I. J. Computer Network and Information Security, **2015, 3, 10-17**
Published Online February 2015 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2015.03.02

Forensics Investigation of Web Application Security Attacks

Amor Lazzez, Thabet Slimani
College of Computers and Information Technologies, Taif University, Kingdom of Saudi Arabia
Email: a.lazzez@gmail.com, thabet.slimani@gmail.com

Abstract—Nowadays, web applications are popular targets for security attackers. Using specific security mechanisms, we can prevent or detect a security attack on a web application, but we cannot find out the criminal who has carried out the security attack. Being unable to trace back an attack, encourages hackers to launch new attacks on the same system. Web application forensics aims to trace back and attribute a web application security attack to its originator. This may significantly reduce the security attacks targeting a web application every day, and hence improve its security. The aim of this paper is to carry out a detailed overview about the web application forensics. First, we define the web applications forensics, and we present a taxonomic structure of the digital forensics. Then, we present the methodology of a web application forensics investigation. After that, we illustrate the forensics supportive tools for a web application forensics investigation. After that, we present a detailed presentation of a set of the main considered web application forensics tools. Finally, we provide a comparison of the main considered web application forensics tools.

Index Terms—Web application, Security Attack, Forensics Investigation.

I. Introduction

With the continuous evolution of the development and networking technologies, Web applications become a fundamental mean of information transmission and management in government agencies, enterprises and individuals. Actually, nowadays, the web technology underlies the majority of recent network applications such as the e-commerce, e-banking, e-learning, e-medicine, e-mail, etc. Being deployed in various critical sectors as the marketing, the trading, and the banking areas, the web applications become popular targets for security attackers. Moreover, the variety of dependencies upon which a web application relies multiplies its vulnerabilities [1, 2]. These dependencies include the network infrastructure, the web server, the database servers, the web browsers, and the operating systems upon which servers are installed [3]. Fig.1 shows the interactions between the different components of a web application and shows where vulnerabilities may affect a web application.

The above presented analysis shows that web applications constitute a motivating environment for attackers to perform security attacks. This involves the development of various methods to perform a security attack on a web application. The famous are: Cross-Site Scripting, SQL injection, Code Injection, and Buffer Overflow [1]. As long as web applications constitute the most important mean of data communication over the Internet, different techniques have been developed to protect web applications against hackers. Firewalls and systems' security patching are used for attack prevention; intrusion detection systems and antivirus are used for attack detection [1, 4].

Based on the proposed attack detection schemes, we can detect that a web application has been attacked, but it is hard to find out the criminal who has carried out the security attack. Being unable to trace and follow up a hacker, attackers may always conceal themselves and launch new attacks. Therefore, it is crucial to build the capability to trace and attribute attacks to the real cyber criminals, which may significantly reduce the attacks we face every day.

Tracing back a security attack on a web application in order to identify, where an attack has been originated, how it was propagated, and what computer(s) and person(s) are responsible refers to as web application forensics [5, 6, 7]. Dealing with security attacks on web application, the web application forensics constitutes a specific branch of the digital forensics that deals with cyber criminals in general [8, 9].

To trace back a security attack on a web application, a forensics investigator relies on the fingerprints (digital evidences) left by the hacker on the crime scene, and which are recorded in the different configuration and log files of the various components upon which relies the web application [3, 9, 10]. The digital evidences needed to investigate a web application attack may be gathered form one of the following files: web server(s) and application server(s) logs, server side scripts which are used by the web application, web server(s) and application server(s) configuration files, any third party installed software logs, and the operating system logs [9, 10].

While the above mentioned files constitute the main source for digital evidences collection, they sometimes lack data needed to conduct a forensics investigation of a given security attack [9, 10]. To overcome this issue, supportive forensics tools should considered to help the collection of the needed digital evidence that cannot be offered by the logging options of a web application. A required digital evidence may be provided by a network or an operating system forensics tool or by a web application forensics tool offering extra logging facilities [9, 10].

