sources, which allows the visualization system to handle large scale streaming of network security data efficiently.
- A comprehensive real-time visualization system with multiple coordinated views for network security data to facilitate timely analysis and exploration of the data.
- An novel radial visualization that intuitively fuses firewall data and IDS data for the monitoring and analysis of crucial node activity.

978-0-7695-5014-5/12 $26.00 © 2012 IEEE
DOI 10.1109/CyberSecurity.2012.9

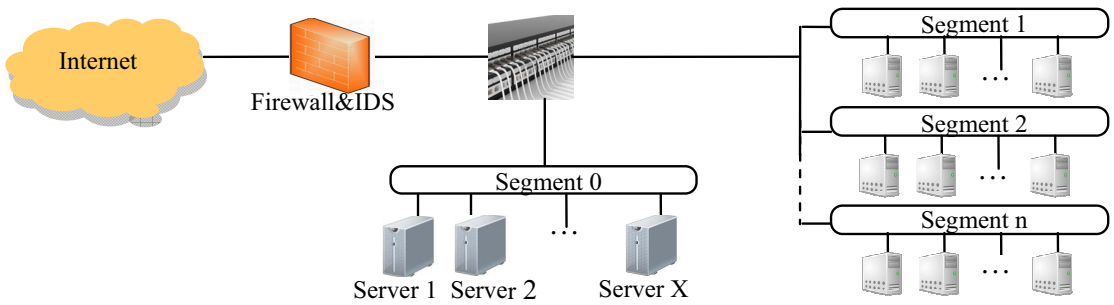


Fig. 1. A typical corporation network includes several network segments, firewall facilities and IDS. Servers play more important roles for network daily operation.

II. Related works

Network security visualization is a very diverse field, with many kinds of data sources at different levels of communication [20]. The large scale and multidimensional nature these kinds of data can conceal many anomalous events. For real-time monitoring, even processing large scale data can require extensive computation. As such, there are two important aspects of monitoring investigated in many prior works: data processing and visualization.

Data processing is a challenge because of the scale and complexity of security data sets. To accelerate the process, some solutions include hardware assistance [4] and software buffering [19]. Software buffering uses caching and buffering to improve the runtime performance as data is classified and gathered for visualization. When the visualization data structure and exchange policy are well defined, software buffering becomes more general and applicable.

Most security visualization works can be classified according to the data aspect they focus on, such as host/server monitoring, internal/external monitoring, port activity, attack patterns, and routing behaviour [20]. While most works focus on one data source, very few cover two or more of these classes[12][14]. Specifically, TVi is the only previous work to our knowledge that fuses network level data with IDS alerts [3]. But while that work addresses similar data sources, it is completely offline, and does not consider any of the concerns of real-time monitoring as in this work.

Techniques employed in security visualization are quite varied, including glyphs [7], node link diagrams[15], heat maps[1], parallel coordinate views[13], radial panels[5], histograms[21], and scatter plots[18]. Each of these techniques has advantages for particular perspectives of the security data [22]. Our approach combines several of these techniques, including a timeline histogram, a parallel coordinates, a matrix representation, and a novel radial glyph representation, where each was selected to address a particular aspect of the data.

For server monitoring applications, the most important visualization tasks include the status of the critical servers and the network connections involving them. In previous works, glyphs, scatter plots, and node link diagrams are the most employed techniques for these goals[20]. Most applications employ multiple techniques for data analysis. Some recently remarkable applications include ClockView, Spinning Cube, and TVi. ClockView is a scalable glyph representation of host activities. Multiple variables are mapped to the different sectors of a clock panel[11]. The system supports monitoring large IP spaces over time, and detailed analysis on sub networks or individual computers. Spinning Cube[27] visualizes network data in 3D space, in which a point maps a connection of features defined by the axes. Lastly, TVi combines histograms, node-link diagrams, and matrix representations for different levels of visual-based querying and reasoning of complex traffic data[3], so while it addresses similar data sources, the approach is quite different.

Eight projects were carried out in VAST challenge 2011 on the same data set [25] we used in this work. They employed various data mining and visualization tools for analysis, and many of these projects found the same intrusion and flooding events as we did. Most of them used statistical method, but some utilized visualization extensively like our system[24]. Also, unlike most of those, our system is specialized for real-time monitoring of massive data through coordinated multiple views, which provide different aspects and scalable detail of network activities.

In summary, many existing techniques have been developed for a variety of different approaches to network monitoring. With well-defined goals, these approaches can provide good results for network traffic overview, anomaly finding, or port usage monitoring. This paper focuses on a practical middle-scale intranet monitoring system, and the emphasis is on real-time analysis of crucial nodes and their external access activities.

