A Survey : Server timeline analysis for web forensics

Rohit Chandrawanshi¹, Hitesh Gupta²
Research Scholar¹, HOD², Dept. of Software System
Patel College of Science and Technology
Bhopal, India
rohit_chandrawanshi1@rediffmail.com¹, hitesh034@gmail.com²

Abstract: This paper describes an extensive study on the existing methods and techniques for the web server analysis. The web server analysis is important in forensic study as it analyses the web log files to discover user-accessing patterns of web pages. In order to effectively manage and report on a website related to any miss happening, it is necessary to get feedback about activity on the web servers. The aim of this study is to help the web designer and web administrator to improve the impressiveness of a website by determining occurred link connections on the website. Therefore, web logs files are preprocessed and then path analysis technique is used to investigate the URL information concerning access to electronic sources. The proposed methodology is applied to the web log files in the web server. The results and findings of this experimental study will be used by the forensic investigators for the investigative purpose. On the other side, the proposed timeline analysis can be used by the web designer in order to plan the upgrading and enhancement to the website.

Keywords: Activity logs, Time line analysis, web forensics, cyber security

1. INTRODUCTION

Web servers and web-based applications are popular attack targets. Web servers are usually accessible through corporate firewalls, and web-based applications are often developed without following a sound security methodology. Attacks that exploit web servers or server extensions (e.g., programs invoked through the Common Gateway Interface [1] and Active Server Pages [2]) represent a substantial portion of the total number of vulnerabilities. For example, in the period between April 2001 and March 2002, web-related vulnerabilities accounted for 23% of the total number of vulnerabilities disclosed [3]. In addition, the large installation base makes both web applications and servers a privileged target for worm programs that exploit web-related vulnerabilities to spread across networks [4].

To detect web-based attacks, intrusion detection systems (IDSs) are configured with a number of signatures that support the detection of known attacks. For example, at the time of writing, Snort 2.0 [5] devotes 868 of its 1931 signatures to detect web-related attacks. Unfortunately, it is hard to keep intrusion detection signature sets updated with respect to the large amount of vulnerabilities discovered daily. In addition, vulnerabilities may be introduced by custom web based applications developed in-house. Developing ad hoc signatures to detect attacks against these applications is a time-intensive and error-prone activity that requires substantial security expertise.

The occurrence that makes modifications in the state of a computing system is called an event e. A crime or incident is an event that violates policy or law. An event chain E = e1, ... , en is a sequence of events with a causal relationship. The latter definitions are adopted from [6] [7]. Evidence dynamics are described in [8] to be “any influence that changes, relocates, obscures, or obliterates physical evidence, regardless of intent”. A central issue in evidence dynamics is to identify the causes and effects of events. The evidence dynamics of different digital media varies. A file can be modified or deleted, and timestamps can be updated. Unallocated data on a disk can be overwritten, and volatile memory can be overwritten or moved to pagefiles. Data transmitted on a network may leave traces in log files and monitoring systems.

Logs offer an endless well of valuable information about systems, networks, and applications. Through logs, audit records, and alerts, information systems often give signs that something is broken (or “broken into”) or will be broken soon. They can also reveal larger weaknesses that might affect regulatory compliance and even corporate governance. However, more often than not, system and application logs contain raw data rather than information, and thus require extra effort to extract or distill this data into something useful, usable, and actionable [9].
2. LITERATURE SURVEY & COMPARATIVE STUDY

This section provides a detailed overview of the existing systems that help the investigator to find out the prescribed reports and evidences. These reports and evidences then use in the court against the accused of the crime.

A work has proposed neural networks for automated event reconstruction [17]. However, the approach in this paper searches for patterns of events in the low-level timeline based on predetermined rules. The approach is based on a plugin framework where each plugin is a script that detects a particular type of high-level event. Each ‘analyser’ script contains criteria that specify the low-level events that should be present if that high-level event occurred, and searches the entire timeline for low-level events that match within a specified period of time. An analyser is made up of a number of components.