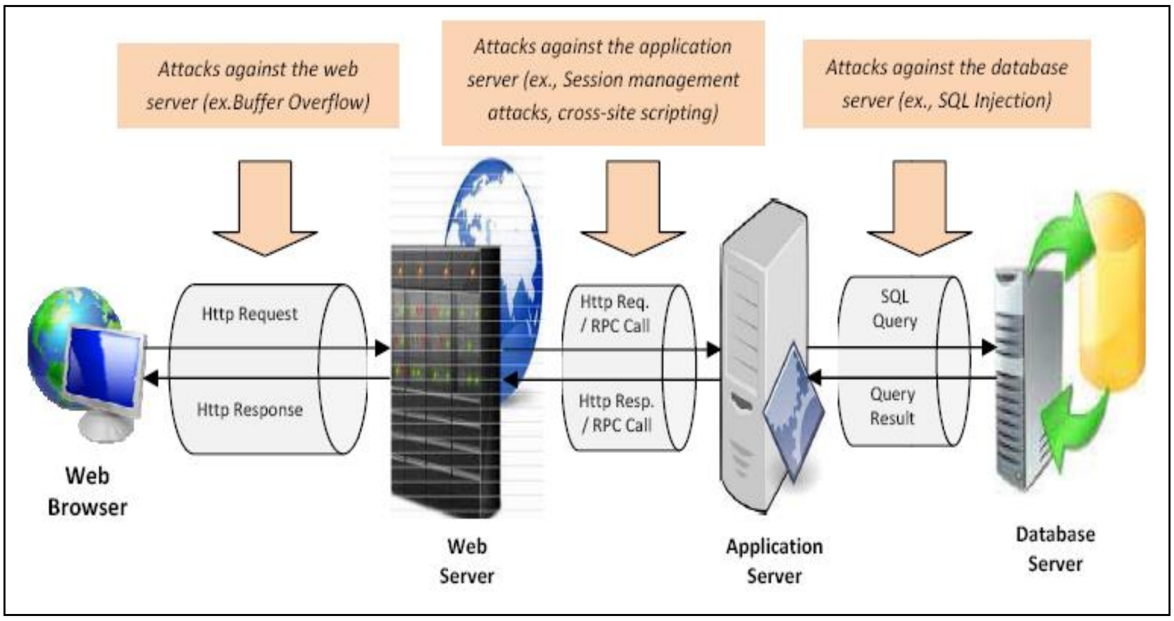

Attacks against the web server (ex.Buffer Overflow)
Attacks against the application server (ex., Session management attacks, cross-site scripting)
Attacks against the database server (ex., SQL Injection)
Http Request
Http Response
Http Req. / RPC Call
Http Resp. / RPC Call
SQL Query
Query Result
Web Browser
Web Server
Application Server
Database Server

Fig.1: Web Application Architecture

To effectively perform web application forensics, several techniques have been proposed to help an efficient management of the different sources of digital evidences. Referred to as web application forensics tools, these techniques provide an efficient analysis of the data they contain. Microsoft LogParser, EventLog Analyzer, Pyflag, Http-analyze, Analog, Open Web Analytics, Mywebalizer, CORE Wisdom, Logjam, Sawmill, and Lire are examples of the main tools used investigate a web application security attack [3, 9, 16-25]. In addition to the use of a specific forensics tool, following a standard methodology is crucial for a successful and effective forensics investigation of a web application security attack [9, 10].

The aim of this paper is to carry out a detailed overview about the web application forensics; a topic that aims to improve the security of web applications through the tracking and persecution of hackers. First, we define the web applications forensics, and we distinguish the different branches of the digital forensics. Then, we present the methodology that should be followed to help a successful accomplishment a forensics investigation of a web application security attack. After that, we provide a detailed presentation of a set of techniques proposed to help a successful accomplishment of a forensics investigation of a hacked web application. Finally, we provide a technical comparison of the main considered web application forensics tools.

The remaining of this paper is organized as follows. Section 2 defines the web application forensics and distinguishes it from the other branches of the digital forensics. Section 3 presents a general methodology of a forensics investigation of a web application security attack. Section 4 provides a detailed overview about the main web application forensics tools. Section 5 concludes the paper.

II. Web Application Forensics

Web application forensics aims to trace back and attribute a security attack on a web application to its originator [5-7]. Dealing with security attacks on web application, the web application forensics constitutes a specific branch of the digital forensics that deals with cyber criminals in general [8-9]. In addition to web application forensics, digital forensics incorporates other branches such as Operating System Forensics, Digital Image Forensics, and Network Forensics [8-9].

Web Application Forensics aims to trace back and attribute a security attack on a web application to its originator. To trace back a security attack, web application forensics mainly relies on the analysis of the log files of the different components of a web application (web browser, web server, database servers, application server) [3, 10]. Web application forensics does not matter with the analysis of network level protocols and components which is in the focus of Network Forensics [6, 9]. Yet, examining the log files of the network level equipments (IP Routers, IP Switches, Intrusion detection systems, Firewalls, etc.) may be helpful for the accomplishment of a web application forensics investigation. Therefore, a web application forensics investigator should consider the supportive function of the network forensics tools towards a successful investigation of a security attack on a web application.

The operating system forensics deals with the analysis of system log files towards the investigation of a system alteration [8]. Whereas, the digital image forensics considers the image manipulations [8]. As the network forensics tools, the operating systems and digital image forensics tools constitute an appropriate support for a successful deployment of a web application forensics investigation [7-9].