III. Data and considerations

In this section, we introduce the kinds security data that we use, and identify the main considerations for using the data in real-time monitoring and visual-based analysis.

A. Data types

Many facilities are employed for network security. Firewalls, IDSs, network scanners, and operating system event logs

all report security events from different perspectives. Firewalls and IDSs, such as Cisco's adaptive security appliances and Snort, are the most commonly combined facilities to record and prevent network intrusions. Both firewalls and IDSs generate log records whenever network traffic violates their rule sets. These log files contain a huge amount of information which provide detailed descriptions of network behaviors over any given time period. For example, firewall log files have useful fields such as time, priority, operation, message code, protocol, source IP/port, destination IP/port, destination service, etc, and an IDS log file includes the rule violated, the priority of alert, the time at which it occurred, the source IP/port, the destination IP/port, etc. In our system, we consider both the IDS logs and the firewall logs because they complement each other: the IDS alerts may have many false positives, so they need to be verified against the network traffic, and the network traffic can reveal attacks that were missed by the IDS's rule set.

One major issue for performing this task is to improve the efficiency of data processing. The huge scale of streaming data, particularly during times of peak usage, often causes difficulties for real-time analysis of security data. Firewall logs may includes billions of records per day. During a peak time, the rate can even reach millions of records per second. It is an overwhelmingly daunting task to parse and analyze these log files in real-time, and existing systems cannot meet the demand of interaction and monitoring.

Combining multiple types of log files also leads to other problems for real-time analysis of security data. Log files are often encoded with different structures, and in some cases can be quite noisy. For example, different log files may be coded with time ascending or descending, individual computer may be recorded with different identifiers (IPs or hostnames), or some records can be missing certain fields or have invalid values. Filtering and synchronization must be done before data transfer for visualization.

As the scale of the network grows, it is quite possible for it to become infeasible to consider the entire network in real-time, as there might not be sufficient bandwidth to transfer the relevant data to the analyst, and there might be too much information to directly visualize without further computation.

B. Considerations

Computer importance varies depending on the role it plays in a network. Servers providing data, mail, web or DNS services play important roles: an intranet operation heavily depends on these servers. These servers process requests from the Internet and from other computers in the intranet. If one server refuses to work, the whole intranet may stop working properly. Servers are also the most valued target for network intrusion or attack, so it is crucial to monitor these nodes carefully, more so than the rest of the intranet.

To design an interactive visualization system which can support crucial nodes monitoring and analysis, we have the following considerations and questions to answer:**C.1**: How to process the huge scale security data for real-time visualization? **C.2**: How to visually fuse data from multiple source logs for crucial nodes monitoring in overview? **C.3**: How to narrate the events between crucial node(s) and other computers? **C.4**: How would an analyst practically use this system to monitor a network to find malicious anomalies?

IV. Methodology

This section introduces our overall visual approach, including a system overview, data categorization and exchange, and major visualization techniques.

A. Overview

The major steps and components of the crucial node centric network data visualization are shown in Figure 2. The system includes three components: data parsing, data operations, and network visualization. The data parser loads data from firewall and IDS log files, then the data operations component takes multiple sources of data and processes them by synchronizing, filtering, and categorizing. The results are then buffered for real-time data updating and historical access. Finally, data visualization and interactive analysis are provided by network visualization component . In our current implementation, the three components work as two threads: a parsing and operation thread works as a background data provider, and a network visualization thread works as a user interface.

The parsing and operation thread continuously parses data and updates the buffer pool. It includes several kinds of log file parsers which read batch data with the same time stamp. The data is synchronized by time stamp, then validated and categorized into several types of collections for visualization. Finally, the original data and collections are pushed onto the buffer pool.

The network visualization thread is designed to provide scalable detail and multiple perspectives. Because network monitoring is a time-varying task, the visualization must be easy to understand to ensure rapid perception. In particular, when crucial nodes are focused for monitoring, perspectives should cover all information needed for analysis. In this system, the network visualization component includes a crucial node view, a network segment view, and an inter-segment view. They present the data with two types of perspectives: direct security information of crucial nodes, and summaries of events between crucial nodes and other segments. These perspectives provide insight to the data from a high level to detailed level, and all the views are coordinated.

The system is designed both for real-time monitoring and forensic analysis. In real-time monitoring, the network monitors and data parsers continually drive the visualization's animation. In forensic analysis, the network visualization component drives the system by issuing data requests to the parsers on demand.

