Implementation Of An Automated Server Timeline Analysis Tool For Web Forensics

Rohit Chandrawanshi[1], Hitesh Gupta[2]
Research Scholar[1], HOD[2], Dept. of Software System
Patel College of Science and Technology
Bhopal, India

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 pre-processed 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 digital crime scene is non-other than various computing, storage devices and connecting network. In fact the digital crime scene can consist of a number of computer systems, which are broadly divided into three categories: namely attack hosts, victim hosts, and third-party hosts. The third-party hosts may, for instance, include network or security services that perform logging, or other service providers such as certification authorities. All evidence is analysed on analysis hosts, which are not part of the digital crime scene.

The Digital evidence is a digital data obtained from any reliable sources and must contain appropriate information that supports or refutes a hypothesis about an incident. Digital evidence may be found on the hard drives or in the volatile memory of all the hosts involved, as well as in capturing network traffic, referred to as network dumps. A variant of the network dump is preprocessed network traffic, such as network intrusion detection system alert logs. All analysis is assumed to be performed on copies of the evidence in order to preserve the integrity of the evidence.

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.

These log files are used further to detect network and cyber anomalies. Anomaly detection is a process in which localize object that are different from other objects, and that know as anomaly. Anomalies have attribute values that deviate significantly from the expected or typical attribute values or behaviour. The goal of anomaly detections to find object that are different from object. It is important for find unusual behaviour in data. An anomaly commonly causes due to different class data, different natural variations.

2. LOGS AND ATTACKS

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].

At the very highest level, logs are a vehicle of accountability. Of course, an organization has many other mechanisms for accountability, but logs pervade all of its IT, and if its IT isn't accountable, the organization probably isn't either. Various logs are also valuable for regulatory compliance programs. Many recent laws and mandates have items related to audit logs. For example, a detailed analysis of the security requirements and specifications outlined in the Health Insurance Portability and Accountability Act (HIPAA) reveals some items relevant to auditing and logging, such as the audit, logging, and monitoring controls for systems that contain patient information. Similarly, the Payment Card Industry Data Security Standard (PCI DSS) explicitly mentions logging, log collection, and data retention and log analysis for systems involved in credit-card transactions. Clearly, the importance of logs for compliance will only grow as standards become the foundations for new regulations that are sure to emerge.

3. LITERATURE SURVEY

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.

Similarly, 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 sub-section 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.

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.

4. PROPOSED METHODOLOGY

This section will provides a detailed proposed work, which is already published [23]. The proposed research aims to determine the extent to which it is possible to automate the manual process that an investigator can undertake to combine the website analysis from an IIS server or APACHE web server. So, that one can know how to manage the website. This will also help the admin to control the web server because the proposed tool derives the indicators about when, whom, and how a web server is visited. This will also help the optimization of search engines in order to identify the context based on association with website design of an e-commerce web portal that demands high security.

To establish foundational support for the forensic soundness of remote digital evidence collection, it is useful to understand how the chain-of-custody has been practically applied for physical evidence. Chain-of-custody is a process consisting of methodical checklists and procedures during the collection, preservation and analysis of evidence for the purpose of establishing authenticity and reliability of evidence. In other words, the evidence offeror tries to prove the chain-of-custody in order to rebut or minimize the charges that evidence may be tainted or altered. Since chain-of-custody is functionally abstract, we can apply the court-proven principles, policies and procedures that control for the admission of physical evidence to digital evidence.

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.

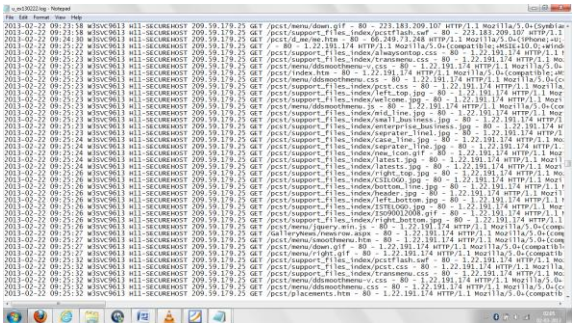


Figure 1: Sample Log file

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 interpretation of the fields shown in figure1 is:

1) 2010-11-19: Date at which the entry is recorded.

2) 19:24:24: Time at which the entry is recorded

3) W3SVC1: Name of the server

4) 192.168.1.57: IP address of the server

5) Get: Method of HTTP request

6) Pcst/login.aspx: The resource requested

7) userid=rohit&pwd=rohit123: Parameters associated with the resource requested

8) 80: Port number

9) - : This is the user name if the site requires user authentication. If not the hyphen is placed

10) 93.186.23.240: Client IP address

11) Mozilla/4.0+ (compatible;+MSIE+4.01;+Windows+NT): Browser name and version and the operating system.