At last, we should note that the web application forensics does not deal explicitly with security attacks on web services [8-9]. The forensics investigation of a security attack on a web is covered the Web Services Forensics; another branch of the digital forensics that should be discerned from the web application forensics. Given that the web applications and web services are both parts of the Cloud-Computing concept which refers to the Internet based applications and services, the authors in [9, 11, 26] define the Cloud-computing Forensics; a novel branch of the digital forensics which integrates the web application forensics and the web services forensics. Fig.2 presents the taxonomic structure of the digital forensics as it is presented in [9, 11]. The figure illustrates that the digital forensics (DF) is subdivided into four branches to know, the network forensics (NF), the operating systems forensics (OSF), the digital image forensics (DIF), and the cloud-computing forensics (CCF). The cloud-computing forensics (CCF) is in its turn decomposed into two sub-branches: the web application forensics (WAF), and the web services forensics (WSF).

III. Forensics Investigation of a Web Application Security Attack

A successful forensics investigation relies on a preliminary analysis phase and needs to follow a standard methodology [7, 9, 10, 12, 13]. In the following subsections, we first present the preliminary actions that should be considered towards a successful accomplishment of a web application forensics investigation. Then, we present the steps that should be flowed by a forensics investigator to conduct a thorough analysis of the hacking attempt. Finally, we illustrate how network, digital image, and operating systems forensics may support the achievement of a web application forensics investigation through the provision of further evidence.

A. *Preliminary Analysis*

The following preliminary actions are required for a successful forensics investigation of a security attack on a web application [7, 9, 12, 13]:

Application Forensics Readiness: The web application should be well prepared for a forensics investigation. This is may be reached by:

<u>Evidence collection:</u> To prepare a web application for an eventual forensics investigation, it is highly recommended to enable the logging options to collect the maximum of digital evidences. If the logging options are left at the default settings, the evidence collection will be incomplete and the application will not be ready for a forensic investigation.

<u>Evidence protection:</u> given that the log files will constitute the main source of digital evidence to perform a forensics investigation, it is crucial to protect these files to ensure the integrity of the data they contain, and hence guarantee the accuracy of the digital evidences they provide. The following actions may be considered to protect the log files:

- Setting the proper permissions to the log files.
- Keeping the log files out of the hacker’s reach. This can be done by using some sort of backup utility, which will save the log files on a remote server.
- Using some sort of checksum in order to verify the log files integrity.

Supportive forensics: The forensics readiness of a web application ensures the collection of the maximum of digital evidence. But, it does not guarantee the existence of all digital evidences required by a forensic investigation. Therefore, supportive forensics tools should be used to help the collection of the needed digital evidence that cannot be offered by the logging options of a web application. A digital evidence required to perform a forensic investigation of a web application security attack may be provided by a network or an operating system forensics tool or by a third party offering extra logging facilities. More details about web application supportive forensics are presented in section IV.

Forensics investigator abilities: the forensics investigator should

- Have a good understand of web application: architecture, components, intended application flow, etc.
- Have a good understand of the security issues of web applications: vulnerabilities, security attack methods, etc.
- Well trained for forensics investigation.

B. *Methodology*

Following a standard methodology is crucial to perform a successful forensics investigation of a web application security attack. In order to conduct a thorough analysis of a web application security attack, a forensics investigator should follow the following methodology steps [7, 9, 10, 12]:

1. Protect the web application (could be several servers) during the forensic examination to prevent any modification of the evidence files.
2. Discover all files needed for the forensics investigation. This includes:

- Web server(s) and application server(s) logs.
- Server side scripts which are used by the web application.
- Web server(s) and application server(s) configuration files.
- Any 3rd. party installed software log files.

- Operating system log files.