During the monitoring process with our system, the three coordinated views are animated as time passes by. Users can switch among them at will. Generally, the user will start with the crucial node view as it provides high level awareness of the current situation about all crucial nodes, and the user can

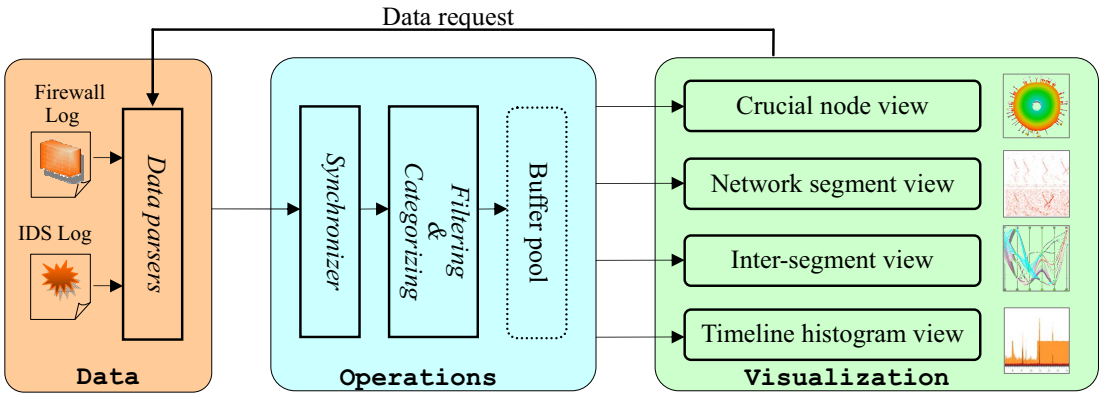


Fig. 2. The major steps and components of the crucial node centric network visual monitoring. Data is synchronized, filtered and categorized before visualization, and different views provide multiple perspectives of the security data. Two threads are employed to ensure real-time updating of massive data input.

optionally focused in on a particular system to show in more detail. Once suspicious events occur, such as sufficient IDS alerts or overwhelming traffic, the user can freeze time and switch to other views for further analysis(C.4).

B. Data categorization and exchange

As addressed in Section III, The massive data from firewall and IDS logs provides detailed information for monitoring, but it is hard for the user to deduce what parts of this data is actually relevant. One of the challenges of using this data is to extract meaningful high-level data in a timely manner. For our crucial nodes centric monitoring, we considered methods for data parsing, synchronizing, filtering, real time data exchange, and categorization for multiple perspectives.

Data parsing and synchronization correlates entries from many log files, orders them chronologically, and associates relevant records. This is implemented by having the firewall parser and IDS parser simply parse the records of the log file and forward them to a synchronizer, which takes the multiple sources of data and merges them according to their time stamps(Figure 2 "*synchronizer*"). Next, we perform data filtering to remove invalid records. Finally, we categorize and agglomerate data according to the desired visual representations, which is a high, level abstract, accumulation process. In this manner, firewall data and IDS data are grouped, combined, and fused.

In our initial experiments, we found that the data processing often causes situations when data rates can be over 10MB per update, and we found that just transferring the data cost much more time than visualizing it without applying optimization. To improve system efficiency, we implemented a shared buffer pool between the data parsing and the network visualization component in order to cache data and reduce communication costs. A FIFO data exchange policy defines how the data transfers between the data parsing and network visualization component. The shared buffer pool acts as a sliding window, with new data entries are incrementally appended to the queue head, while the oldest ones are popped from the queue tail to release memory. The queue has an alterable length which enables the user to explore the data in a period appropriate to their system limitations (C.1).

When new data is appended to the data buffers, the visualizations update accordingly. When visualizing a single point of time, the data can be transferred from the data buffer quickly. However, when visualizing data for a range of time, data exchange from the buffer pool to the visualization component may still be time consuming. We solve this by adding an additional layer of caching in the visualization components. As an update occurs, newly appended data points are accumulated, and the oldest data points are removed. These local visualizing data buffers significantly accelerate real-time animation and the shared buffer pool helps for forensic analysis. By allocating dual threads to these buffering pools, the rendering rate improved significatively: Specifically the FPS improved about 5 to 10 times.

C. Timeline histogram

A timeline histogram provides an statistical view of the overall network activity by displaying a stacked plot of the level of system events, IDS logs and firewall logs.These three types of events are stacked from bottom to top in different colors. As the data collection system produces more data points, this histogram grows, providing an overview of events across the elapsed time. In the case of forensic analysis, the entire timeline can be pre-calculated.

The interface supports standard user-interactions, such as zooming, panning, and time-range selection. By default, the system will automatically select the most recent time-step for real-time monitoring, but the timeline provides a means for the user to traverse back and analyze recent history, for instance to see what activity led up to a recent intrusion.

