Framework for Visualizing Browsing Patterns Captured in Computer Logs Using Data Mining Techniques
Rachael Fernandez and Noora Fetais
Pages 83 – 87
DOI: http://dx.doi.org/10.21700/ijcis.2016.111

Framework for Visualizing Browsing Patterns Captured in Computer Logs Using Data Mining Techniques

Rachael Fernandez and Noora Fetais
Department of Computer Science, Qatar University
Doha 2713, QATAR
[rf1405233, n.almarri]@ qu.edu.qa

Abstract: *An Intrusion Detection System (IDS) is used for monitoring computer security breaches by monitoring and analyzing the data recorded in log files. However, it is difficult to manually investigate the vast amounts of textual information captured in these logs. In this paper, we propose a framework for an IDS using an Information Visualization (IV) approach, which will aid the IDS administrator in effective and efficient decision-making. The proposed framework works by recording events in different logs and uses a log summarizing mechanism to limit the size of the logs. Each record or event in the log is visualized as a pixel on the screen, where each pixel can be selected to retrieve more information. A prototype of the IDS App for a simple file portal system has been developed to demonstrate the functional capabilities of the proposed framework.*

Keywords: *Information Visualization; Intrusion Detection System; File portal system; Log Summarization*

Received: July 30, 2016 | ***Revised***: August 10, 2016 | ***Accepted***: August 25, 2016

1. Introduction

An Intrusion Detection System is considered as a burglar alarm for a computer system that alerts the user in the event of an intrusion. Consider a file portal system for which the IDS analyst would like to monitor the users who access the system. Users who try to access this portal can be broadly categorized into 3 types. The first type is an authorized user who accesses the system to view the files that he/she is permitted to access. The second type is an authorized user who tries to abuse his privileges by accessing files that he is not permitted to view. The third type of users are unauthorized users who use the system with an intention to access files or harm the system. The IDS should aid the admin. in recognizing both these types of intrusions.

Traditionally, an IDS analyst is responsible for detecting intrusions in a system by performing activities that can be split into 3 phases, namely: i) Monitoring ii) Analysis and iii) Response [1]. The IDS analyst starts by monitoring the system, application and network logs to find attacks against the system. If an abnormality is observed, the analyst moves to the analysis phase in which he tries to diagnose the attacks by analyzing the users' activity pattern. After the reason has been diagnosed, appropriate steps are taken to resolve the attacks in the response phase. The analyst's job is time-consuming and inevitably prone to errors due to the large amount of textual information that has to be analyzed [2].

A survey of IDS analysts revealed a popular demand to use visualization techniques to represent the information, as opposed to using textual data that require more time for analysis [1]. For example, the severity of an attack that is represented by a red colored pixel is easier to spot and comprehend when compared to the long list of records in a log that are tagged as 'severe'. In addition, visualization approaches have the power to reveal patterns and outliers much more effectively when compared to textual data.

In this paper, we propose a framework for developing an IDS for a simple file portal system which works by monitoring the log files. The rest of the paper is organized as follows; in the next section we will present the related research, followed by the proposed framework. The section after that provides a brief description of the IDS App. Finally, in the last

section, we will present the conclusion of the paper and the future work for the research.

2. Related Research

Though there have been various frameworks for visualizing information, there hasn't been much research aimed at visualizing the events that are captured in logs. Though each line in a log can be considered as a line of information, it is important to make use of all the information recorded in the logs as they are a brief description of the events that have occurred. Komlodi et al. (2004) proposed a popular framework for IDS which are enriched with a good set of requirements for visualizing the intrusions. However, they do not provide any details for handling the data in the logs which is essentially the source of data for an IDS [1].

We follow an approach that is similar to the methodology followed by Kato et al. (2001); the authors proposed a methodology of log summarization in which interesting browsing patterns are extracted from logs which are already classified by the website domain which the user had accessed. Also, the log summarization mechanism is merely a way to consolidate all the events from different logs into a single log file which leads to a proliferation of the summarized log [3].

However, in this paper, we extract the patterns from a summarized log to classify the browsing pattern as attack, not attack or suspicious. All of the events are recorded in different logs and finally a central log is created which is a summary of all the activities of a client IP that accessed the file portal. This helps to avoid repetition of information in the central log thereby curbing its size. A detailed methodology of the log summarization is given in *Section 3.3.1*.