Formal frameworks for the reconstruction of digital crime scenes are discussed by Stephenson [10] and Gladyshev et al. [11]. Stephenson uses a Petri Net approach to model worm attacks in order to identify the root cause of an attack. Gladyshev et al. present a state machine approach to model digital events. Their approach uses a generic event reconstruction algorithm and a formal methodology for reconstructing events in digital systems.

A significant challenge in digital forensics is to achieve automated evidence analysis and automated event reconstruction. Stallard and Levitt [12] [13] have proposed an expert system using a decision tree to search for violations of known assumptions about data relationships, and Abbott et al. [14] have proposed a framework for scenario matching in forensic investigations based on transaction logs with automated recognition of event scenarios based on a stored event database. These approaches do not suggest replaying the scenarios on a testbed, but the output of their systems could be used as a basis for realistic testing in ViSe. This would provide a far more thorough analysis and a more convincing case in court.

Elseasser and Tanner [15] have proposed an automated diagnosis system that generates possible attack sequences based on profiles of the victim host configuration and of the unauthorized access gained by the attacker. The hypothesized attack sequences are simulated on a model of the victim network, and a successful simulation indicates that the attack sequence could feasibly lead to unauthorized access. Neuhaus and Zeller [16] have recently proposed a method for automatically isolating processes that are necessary for an intrusion to occur. They propose to capture system calls on a live host and then replay these on a testbed. Their implementation, Malfor, has proved able to identify both the root cause and all intermediate steps needed to reproduce an attack.

In the approach proposed by Olsson and Boldt et. al [18] improved upon file metadata based timelines with the Cyber Forensic Time Lab (CFTL). This tool not only recovers file system times from FAT and NTFS volumes, but also extracts times from a variety of files, for example EXIF data, Link files, MBOX Archives and Windows Registry files. Interestingly, CFTL also maintains some information about the source of extracted events. It is also suggested that an extension of the work could be to automatically search for “certain predefined patterns of suspicious activity, helping the investigator to spot interesting parts of the timeline more efficiently”.

Also, log2timeline in Guðjónsson, 2010 [19], with the time-scanner enhancement can automatically and recursively examine files and directories. If an appropriate ‘input module’ is available for a file, times are extracted and added to a timeline. Reference (Guðjónsson, 2010) also hints at the possibility of grouping events that are part of the same activity when describing the potential future use of the ‘super event’ table in the SQLite output format. A more detailed review of available timeline software is available in Carbone et. al. [21] but the examples in this subsection demonstrate that there are a number of benefits to using an ‘enhanced’ timeline in addition to improving the richness of the timeline, i.e. increasing the number of events. As discussed in [18], a tool such as Time stamp could be used to clear file system times, but this would not affect timing within files. Even if not overwritten maliciously, file access times can be updated in bulk by anti-virus products [19] or the updating of them disabled by default in modern operating systems or by altering a Registry key.

There is also some work that discusses the visualisation of digital forensic timelines. For example, EnCase’s visualisation is mentioned in [20]. Buchholz and Falk [22] developed Zeitline, which is a GUI based tool that allows file system times to be imported from The Sleuth Kit and other sources (using Import Filters). This tool provides searching and filtering of events. It also introduces the concepts of atomic events and complex events, where the former are “events that
are directly imported from the system” and the latter are “comprised of atomic events or other complex events”. Zeitline [22] allows an investigator to manually combine atomic events into complex events. Aftertime (Netherlands Forensic Institute (NFI Labs), 2010) is a Java based application that not only performs enhanced timeline generation from a disk image, but also visualises the results as a histogram, with time on the x-axis against numbers of different events on the y-axis.

Lerche and Koziol give the overview of visualization of forensic data. Basic and fundamental visualization was explained in his work. How different techniques could be used in forensic process also discuss. Also they focus how visualization helps to detect anomalies and attack in network forensics [25].