D. Crucial nodes view

We designed the crucial nodes view with real-time monitoring in mind. That is, the aim of this view is for a user

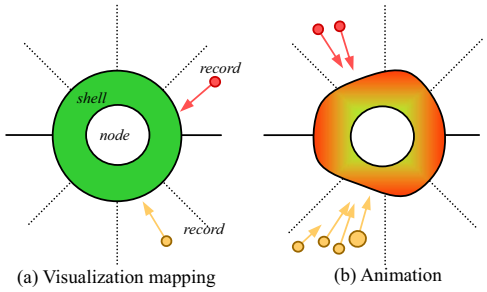


Fig. 3. The crucial nodes view. A crucial node is visualized as a cell with a shell. (a) Incoming firewall log records are distributed around the shell as attackers depending on their different categories, internet attackers are rendered with red color. (b) The shell shrinks on firewall events quantity, and changes color for green to yellow and red on IDS rules violation. Cycle size by the attacker's rail enlarges if the attacker violates IDS rule at the same time, attacker moves closer to the centre while it's more active.

to be able to read the plot and comprehend the status of the network very quickly. The crucial nodes are represented as radial glyphs which provide a combined overview of both the firewall and IDS data. This radial representation is divided into wedges, one wedge for each of a number of selectable categories, which has been shown to be flexible to the number of categories [6].

Figure 3 illustrates the crucial node visualization approach, the radial representation consists of a central circle (Figure 3(a) "*node*"), a surrounding circular area (Figure 3(a) "*shell*") and arrows around the outside. Information is encoded for visualization in four major ways(C.2).

(1) Firewall categories to wedge's angle. From the center of the central circle as the origin of polar coordinate system, firewall log records (Figure 3(a) "*record*") are distributed outside the wedge area around the *shell* as arrows; every wedge's polar angle is determined by its category. The categories can be defined by network operations, protocols, ports, network segments, or a composition of these. By default, we use network operation, protocol, and network segment.

(2) Event quantity of unique IP to arrow size and position. Within each category, firewall events are accumulated according to unique source IP addresses. The distance to the center illustrates the quantity of events for a unique source IP. And the size of the circle by the arrow tail indicates the number of IDS alerts associated with this source IP address.

(3) Event quantity of category to wedge shrinking degree . The shell animates based according to a "*stress-response*" approach, where each wedge of the *shell* shrinks smoothly according to the quantity of firewall events in the corresponding category; the depth of the shrinking is positively correlated with the quantity of firewall events. Figure 3(b) gives an illustration of the visual result.

(4) IDS type and quantity to shell color. IDS log records corresponding with the current node are filtered and categorized into several types according to severity. The quantity of these types are mapped into the percentage of color region (green, yellow, orange and red) in the *shell*. The color varies from green to red depending on the level of danger caused. E.g. if there are a large number of very severe events, the red regions will grow so the shell turn to red.

The system displays many crucial nodes simultaneously in a small-multiples matrix. But the user can expand a single node, 2×2 nodes, or 3×3 nodes to bring them to focus. The remaining nodes are displayed as thumbnails around the sides of the focused node(s). Thumbnail can be brought into the focus area by dragging it to the desired location to swap with a focused node.

The crucial nodes visualization provides a high level view of firewall and IDS events. It is the main interface for crucial node monitoring, and is the primary view from which to begin analysis. Different levels of details are provided by additional histograms when needed.

E. Network segment view

The network segment view depicts a summary of the time varying firewall events between the selected crucial node and other computers. We do this using a matrix representation where the y axis denotes the non-crucial computers' IP addresses and the x axis denotes time. Each point in the matrix changes opacity with respect to the number of firewall records involving the selected crucial node, from fully transparent when there is no traffic to opaque for the maximum observed traffic levels.

The matrix is divided in half, with the top half corresponding to traffic from the network to the crucial nodes, and the bottom showing the traffic from the crucial nodes to the rest of the network. This allows for users to compare in/out flows at any time (C.3).

F. Inter-segment view

A parallel coordinates provides a direct view of the network flows. The axes generally include at a minimum the non-crucial node IP addresses (source addresses) and crucial node IP addresses (destination addresses), and can also include proxies, source and destination ports, requested services, etc... (C.3). Records are shown as spline curves to reduce overlap and improve readability, as in the work of Graham et al [9]. We also color the spline curves according to a user-selected index axis (Figure 4(b) "*idx*") to improve resolvability. The user can also modify the sequence of axes by dragging axes to swap them.