3. Proposed Framework

In this section, we will look at the proposed framework that will help us to design an IDS that is enriched with functions and is easy to use at the same time.

3.1 General Details

Before, we begin to design an IDS, the following design parameters should be ascertained.

1) *Visualization Technique:* One of the popular approaches is to visualize each event as a pixel in an IP matrix made up of two 2D-Matrices, which leads to a scatterplot of pixels [4]. Another approach that is used to monitor user's activities is to use a pie chart variant which is composed of concentric circles instead of slices, where the events in the outermost circle indicate the file accesses from the same network domain, whereas the events in the innermost circle represent accesses from a foreign domain which could potentially be an attack [5].
2) *Graphical Representation:* The design of the graphical interface has to be defined, which includes decisions about the ancillary screens. These secondary screens should display more information about the selected event in the primary screen, without providing an overload of information at the same time. The layout of these screens have to be decided as well.
3) *Periodicity of Refresh:* If the periodicity of refresh is short, many events that occurred recently might be lost due to the constant refreshing of the screen. On the other hand, a longer periodicity could lead to a frame that does not have much connection to the frame displayed before the last refresh. The periodicity should be defined such that there is a balance the two issues, so that the frame of data that appears after the last refresh is not vastly different from the frame just before the refresh [6].
4) *Types of Events:* The events that the system is expected to capture are defined next. For example, in our IDS App for the file portal system, the IDS App captures the user's sign-in/sign-out, file accesses, file addition and file deletion activities in the portal.

3.2 Activities corresponding to the three phases of Intrusion Detection

In this section, we will discuss about the activities pertaining to each phase. These tasks are in accordance with Shneiderman's visualization task list [7]. At the end of each sub-section, the log agent tasks for that phase are presented.

3.2.1 Monitoring

In this phase, the events that are displayed as pixels on the screen are monitored for detecting any anomalies. This phase requires the following capabilities [2]:

1) *Zoom:* The user should be able to zoom into a pixel or a set of pixels for more clarity. The zoomed view should be visualized in the ancillary display/s and provide extra information regarding the selected pixel/s.
2) *ToolTip:* The system should provide some basic information about the selected pixel when the cursor hovers on a pixel.
3) *Alert:* In the event of a suspicious or potential attack, an alert should be generated to notify the IDS analyst.
4) *Log Agent Tasks:*
   i. Each event should be recorded in the corresponding log, based on a pre-defined format. For example, all sign-in and sign-out activities are recorded in detail in the user activity log.

ii. The log agent creates a central log, which contains the summary of user activities and other information like date and time for every client IP address that accessed the portal. The log agent does this by summarizing the events from all the log files that correspond to a particular IP address.

3.2.2 Analysis

After an alert has popped-up, the IDS moves into the analysis phase in which we try to diagnose the reason for the notification. The following features have to be offered in this phase.

1) *History:* If the type of the attack is recognized, then the details about how it was resolved before, should be displayed in the ancillary screen.
2) *Filtering:* The system should allow for displaying the events based on their type and severity.
3) *Blinking:* The change in type of an event should be emphasized to the user by a blinking of the pixel to grab the attention of the user.
4) *Log Agent Tasks:* The log agent should be capable of recognizing the patterns in the log by using a Data Mining Engine (DME)**.**
5) *Pattern Similarity Score*: If the analyst has some suspicions regarding the browsing pattern of a client, then the analyst should be able to view a similarity score between the browsing pattern of a suspicious IP and the browsing patterns of other IPs that have been recognized as an attack. This score will aid the analyst in decision-making in case of a 'suspicious' event.

3.2.3 Response

In this phase, if the IDS analyst is confident that the user's browsing pattern is an "attack", then the IDS analyst should be able to report the attack along with some more details regarding the resolution of the attack.

1) *Report Information:* Once the issue has been resolved, the details regarding the resolution of the attack should be recorded. This information is saved as *History* and can be used later for calculating the Pattern Similarity Score.