Cluster based groping of similar data of different density in analysis of log files. Choose candidate outlier and compute the distance between candidate point and non candidate cluster. They found to be for then it is anomaly, this approach is presented in [24].

After studying all the major exiting techniques and tools for forensic analysis, we found that there is still an open space for the development and research on automated forensic timeline analysis tool, that can be compatible enough to handle the web log files as well firewall log files with the advanced correlation strategy.

In general the goal of proposed server timeline analysis is to find digital evidence for an investigation. It also provides capacity to store large volume of data. Multiple customized linear searching algorithms provides an ability to extract useful patterns and anomalies from data that could be presented using visualization techniques.

3. PROPOSED METHODOLOGY

In this paper, we present an automated timeline tool for analysing of the web servers for the forensic analysis. In the proposed system, we have developed tools that assist the server administrator and web administrator to improve their website by determining occurred link connections in the website. Firstly, we have obtained access log files, which are recorded in web server. The obtained log files were analysed by proposed methodology. So, raw log files were pre-processed and the path analysis technique was used to investigate the web log files of URL information concerning access to electronic sources. The proposed methodology was applied to the user access log files in the web server. The results and findings of this experimental study can be used by the web administration and web designer in order to plan the upgrading and enhancement to the website.

![Sample Log file](image)

Figure 1: Sample Log file

The above snapshot shown in figure 1 depicts the sample log file, which is maintained and managed by the web server system that stores the information of the users and web contents. It basically manages the records consist of several parameters.

Our work proposed here in this paper will focus on the analysis of events with visualizations for the web access log files events or simply web log file events. Figure 1 is an example of one entry in a log file.

The proposed technique for the web server time line analysis will perform the following actions that provides support to the investigators. First, the experts will take the log files and by using the proposed tool for log files visualization and analysis starts drawing an action plan. They may perform the following steps for investigating the crime scene.

Step1- Analyse the dates and day based on occurrence of event. (Reported by the server administrator)

Step2- Integrate the log files as per the need.

Step3- Identify the parameters for analysis, that helps to collect the evidences of malicious activity like (IP ADDRESS, DATE, MAC ADDRESS etc.)

Step4- In the integrated log files, search the activity for frequent visitors with their ip addresses, file access, byte transferred etc.

Step5- Confirm the IP address of the malicious user, by analysing the behaviour.
Step 6: Reconstruct the event timeline hypothesis by analysing the evidences and log file entries.

Step 7: Generate the reports as the evidences.

Step 8: Present all the reports to the courts, that supports the prosecution to convince the court against the accused.

Figure 2: Proposed approach for the server timeline analysis tool

In figure 2 the algorithm of the proposed tool is presented. The reconstruction time line analysis

The proposed server timeline analysis tool published in this paper supports many of the proposed solutions for automated forensic analysis, and it would be interesting to integrate some of these approaches with our work. It generates hypotheses before executing the process of reconstruction experiments and the problem of performing automated comparison of the results with the digital evidence.

4. CONCLUSION & FUTURE WORK

Digital forensics involves the application of tools and technologies to prove the truth of a past event. The discipline of digital forensics has developed methods to enhance the identification, correlation, and characterization of digital information. As with other forensic disciplines, technology is used to increase information symmetries so that recreations of past events more probably reflect the true event, and justice is served.

In court, a reconstruction will be subject to thorough questioning. It is essential to convince a court that the testing is forensically sound and that it is relevant to the original digital crime scene. Although a reconstruction can neither prove a hypothesis with absolute certainty, nor exclude the correctness of other hypotheses, a standardized environment, combined with event reconstruction and testing, can lend credibility to an investigation and be a great asset in court. Future work on understanding the effects of anti-forensic tools on a reconstruction will add value to the approach.

REFERENCES

[19]. Guðjónsson K. Mastering the super timeline with log2timeline. SANS Reading Room; 2010.