12) 200: It is the status code returned to the user which in this case means that the request is successfully executed.

13) 3223: The bytes sent from the server to the client in response to the user request

These are the general fields that can be recorded in the log file; these fields can be customized (by adding or removing) depending on server administrator needs.

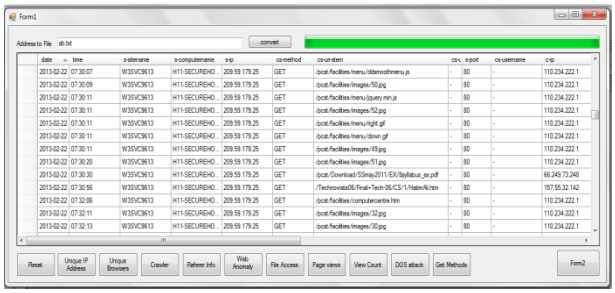


Figure 2: Importing the log files into the dataset

The above figure 2 is the snapshot of the interface of implemented software program that initially parses the event log files taken from the server into the appropriate dataset, which helps the cyber investigators to apply various methods and techniques to find out several conclusions, that helps to identify the malicious activities.

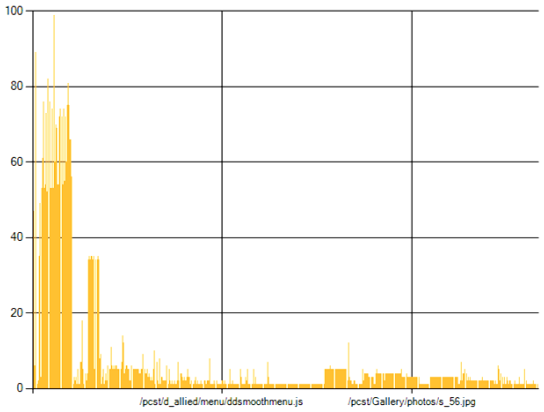


Figure 3: Graph for the page count

The figure 3 specified the visualization of page count, according to the server log file. This visualized graph shows the popularity of the page according to the page count. From the above figure it is distinctly seen that, which page gets the most visits of users.

/pcst/index.htm	99
/	89
/pcst/support_files_index/header.jpg	82
/GalleryNews/newsrow.aspx	81
/pcst/support_files_index/transmenu.css	76
/pcst/support_files_index/asct.css	76
/pcst/menu/jquery.min.js	75
/pcst/menu/smoothmenu.htm	75
/pcst/support_files_index/left_top.jpg	74
/pcst/menu/ddsmoothmenu.js	74
/pcst/support_files_index/right_bottom.jpg	74
/pcst/support_files_index/mid_line.jpg	73
/pcst/support_files_index/bottom_line.jpg	72
/pcst/support_files_index/right_top.jpg	72
/pcst/support_files_index/left_bottom.jpg	72
/pcst/menu/ddsmoothmenu.css	70
/pcst/menu/ddsmoothmenu-v.css	69
/pcst/menu/down.gif	66
/pcst/menu/right.gif	66
/pcst/support_files_index/favicon.ico	61
/pcst/support_files_index/seprater_line.jpg	60

Figure 4: Table for the different pages with their counts

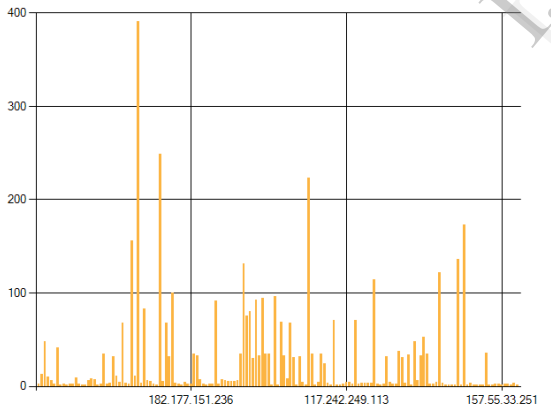
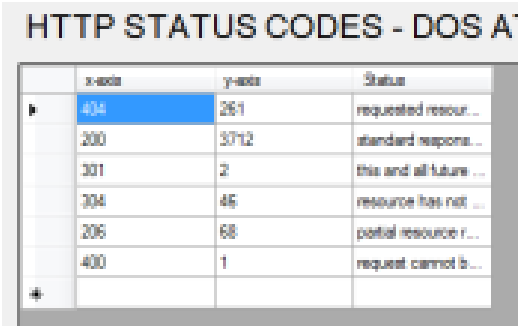
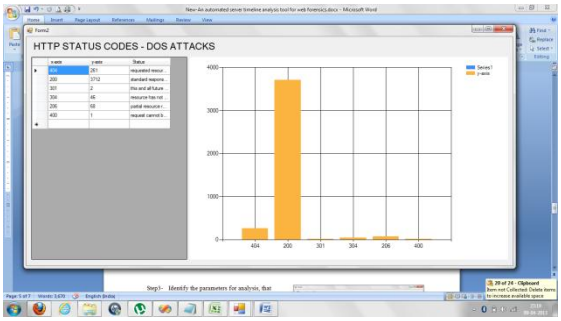