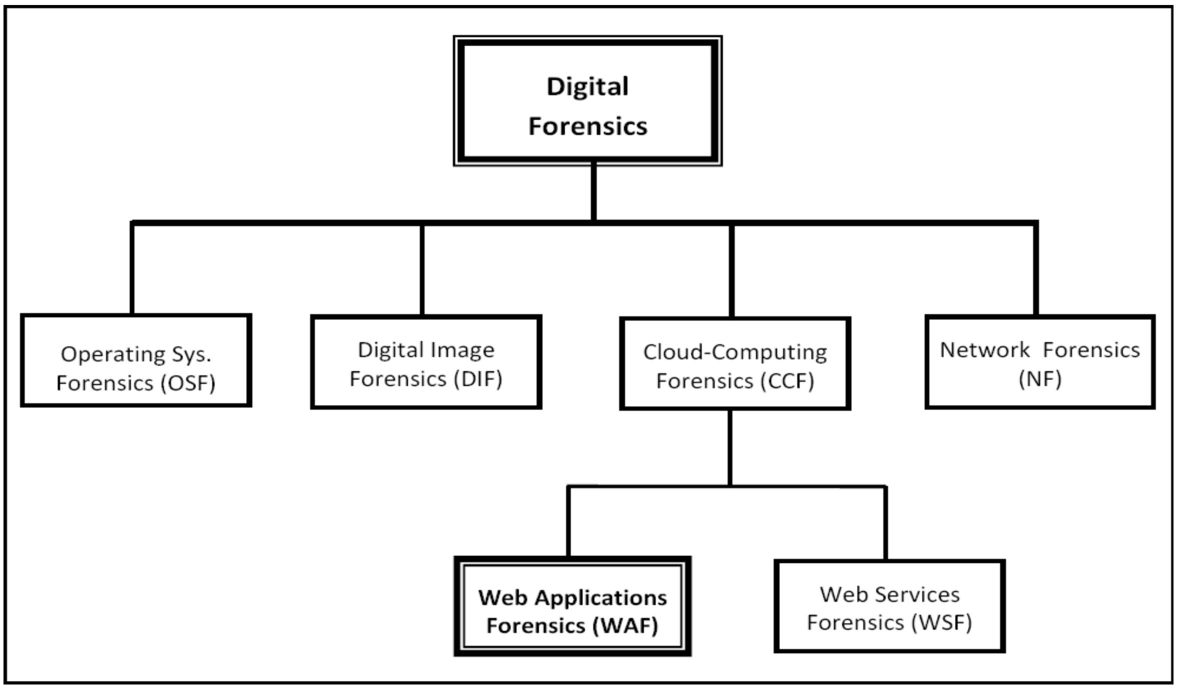

Digital Forensics
Operating Sys. Forensics (OSF)
Digital Image Forensics (DIF)
Cloud-Computing Forensics (CCF)
Network Forensics (NF)
Web Applications Forensics (WAF)
Web Services Forensics (WSF)

Fig.2: The Taxonomic Structure of the Digital Forensics

3. Perform a forensics analysis of the considered files to determine the sequence of events and the degree of compromise. During this step, the forensics investigator should divide the log files according to user sessions, which may give better understanding of the session flow and timeline, and remove noise created by other users in the log files. During the analysis of a user session flow, the forensics investigator should be alerted by any of the fingerprints of a web application security attack. The following are examples of fingerprints and patterns left by web application hacking attempts:

- Unusual entries in the Logs (GET requests to ASP pages which normally receive POST requests).
- Script abuse (CMD.exe, Root.exe, Upload. ASP).
- Excessive attempts from the same IP address.
- Unusually long processing times (SQL Injection attempt)
- Files created or modified around the time of the suspected attack.
- Etc.

4. Prepare a report based on the data extracted from the web application.
5. Recommend post event actions.

C. *Supportive Forensics*

The aim of this subsection is to show how network forensics, digital image forensics, and operating systems forensics may support the achievement of a web application forensics investigation through the provision of further evidence [9]. Actually, the log data derived from an intrusion detection system may help a more accurate detection of an intruder's activities on a web application [6, 9]. Moreover, the forensics investigations on digital images uploaded to a compromised web application may, in some case, assist the attribution of the intrusion to its originator [7, 9]. Finally, we should also note that the digital evidence collected from the cache memory of a hacked web application server, that has not been restarted during or after the attack scenario may be helpful to perform a successful forensics investigation of the considered attack even in the case of the lack of sufficient digital evidence in server's log files [9, 14].

IV. Web Applications Forensics Tools

Given the huge amount of the logged data that need to be examined during a web application forensics investigation, automated tools have been proposed towards a successful deployment of the web application forensics [3, 7, 9, 10, 16-25].

In the following subsections, we first present the main requirements for a web application forensics tool. Then, we present a brief overview of the most important web application forensics tools. Finally, we present a comparison of the presented web application forensics tools.

A. *Requirements for a web application forensics tool*

A detailed presentation of the main requirements for a web application forensics tool is presented in [9, 15]. The following are the basic requirements of a web application forensics tool:

- Analyze log files in different formats.

- Take two independent and differently formatted evidence files and combine them.
- Handle big log files.
- Utilize regular expressions and binary logic on any observed parameter in a log file.
- Perform normalization by time to consider a proper investigation on time-stamps.
- Maintain a list of suspicious requests, which should indicate a potential compromise.
- Utilize, decoding of URL data so that, it can be searched easier in readable format.

B. *Web Application Forensics Tools*

In the following subsections, we present a brief overview of the most important web application forensics tools: Microsoft LogParser, EventLog Analyzer, Http-analyze, Pyflag, Analog, Open Web Analytics, Mywebalizer, CORE Wisdom, Logjam, Sawmill, and Lire[3,9, 16-25].