All events from crucial nodes are shown as dot curves in individual color (Figure 4(b) "*reply*"). Wavy green regions on coordinates (Figure 4(b) "*stat*") indicate amount of records intersecting at the same position.

V. Case Study

We conducted a study on the IEEE VAST challenge 2011[25] data set which collected network events from a fictional organization known as "All Freight Corporation". Three network segments are defined in the data set: a servers segment, an office segment, and the rest of the Internet. The server segment includes a web server, a mail server, a file

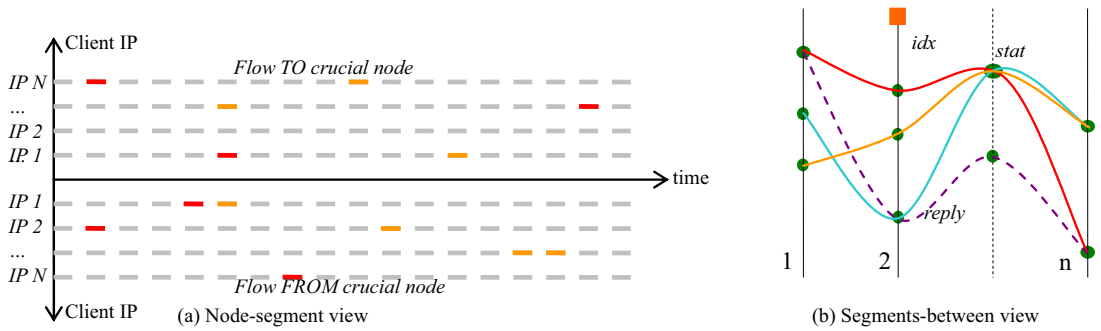


Fig. 4. Network segment and inter-segment views. (a) Network segment view gives an timeline overview of firewall events corresponding with the focused crucial node; cyclic events over outside addresses segment are distinctive in this view. the y axis denotes the non-crucial computers' IP addresses. Quantity of events is mapped into color saturation of pixels. (b) An parallel coordinates display all firewall and IDS events between different segments. Spline curve, color index, and interchangeable axes are employed to improve readability.

server, an HR database server, and two data servers. The office segment includes 256 IP addresses; these computers are workstations for the staff's daily work. The data set includes firewall logs and IDS logs from April 13 to April 15. The firewall log contains about 13 million records, with a peak rate of about 200 thousand records per minute. The IDS log is much smaller, consisting of about 30 thousands records.

In the case study, we select to analyze several columns from the firewall log for analysis: protocol, operation, source and destination IP address, destination port, and requested services. The IDS log alerts include bare byte unicode encoding, TCP window scale option, fragmentation overlap, TCP port scan, TCP port sweep and TCP/IP flooding. We use the selected firewall fields and IDS record types throughout the visualization, for arrow positioning, sizing and color mapping, etc... as described in section IV.

We used our tool to detect two anomalous patterns. The first is an IP address segment scan followed by flooding events that occur on April 13th and 14th. The other was an automated regular accessing event.

A. *Intrusion and flooding on web server*

Although there are randomly scattered IDS records which violate port-scan rules all the time, our system reveals some concentrated scan activities from two IP address segments, followed by a TCP/IP flooding intrusion on web server. While analyzing this intrusion event, a series of network behaviors appear.

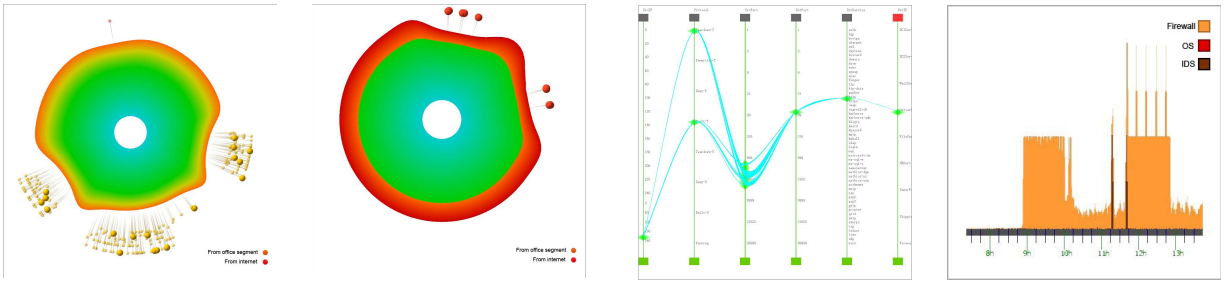
Figure 5 shows some snapshots of the intrusion process in chronological order. ***Stage 1***: TCP window scale option alarms are reported by the IDS. Figure 5(a) shows the status of data server1 at that moment (data server2 is similar). ***Stage 2***: Large scale TCP/IP flooding happens on the web server 20 minutes later, as shown in Figure 5(b) and (c). This lasts for 4 minutes. ***Stage 3***: Subsequently, steady and vast traffic comes from internet, which last for about an hour, as shown in Figure 5(d).