Figure 5 : Graph specifying the frequent visitors with ip-addresses

The figure 5 specified the visualization of records of different IP-addresses visits, according to the server log file. This visualized graph shows the visit count of different ip-addresses. From the above figure, it can be easily identified the frequent users and their behaviours towards the web page.





x-axis	y-axis
110.234.222.1	391
117.225.196.249	249
103.2.232.82	223
117.201.232.53	173
117.200.88.171	156
122.168.229.149	136
111.93.64.59	131
123.238.176.215	122
223.196.187.148	114
122.167.10.31	100
115.248.250.181	96
1.22.191.174	94
223.183.15.39	92
223.186.94.153	91
122.164.251.253	83
49.202.178.251	80
122.168.135.8	75
122.168.200.236	71
117.241.49.184	71
122.168.212.149	69

Figure 6: Unique IP visit count

HTTP STATUS CODES - DOS

	x-axis	y-axis	Status
▶	404	261	requested resource could not be found
	200	3712	standard response for successful HTTP requests
	301	2	this and all future requests directed to the given ...
	304	46	resource has not been modified since last reque...
	206	68	partial resource return due to request header
	400	1	request cannot be fulfilled due to bad syntax
*			

Figure 7:HTTP status codes - DOS Attacks

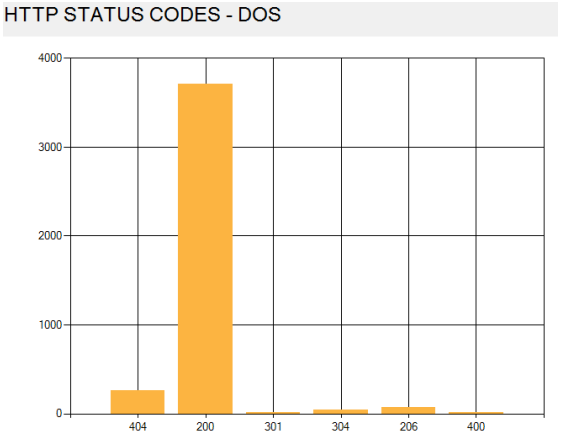


Figure 8: Visualization of DOS attacks

EXPERIMENTAL CASE – Consider the following scenario:

Condition: cyber security experts are reported by the system administrator that for any website (www.pcst.com) some user trying to access the restricted material, that is not authorized for external users. (Note: This case is similar to the problems arises by the wiki leaks, which avails the sensitive confidential document of various organisations to public.)

Prerequisite: Log records for the websites of any specified of dates.

Action taken by the experts: 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.

Step6- Reconstruct the event timeline hypothesis by analysing the evidences and log file entries.

Step7- Generate the reports as the evidences.

Step8- Present all the reports to the courts, that supports the prosecution to convince the court against the accused.