Microsoft LogParser: Developed by Microsoft, LogParser [3, 9] is a flexible command line utility that provides universal query access to text-based data such as log files (web server log files, DNS log files, HTTP error log files), XML files, W3C files, TSV files and CSV files, as well as key data sources on the Windows operating system such as the Event Log, the Registry, the file system, and Active Directory. Logparser produces output in standardized formats such as CSV, TSV, XML, Syslog, W3C, IIS, SQL, and non-standard formats, which require either immediate presentation or include graphical output such as DATAGRID, CHART and NAT. Logparser does not provide a graphical user interface, but provides functionality through a command line invocation by script, or through a direct manipulation of queries by prompt interface. For a more convenient use of the Logparser tool, two programs that provide graphical user interface and graphical outputs have been developed; Logparser Lizard and Visual Logparser. The Logparser language includes a set of functions that perform string manipulation, arithmetic operations, and provide access to system details. Each of these functions can modify or manipulate the content of fields in some manner. The log file conversion capabilities of Logparser help the adaptation of log files to queries for performing analysis including correlation. For correlation, Logparser has the capability of combining the data from multiple sources, and then performs queries upon it. Logparser has been used to monitor user activities, monitor system file integrity, check for SQL injection attacks, check for excessive failed logon attempts, determining malicious modification, identification of brute force attacks, and reconstructing intrusions. As a limitation of this tool, Logparser does not include methods of analysis, only the strength to perform the queries. The user must create useful queries to satisfy any analysis requirements.

EventLog Analyzer: EventLog Analyzer [3, 16] is a real-time web-based log monitoring and compliance management solution for security information and event management that improves internal network security. EventLog Analyzer can collect, analyze, search, report, and archive an extensive array of machine generated logs received from Systems (Windows, Linux, UNIX…), network devices (routers, switches, etc…), applications (Oracle, Apache, IIS, etc…) and then provides important insights into network user activities, policy violations, network anomalies, system downtime, and internal threats. EventLog Analyzer can be used to generate archive files, which can be stored for later analysis. It can also define automatic alerts, generate historical trends based on system events, group host information together to show interactions, show failed logins, malicious users and show applications that are causing performance or security issues. In addition, EventLog Analyzer includes prebuilt reports and allows the choice of the appropriate data and format for the generation of custom reports and templates. Reports are output at specific intervals and can be exported to HTML, PDF and Comma Separated Values (CSV) formats. The analysis can generate both graphs and text-based representations as output. EventLog Analyzer performs many qualities essential to a forensic log analysis tool. However, it does not include automatic correlation between log files, or extensibility with regards to the types of log files that may be analyzed.

Http-analyze: Http-analyze [3, 17] is a multiplatform log file analyzer that can process data in only three log file formats, the CLF, ELF and Distilled Log Format (DLF). The http-analyze tool provides the option to generate one of two different standardized HTML reports to include statistical and access load information summaries. These reports include graphs, tabulated data, and three dimensional forms. Real-time analysis is only achieved by scripting the rotation of the log files, used in conjunction with the automated calling of this tool. Http-analyze forensics tool does not generate or format the log file information in any way. It does not put the information into a database, nor does it perform correlation between the web server files with any other system available information.

Pyflag: Pyflag [3, 18] is an open source web-based application that allows a forensics analysis of log files through an extensive Graphical User Interface (GUI). This tool is capable of handling large volumes of log files in many different file formats, disks or images, and network traffic data such as Tcpdump data. Data can be added to a MySQL database for faster queries but the log types are still specified by the end user and due to the fact that this tool's basic log function is only to view the log files, the analyst must perform the required analysis using prior knowledge and experience. The interface for analysis of log data includes querying, sorting and graphic representation of the log data. In addition to the statistical log analysis functionality, Pyflag provides the option to analyze many formats other than just log files pertaining to web applications.

Analog: Analog [3, 19] is an open source web log file analyzer that may accept AWS or IIS W3C formatted files as input. As output, Analog produces complex graphs and report styles when used in conjunction with another tool called Report Magic. Statistical reporting or the conversion of statistical information into a graphical representation forms the basic functionality of this web application forensics tool. Analog generates general summaries, time-based reports, host, domain and organization reports, file-based, browser-based as well as user and status-based reports. The validity of information retained in these reports is only maintained when the necessary server configuration is performed. Analog requires extra configuration on the part of the analyst to provide a concise account detailing information that would be beneficial to both marketers and forensic investigators.

Open Web Analytics: Open Web Analytics [3, 20] platform is a generic web analytics framework that provides analytical data regarding any web application. Due to its nature as a web application, Open Web Analytics can function on any operating system that contains a browser and it can easily be added to existing popular web applications using JS, PHP or REST application programming interfaces. Open Web Analytics provides built in support for Word press or MediaWiki applications. The Open Web Analytic tool ’ s main function is to provide real-time tracking, monitoring and reporting of web usage statistics. Some examples of information it can provide include visitor click streams, geolocation of visitors, browser information and web application specific features. It also provides the means to develop additional functionality through the use of plugins.