3.3 Log Agent

The Log Agent consists of two components; the Log Summarizer (LS) and the Log Files Collector (LFC). The LFC collects all the log files in which each file contains details about the corresponding events. For example, the File Access Log contains the IP Address, Time and File Name of each file that was accessed in the portal. The bundle of log files is fed to the LS which creates a Central Log file. This file helps in summarizing all of the user's activities in a single file. However, the detailed information that is found in each individual log file is lost in the Central Log.

In Figure 1, we can see that the File Access Log contains the name of the file that was accessed (E.g. File A or B). However, this detail is replaced by the keyword Access in the central log where we are only interested in capturing the browsing pattern of the user without worrying about the files that were accessed.

3.3.1 Log Summarizing Mechanism

The events that occur are recorded in the corresponding logs. The LS is responsible for summarizing the events in the logs and writing the summarized events to the Central log from which the events are visualized. Consider the following example, where a User Signs-in and accesses some files multiple times within a certain period. The summarized log contains only one line of information associated to a session of a specific IP address. The time-stamp corresponds to the first activity of the user within a specific time period followed by the activity that the user accomplished in that session without any additional details regarding the files that were accessed. Just by reading one line of information from the central log, the system can identify the user's browsing patterns.

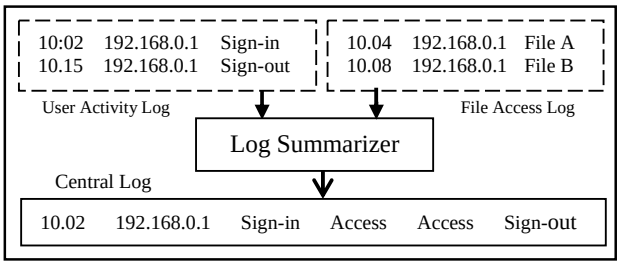

Figure 1. Working of the Log Summarizer

Consider the case, where the central log is set to be backed up at 10:00 (scenario depicted in Figure 2). We assume that the system automatically signs-out users who are inactive for more than 5 minutes. If a user signs-in at 9:58 and then (central log will contain the information for the "Sign" activity) remains inactive for 3 minutes, the time will now be 10:01 and the events before the last backup at 10:00 are archived but lost in the new central log. If the user now accesses File A, it would lead to a new line in the central log corresponding to the "File" activity as the system considers it a new session. This could be misinterpreted as an "Attack" as there is no corresponding "Sign-in" activity, which could signify a hack. In order to avoid this, the last 5 minutes of activity which is the window corresponding to the time that a user will be logged-out being inactive, should be copied to the new log before backup.

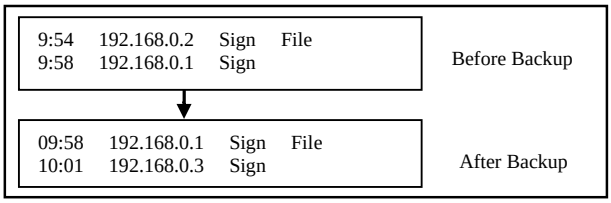

Figure 2. Contents of the Central Log before/after backup

4. Prototype of the IDS App

In this section, we will present the prototype of the IDS App for a file portal system that was built using the proposed framework.

All users of the file portal system can Sign-in/Sign-out and access the files in the file portal. However, only the higher management users have the provision to add and delete the files. A small group of 5 users were asked to simulate the behavior of working with the files in the portal and these activities were captured for the purpose of this research. Some of these users were given permission only to access the files, whereas the other users were given the special permission to carry out addition and deletion of files.

The IDS application's graphical interface consists of two screens. The primary screen displays the events, which are visualized as pixels of different colors based on the severity: Attack (Red), Suspicious (Yellow) and Not Attack (Green). Five events were captured, namely: Sign-in/Sign-out, File Access, File Addition and Deletion. A consolidated view of the primary and secondary screens in the IDS App is shown in Figure 3.

The IDS App visualizes each event in the log as a pixel on the primary screen. The screen on the top left corner in Figure 3 displays the pixels for the events captured in all the log files. This screen gives the users the opportunity to visualize events from a single log file or from all the log files. When the pixel on this screen is selected, details about this pixel is displayed as a tooltip. It also displays some additional information like the browsing history as a graph for easy viewing (small graph shown in Figure 3) and a map displaying the origin of the client IP request.