After noticing these anomalies, we analyze them further with our tool. In ***Stage 1***, the alarms originate from Snort decoder rule violations from the office segment to the crucial servers. Further analysis shows that these are due to a network scan originating from office segment computers. The inter-segment view reveals that these scans flow from computers 171, 172, and 173 to the entire office segment and server segment, and that the scans connect to IP addresses sequentially. Interestingly, while the firewall log flow from computer 171, 172, and 173 is stable, the suspicious intrusion records come from internet nodes. In ***Stage 2***, TCP/IP flooding originates from a short continuous internet IP address segment (9 IP addresses) and target the web server. In ***Stage 3***, the steady and vast traffic has the same sources and destination as stage 2. All records comes from 9 continuous source IP addresses and target the web server. They request *http services* from port 80 to multiple continuous ports.

From this we can deduce that the overall intrusion likely originated on the Internet, used trusted office nodes to scan and/or infiltrate the web server, then directly targeted the web server to transfer data between the web server and the attacker's network. While there is not a direct indication in the data, the tool provides sufficient reason to suspect that computers 171-173 are compromised because these computers perform a continuous scan before accurate intrusion from internet.

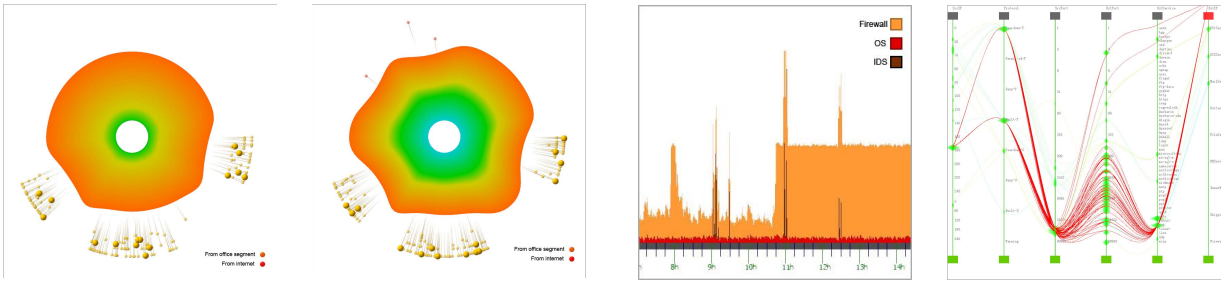
A similar event appears the next day as shown in Figure 6. Computers 174 and 175 start scanning on both the office segment and the server segment which triggers IDS alarms (Figure 6(a) and (b)). The timeline shows that this happens four times: at 09:01-09:05AM, 09:27AM, 10:58-11:02AM, and 12:27-12:30PM (Figure 6(c) dark brown). This is followed accompanied by large amount of traffic in the firewall log for 2 hours, with most of the traffic flowing from computer 174 and 175 to the server segment, as shown in Figure 6(d).

Based on these observation and analysis, computers 174 and 175 are most likely compromised as well and need inspection. These computers may have a security hole or be infected by viruses that allow them to be controlled remotely.



(a) 11:17 data server1 (b) 11:43 web server (c) 11:43 firewall (d) Timeline statistics

Fig. 5. Intrusion and flooding events on web server at April 13. (a) Start: intrusion attempt from the office segment. Data server1 as example. (b) Important event: intrusion from internet happens on web server. (c) Important event: Concentration of firewall flows accompany with the intrusion. (d)Persistence: explosive access after the important event. orange: quantity of firewall records , dark brown: quantity of IDS records.



(a) 9:05 data server1 (b) 9:05 data server2 (c) Timeline statistics (d) 12:10 firewall

Fig. 6. Intrusion and flooding events at April 14. (a) and (b) Intrusion attempt1: intrusion from computer 174 and 175 to server segment, data server1 and server2 as examples. (c) There are 4 attempts, followed by a large amount of network traffic (orange: quantity of firewall records , dark brown: quantity of IDS records). (d) the traffic goes from 174 and 175 to the web server, and targets many ports.