Mywebalizer: Mywebalizer [3, 21] is an open source that generates highly detailed and easily configured HTML reports about web server usage statistics in both tabular and graphical formats. Entirely written in the C programming language, Mywebalizer tool is extremely portable to the different UNIX operating systems (Linux, AIX, Solaris, etc.). Mywebalizer may analyze different log file formats including CLF, Xferlog, and Extended W3C formats . In addition, this tool provides decompression capabilities so that bzip2 and gzip compressed logs (bzip2 and gzip) may be used directly without the need for uncompressing which saves on space, and provides analysis compatibility for larger log files. This tool does not provide automatic detection of log file type, nor does it run in real-time or provide any correlation capabilities.

CORE Wisdom: CORE Wisdom [3, 22] is mainly characterized by its ability to generate unique graphical log analysis report beyond mere pie charts and rudimentary graphs. This provides the necessary understanding to enhance evidentiary reports prepared for the analysis of log data. The main problem with this log analyzer is that the analyst has to define the rules for importing the log file itself. CORE Wisdom does not provide any correlation ability. The analysis this tool performs is accomplished in real-time. Hence, the analyst needs to be able to define events that should be flagged as alarms, and to know what visual cues to look for.

Logjam: Logjam [3, 23] is a web traffic analyzer that provides tailed d statistical analysis of W3C ELF log files. It performs only on a MS Windows server that includes Active Server Pages (ASP) and Microsoft Data Access Components (MDAC). Preliminary to analysis, log files are jammed into an SQL database for analysis. Logjam includes default reports and a customizable report generator that creates SQL queries based on user preferences. This web traffic analyzer has neither implemented correlation nor does it performs in real-time.

Sawmill: Sawmill [3, 24] is sophisticated log file analyzer that includes plugins allowing the detection of over 800 distinct log file types, and provides a method for defining plugins for non standard log file types. The architecture of this tool includes log importers, a database, a reporting interface, a web server, a command line interface, a scheduler, and two languages used to manipulate data and how data is stored for analysis. Referred to as Salang, the first Sawmill language is used to display pages and to define log filters. As C language, Salang contains the elements necessary for the definition of filters including regular expressions, and conditional logic. The second language is the Sawmill structured query language. This language is a subset of most SQL commands, and is utilized to access internal database information within predefined table sets of the Sawmill database. To provide a forensically sound basis, external methods protecting the original log data are used. Sawmill provides the access necessary to perform correlation between log data sources, and performs in real-time.

Lire: Compared to the above presented log analysis tolls, Lire [3, 26] can provide analysis for a wide variety of log file types through the use of conversion tools that convert the log files to the DLF log format (Distilled Log Format). This web forensics tool generates statistical analysis reports according to one of thirteen default report templates that can be modified and customized by the analyst. Lire does not provide correlation among log files, nor it performs in real-time.

C. Web Application Forensics Tools Comparison

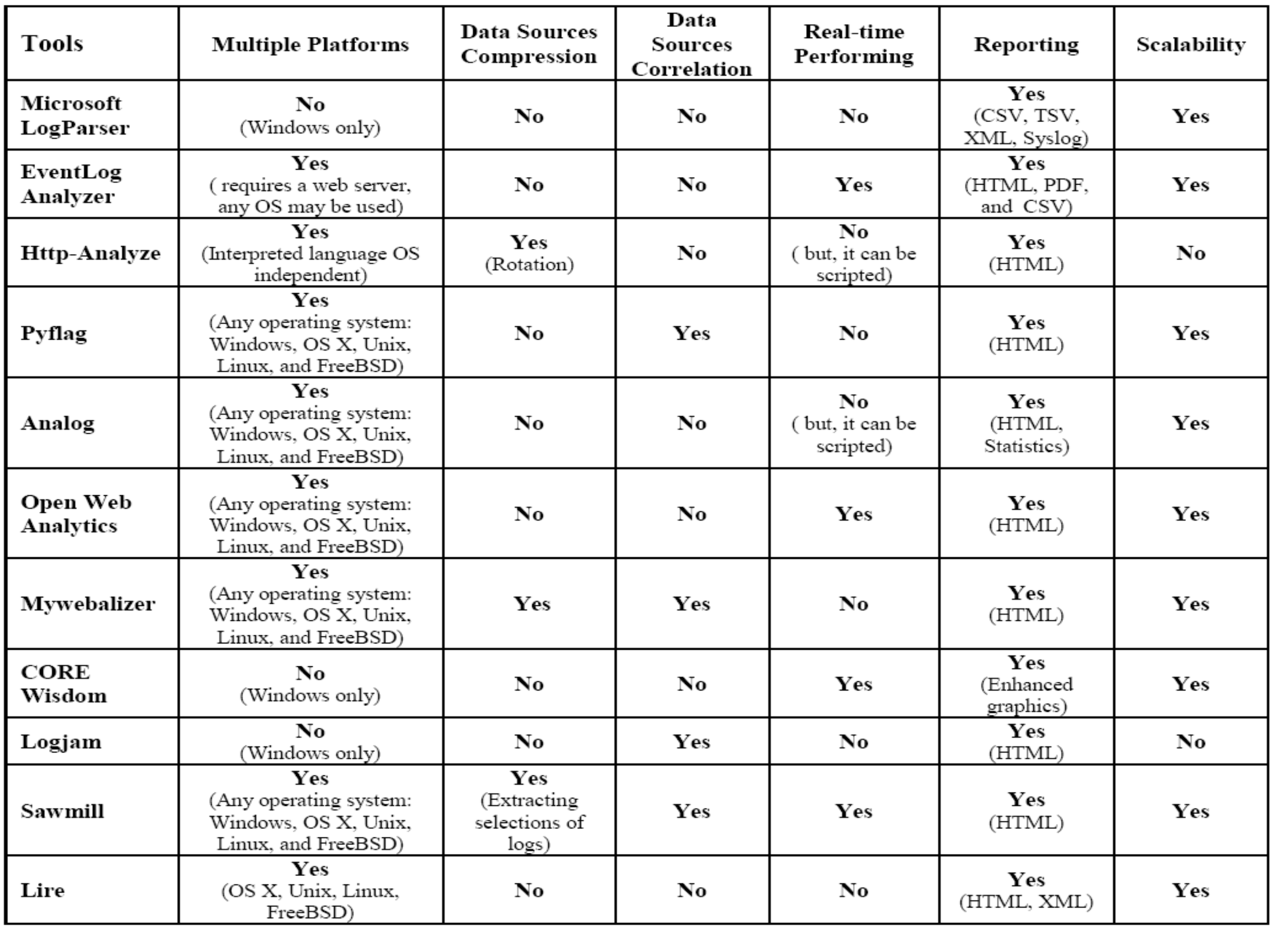
Table 1 presents a comparison of the above presented web application forensics tools as a part of the comparison of the log analysis tools presented in [3]. The following parameters have been considered in the presented comparison:

- Multiple Platforms: tool’s ability to be performed over various operating systems.
- Data Sources Compression: tool’s ability to compress the considered log files
- Data Sources Correlation: tool’s ability to correlated different evidence sources
- Real-time Performing: tool’s ability to perform a real-time analysis log files
- Reporting: tool’s ability to generate an analysis report
- Scalability: tool’s ability to continue to function well as its context is changed in size or volume.

V. Conclusion

In this paper, we have presented a detailed overview about the web application forensics. First, we defined the web applications forensics as a sub-class of the digital forensics. Then, we presented the appropriate methodology that should be followed to help a successful accomplishment of a forensics investigation of a web application security attack. After that, we discussed how network forensics, digital image forensics, and operating systems forensics may support the achievement of a web application forensics investigation through the provision of further evidence. Next, we presented a detailed presentation of the main considered tools to help a successful accomplishment of a forensics investigation of a hacked web application. Finally, we provided a technical comparison of the main considered web application forensics tools.

Table 1: Web Application Forensics Tools Comparison



Tools	Multiple Platforms	Data Sources Compression	Data Sources Correlation	Real-time Performing	Reporting	Scalability
Microsoft LogParser	No (Windows only)	No	No	No	Yes (CSV, TSV, XML, Syslog)	Yes
EventLog Analyzer	Yes (requires a web server, any OS may be used)	No	No	Yes	Yes (HTML, PDF, and CSV)	Yes
Http-Analyze	Yes (Interpreted language OS independent)	Yes (Rotation)	No	No (but, it can be scripted)	Yes (HTML)	No
Pyflag	Yes (Any operating system: Windows, OS X, Unix, Linux, and FreeBSD)	No	Yes	No	Yes (HTML)	Yes
Analog	Yes (Any operating system: Windows, OS X, Unix, Linux, and FreeBSD)	No	No	No (but, it can be scripted)	Yes (HTML, Statistics)	Yes
Open Web Analytics	Yes (Any operating system: Windows, OS X, Unix, Linux, and FreeBSD)	No	No	Yes	Yes (HTML)	Yes
Mywebalizer	Yes (Any operating system: Windows, OS X, Unix, Linux, and FreeBSD)	Yes	Yes	No	Yes (HTML)	Yes
CORE Wisdom	No (Windows only)	No	No	Yes	Yes (Enhanced graphics)	Yes
Logjam	No (Windows only)	No	Yes	No	Yes (HTML)	No
Sawmill	Yes (Any operating system: Windows, OS X, Unix, Linux, and FreeBSD)	Yes (Extracting selections of logs)	Yes	Yes	Yes (HTML)	Yes
Lire	Yes (OS X, Unix, Linux, FreeBSD)	No	No	No	Yes (HTML, XML)	Yes

REFERENCES

[1] Mike Shema, "Hacking Web Apps", Publisher: Syngress, Pub. Date: October 2012, Print ISBN-13: 978-1-59749-951-4, Web ISBN-13: 978-1-59749-956-9, Pages in Print Edition: 296.