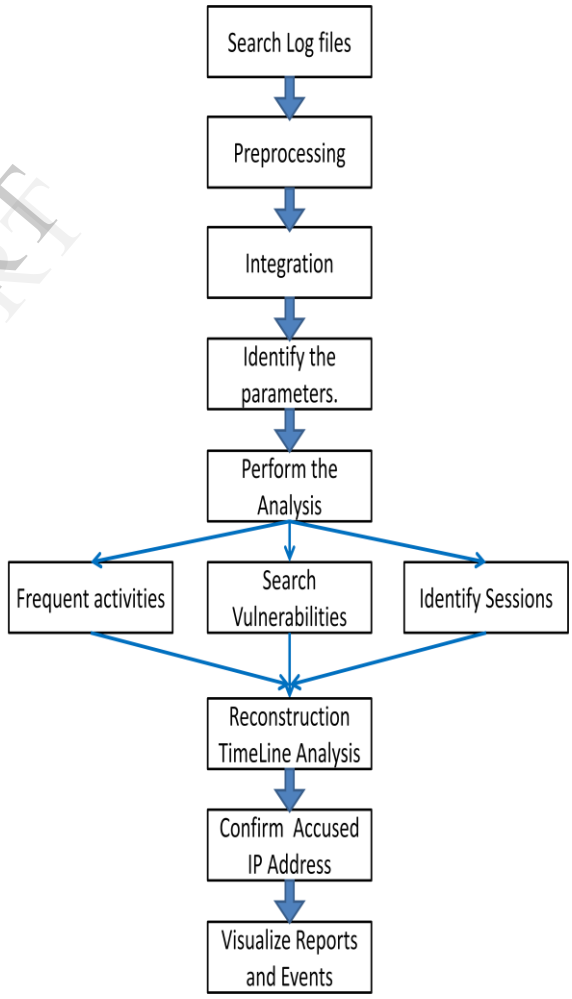


Figure 9: Proposed approach for the server timeline analysis tool

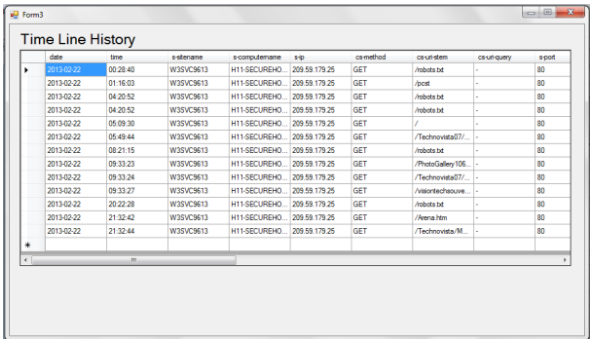


Figure 10: Time Line History for the Suspected IP address

5. 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.

The proposed research published in this paper supports many of the other existing forensic analysis, and it would be interesting to integrate some of these approaches with our work. Of particular importance are the problem of generating relevant hypotheses before performing the reconstruction experiments and the problem of performing automated comparison of the results with the digital evidence. Automating these tasks would dramatically increase the efficiency and usability of performing reconstruction experiments.

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.

6. REFERENCES

[1]. K. Coar and D. Robinson. The WWW Common Gateway Interface, Version 1.1. Internet Draft, June 1999.

[2]. J. Liberty and D. Hurwitz. Programming ASP.NET. O'REILLY, February 2002

[3]. Security Tracker. Vulnerability statistics April 2001-march 2002. http://www.securitytracker.com/learn/statistics.html, April 2002.

[4]. CERT/CC. "Code Red Worm" Exploiting Buffer Overflow In IIS Indexing Service DLL. Advisory CA-2001-19, July 2001.

[5]. M. Roesch. Snort - Lightweight Intrusion Detection for Networks. In Proceedings of the USENIX LISA '99 Conference, November 1999.

[6]. Carrier, B.D., Spafford, E.H.: Defining event reconstruction of digital crime scenes. J. Forensic Sci. 49 (2004)

[7]. Carrier, B.: An event-based digital forensic investigation framework. In: Digital forensic research workshop (2004)

[8]. Chisum, W.J., Turvey, B.E.: Evidence dynamics: Locard's exchange principle crime reconstruction. J. Behav. Profiling 1(1) (2000)

[9]. W. Vogels, "Eventually Consistent," ACM Queue, 4 Dec. 2008; http://queue.acm.org/detail.cfm?id=1466448.

[10]. Stephenson, P.: Formal modeling of post-incident root cause analysis. Int. J. Digit. Evid. 2 (2003)

[11]. Gladyshev, P., Patel, A.: Finite state machine approach to digital event reconstruction. Digit. Invest. 1 (2004)

[12]. Stallard, T.B.:Automated analysis for digital forensic science. Master's thesis, University of California, Davis (2002)

[13]. Stallard,T.,Levitt,K.N.:Automated analysis for digital forensic science: Semantic integrity checking. In: ACSAC 160–169 (2003)

[14]. Abbott, J., Bell, J., Clark, A., Vel, O.D., Mohay, G.: Automated recognition of event scenarios for digital forensics. In: SAC '06: Proceedings of the 2006 ACM symposium on applied computing pp. 293–300.ACMPress,NewYork (2006)

[15]. Elsaesser, C., Tanner, M.C.: Automated diagnosis for computer forensics. Technical report, The MITRE Corporation (2001)

[16]. Neuhaus, S., Zeller, A.: Isolating intrusions by automatic experiments. In: Proceedings of the 13th annual network and distributed system security symposium. pp. 71–80 (2006)

[17]. Khan M, Chatwin C, Young R. A framework for post-event timeline reconstruction using neural networks. Digital Investigation 2007;4: 146–57.

[18]. Olsson J, Boldt M. Computer forensic timeline visualization tool. Digital Investigation 2009;6(S1):S78–87.

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

[20]. Bunting. EnCE study guide; 2008. pp. 235–237.

[21]. Carbone R, Bean C. Generating computer forensic super-timelines under Linux; 2011.

[22]. Buchholz F, Falk C. In: DFRWS, editor. Design and implementation of Zeitline: a forensic timeline; 2005.

[23]. Rohit Chandrawanshi, Hitesh Gupta, "A Survey: Server TimeLine Analysis For Web Forensics", International Journal of Scientific Research Engineering & Technology, Volume 1, Issue 12, March 2013.