B. Periodic access patterns

Many network patterns exhibit periodic patterns. Figure 7(a) shows some such patterns, marked by arrows. The arrows with circles occur daily, and Figure 7(b) shows that this correspond to traffic between the mail server and nearly all the computers in the office segment between 8:00AM to 9:00AM. This is likely normal traffic, as it corresponds with the beginning of the work day.

When investigating the other periodic pattern, we find potentially interesting flows between servers and a range of about 50 continuous IP addresses in the office segment, as shown in Figure 7(c). The noteworthy characteristics are (1) traffic flows from one side to another in ascending or descending scan order. (2) This pattern is not completely constant, as it occurs about every 2 hours, but only from 14:00PM April 13 to 6:00AM April 14, and from 10:00AM to 23:00PM April 15.

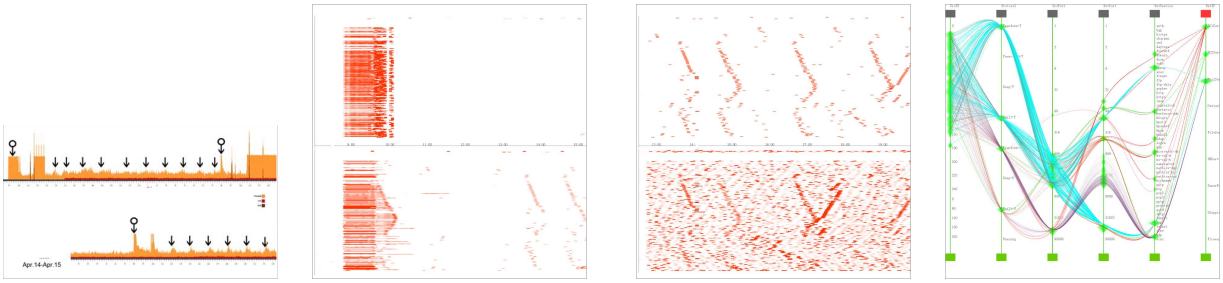
Further analysis was performed with the parallel coordinates shown in Figure 7(d). Here we see that computers in the office segment request TCP service through port 43032 and epmap service through port 135, as seen by the cyan curves. The mail server replies with UDP services through multiple ports, shown as purple dotted curves.

There were no corresponding IDS records relevant to this firewall log pattern. Other servers do exhibit similar patterns though. One explanation could be that the corporation network has an update policy which keeps communicating between servers and office segment according to time dependant variables. But knowing for sure would require further domain knowledge of the network configuration.

VI. Conclusion and Future Work

Network monitoring is an uncertain and ever changing problem. This paper has presented an approach with emphasis on the monitoring of crucial nodes, designed to reduce the cognitive load of analysts allowing for more rapid action intrusion and detection even in real-time situations, yet which also allows for in-depth analysis even in forensic analysis. Implementing such a system with these goals in mind led to the development of an efficient data categorization and exchange mechanism, a comprehensive realtime visualization system with multiple coordinated views, and an intuitive radial visualization that fuses firewall data and IDS data. Finally, the case studies demonstrated how our system is capable of detecting several kinds of anomalous activity, including attack sequences that would go unnoticed by either firewall or IDS monitoring individually.

While we found our system to be effective, there are several ways it could be improved in the future. In a system where the administrators have full control of the network, it should be possible to save full packet traces for certain durations, and extending the system to dig all the way down to the raw data on demand could be vital to intrusion response. The visualizations



(a) Timeline statistics (b) Mail server (c) Data server1 (d) 9:00 firewall

Fig. 7. Periodic patterns in Apr13. (a) The timeline reveals several periodic patterns, some are daily (indicated with arrows with circular tails) and some are more frequently (arrows without circular tails). (b) The daily spikes correspond with very regular peak on mail server usage at particular time (8:00-9:00) every day: each red line represents continuous access from one IP address in office segment. (c) The more frequent patterns occur when data servers are connected to periodically by continuous IP addresses from the office segment, causing the repetitive "Z" patterns. (d) The access patterns is potentially suspicious as it uses uncommon ports, but could be an organization-specific access pattern.

themselves could be improved with UI improvements such as animated transitions and more in-depth details on demand (via mouse-overs or labels). Finally, we would like to deploy it on a large network and test both the scalability and usability by performing a user study with real analysts.

ACKNOWLEDGMENT

This paper was accomplished with help of VIDi research group http://vidi.cs.ucdavis.edu/ at UC Davis. The authors wish to thank all the VIDians for their valuable comments and suggestions on the system and paper.

This paper was supported by the China National Science Foundation through grants 61103081.

REFERENCES

[1] S. Axelsson. Visualizing intrusions: Watching the web server. In *the 19th International Information Security Conference*, pages 59–78, 2004.