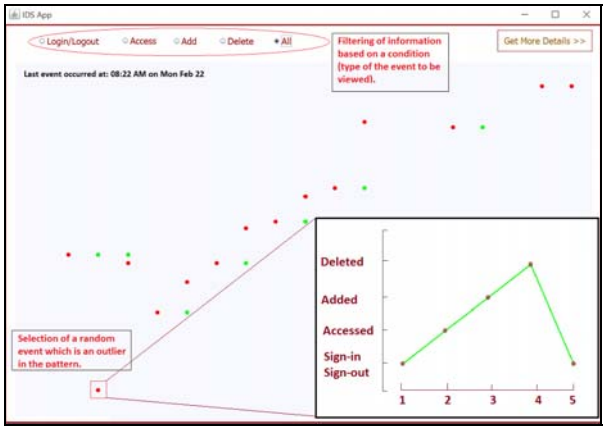

Figure 3: Consolidated view of the screens in the prototype

If an event has been deemed suspicious by the IDS, the Pattern Similarity Score can be calculated between the suspicious IP and the other IPs that were classified as an attack previously, by comparing their browsing patterns in order to help the analyst with the decision of classifying it as an attack/not attack. If the analyst decides that the event should be classified as an attack, then the pertaining IP address can be reported along with the steps taken to resolve the attack. This attack is then added to the list of previously recognized attacks.

A DME is used for analyzing the user's browsing patterns, which are collected from the central log. It has access to a list of patterns that have been classified as an "attack" by the system or by an IDS analyst **Error! Reference source not found.**. The DME compares the browsing pattern of a user to these 'recognized' patterns to detect intrusions into the system.

5. Conclusion and Future Work

The proposed framework endows the prototype of the IDS App with rich functionalities which helps the IDS administrator in recognizing potential attacks by comparing the browsing patterns with the patterns of previously classified attacks and by assisting them in each phase of the intrusion detection process. Future work for this research includes implementing the DME and finding the hopping pattern of the client IP requests, so that we can identify routers in the network which are prone to attacks.

References

[1] Komlodi, Anita, John R. Goodall, and Wayne G. Lutters. "An information visualization framework for intrusion detection." *CHI'04 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2004.

[2] Takada, Tetsuji, and Hideki Koike. "Tudumi: Information visualization system for monitoring and auditing computer logs." *Information Visualisation, 2002. Proceedings. Sixth International Conference on*. IEEE, 2002.

[3] Kato, Hisayoshi, Hironori Hiraishi, and Fumio Mizoguchi. "Log summarizing agent for web access data using data mining techniques." *IFSA World Congress and 20th NAFIPS International Conference, 2001. Joint 9th*. IEEE, 2001.

[4] Koike, Hideki, et al. "Visualizing cyber-attacks using IP matrix." *IEEE Workshop on Visualization for Computer Security, 2005.(VizSEC 05)*.. IEEE, 2005.

[5] De Paula, Rogerio, et al. "In the eye of the beholder: a visualization-based approach to information system security." *International*

Journal of Human-Computer Studies 63.1 (2005): 5-24.

[6] Livnat, Yarden, et al. "A visualization paradigm for network intrusion detection." *Proceedings from the Sixth Annual IEEE SMC Information Assurance Workshop*. IEEE, 2005.

[7] Shneiderman, B. (1996, September). The eyes have it: A task by data type taxonomy for information visualizations. In Visual Languages, 1996. Proceedings., IEEE Symposium on (pp. 336-343). IEEE.

Rachael Fernandez is pursuing her M.S degree at Qatar University, and is currently in the second year of the Computing program. She joined Computer Sciences Corporation (CSC) as an Associate Software Engineer for a brief stint before joining the university as a Graduate Assistant. She has just been awarded a Research Grant from the university for her research in the Information Visualization domain. Her research interests include Data Mining, Information Visualization and Information Retrieval.

Dr. Noora Fetais is the vice-chair of IEEE-Qatar Section as well as the Qatar University Faculty Senate. She is an assistant professor at the Computer Science and Engineering department at Qatar University. She got her PhD from the University of Sussex, UK. Her expertise is in visualizing and interpreting data and she leads over 6 research grants.