[2] M. A. Aydin, A. H. Zaim, and K. Gkhan Ceylan. A hybrid intrusion detection system design for computer network security. *Computers and Electrical Engineering*, 35(1):517–526, 2009.

[3] A. Boschetti, C. Muelder, L. Salgarelli, and K.-L. Ma. TVi: A visual querying system for network monitoring and anomaly detection. In *International Symposium on Visualization for Cyber Security*, 2011.

[4] L. Braun, A. Didebulidze, N. Kammenhuber, and G. Carle. Comparing and improving current packet capturing solutions based on commodity hardware. In *Annual Conference on Internet Measurement*, 2010.

[5] R. E. K. Christensen and Sundberg. Designing visualization capablities for IDS challenges. In *International Workshop on Visualization for Cyber Security*, 2005.

[6] S. Diehl, F. Beck, and M. Burch. Uncovering strengths and weaknesses of radial visualizations-an empirical approach. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):935–942, 2010.

[7] R. F. Erbacher, Z. Teng, and S. Pandit. Multi-node monitoring and intrusion detection. In *International Conference on Visualization, Imaging and Image Processing*, pages 720–725, 2002.

[8] F. Fischer, F. Mansmann, D. A. Keim, S. Pietzko, and M. Waldvogel. Large-scale network monitoring for visual analysis of attacks. In *the 5th International Workshop on Visualization for Computer Security*, page 111, 2008.

[9] M. Graham and J. Kennedy. Using curves to enhance parallel coordinate visualisations. In *Information Visualization, 2003. IV 2003. Proceedings. Seventh International Conference on*, pages 10 – 16, july 2003.

[10] R. R. Kasemsri. A survey, taxonomy, and analysis of network security visualization. Master's thesis, Georgia State University, 2006.

[11] C. Kintzel, J. Fuchs, and F. Mansmann. Monitoring large ip spaces with clockview. In *International Symposium on Visualization for Cyber Security*, 2011.

[12] P. Z. Kolano. A scalable aural-visual environment for security event monitoring, analysis and response. In *International conference on Advances in visual computing*, pages 564–575, 2007.

[13] S. Krasser, G. Conti, J. Grizzard, J. Gribschaw, and H. Owen. Real-time and forensic network data analysis using animated and coordinated visualization. In *IEEE Workshop on Information Assurance and Security*, pages 42–49, 2005.

[14] C. Lee, J. Trost, N. Gibbs, R. Beyah, and J. Copeland. Visual firewall: Real-time network security monitor. In *IEEE Workshop visualizaiton for computer security*, 2005.

[15] F. Mansmann, L. Meier, and D. A. Keim. Visualization of host behavior for network security. In *International Workshop on Visualization for Cyber Security*, 2008.

[16] R. Marty. *Applied Security Visualization*, chapter 5, pages 161–237. Addison-Wesley Professional, 2008.

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

[18] C. Muelder, K.-L. Ma, and T. Bartoletti. A visualization methodology for characterization of network scans. In *Workshop on Visualization for Computer Security*, 2004.

[19] A. Papadogiannakis, G. Vasiliadis, D. Antoniades, M. Polychronakis, and E. P. Markatos. Improving the performance of passive network monitoring applications with memory locality enhancements. *Computer Communications*, 34:1–15, 2011.

[20] H. Shiravi, A. Shiravi, and A. A. Ghorbani. A survey of visualization system for network security. *IEEE Transactions on Visualization and Computer Graphics*, 18:1–10, 2012.

[21] T. Takada and H. Koike. MieLog: A highly interactive visual log browser using information visualization and statistical analysis. In *Systems Administration Conference*, pages 133–144, 2002.

[22] R. Tamassia, B. Palazzi, and C. Papamanthou. Graph drawing for security visualization. In *Symposium on Graph Drawing*, pages 2–13, 2009.

[23] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin. Intrusion detection by machine learning: A review. *Expert Systems with Applications*, 36:11995–12000, 2009.

[24] VisWeek. Ieee vast challenge 2011 submission. http://hcil.cs.umd.edu/localphp/hcil/ vast11/ index.php/ submissions/index.

[25] VisWeek. Ieee vast challenge 2011data set. http://hcil.cs.umd.edu/localphp/hcil/vast11/index.php.

[26] W. Wang, X. Zhang, W. shi, S. Lian, and D. Feng. Network traffic monitoring, analysis and anomaly detection. *IEEE Network*, 25(3):6–7, 2011.

[27] A. Yelizarov and D. Gamayunov. Visualization of complex attacks and state of attacked network. In *International Workshop on Visualization for Cyber Security*, 2009.