[2] Ivan Ristic, "Apache security", O'Reilly Media, Inc., pub-ORA-MEDIA: adr, 2005. Section 1.1.2,.

[3] Ann Fry, "A Forensic web Log Analysis Tool: Techniques and implementation", thesis dissertation, department of Concordia Institute for Information Systems Engineering, Concordia University, Montreal, Canada, September 2011, web site: http://spectrum.library.concordia.ca/7769/1/Fry_MASc_F 2011.pdf. Last accessed on October 2014.

[4] Prof. Dr. David Basin, Dr. Patrick Schaller, and Michael Schläpfer, "Web Application Security", Applied Information Security, 2011, ISBN 9783642244735, pp. 81 – 101.

[5] Vimal Kumar, Akhilendra Pratap Singh, Anjani K. Rai , Manoj Wairiya, "Self Alteration Detectable Image Log File for Web Forensics. In International Journal of Computer Applications, 2011.

[6] Natarajan Meghanathan, Sumanth Reddy Allam and Loretta A. Moore, "Tools and Techniques for Network Forensics", International Journal of Network Security & Its Applications (IJNSA), Vol .1, No.1,April 2009.

[7] Jess Garcia, "Web Forensics", 2006. Web site: http://www.jessland.net. Last accessed on October 2014.

[8] Farhood Norouzizadeh Dezfoli, Ali Dehghantanha, Ramlan Mahmoud, Nor Fazlida Binti Mohd Sani, and Farid Daryabar, "Digital Forensic Trends and Future", International Journal of Cyber-Security and Digital Forensics (2): 48-76 The Society of Digital Information and Wireless Communications, 2013 (ISSN: 2305-0012).

[9] Krassen Deltchev, "Web Application Forensics: Taxonomy and Trends", term paper, Horst Görtz Institute, September 2011, web site: http://fr.slideshare.net/test2v/web-application-forensics-taxonomy-and-trends. Last Accessed on October 2014.

[10] Ory Segal, Sanctum Security Group, "Web Application Forensics: The Uncharted Territory", Sanctum 2002, web site: http://www.sanctuminc.com/pdf/WhitePaper_Forensics.pdf. Last accessed on October 2014
[11] Dominik Birk, "Forensic Identification and Validation of Computational Structures in Distributed Environments", 2010.
[12] Rohyt Belani, Chuck Willis, "Web Application Incident Response & Forensics: A Whole New Ball Game!", 2007
[13] Jess Garcia, "Proactive & Reactive Forensics", 2005. Web site: http://www.jessland.net. Last accessed on October 2014.
[14] Edgar Weippl, "Database Forensics", Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), 2010, Perth, WA.
[15] Robert Hansen, Web Server Log Forensics App Wanted", web site: http://ha.ckers.org/blog/20100613/web-server-log-forensics-app-wanted. Last accesssed on october 2014.
[16] Manage Engine. EventLog Analyzer 6 Managed Server Guide. ZOHO Corporation, 4900 Hopyard Rd, Suite 310 Pleasanton, CA 94588, USA, build 6.2.0 edition, January 2009.
[17] RENT-A-GURU. HTTP-ANALYZE - A Logfile Analyzer for Web Servers, January 2011. Web site: http://http-analyze.org/index.php. Last Accessed on October 2014.
[18] Michael Cohen. PyFlag - PyFlagWiki. http://www.pyflag.net/cgi-bin/moin.cgi, April 2010. Last accessed on October 2014.
[19] Stephen Turner. Analog: WWW logfile analysis. http://www.analog.cx/, June 2005. Last Accessed on October 2014.
[20] Peter Adams. Open Web Analytics - Main Page. http://www.openwebanalytics.com/, December 2010. Last Accessed on October 2014.
[21] The Webalayzer, web site: http://www.webalizer.org/. Last accessed on December 2014.
[22] Core Security Technologies. *Operational Documentation Core Wisdom*. "web site: http://www.coresecurity.com/open-source-projects#wisdom". Last accesed on December 2014.
[23] NEWMAN Services Corp., "LogJam - Web Traffic Analysis", Web site: http://newmanservices.com/logjam/pages/about.asp. Last Accessed December 2014.
[24] Sawmill, "Sawmill: Universal log file analysis and reporting", web site: http://www.sawmill.net/. Last accessed on December 2014.
[25] Ubunto manual, "Lire", web site: http://manpages.ubuntu.com/manpages/hardy/man7/lire.7.html. Last accessed on December 2014.
[26] Seyyed Yasser hashemi, and Parisa Sheykhi Hesarlo, "Security, Privacy and Trust Challenges in Cloud Computing and Solutions", I.J. Computer Network and Information Security, 2014, 8, 34-40.

Authors' Profiles

Amor Lazzez is currently an Assistant Professor of Computer and Information Science at Taif University, Kingdom of Saudi Arabia. Dr. Lazzez received the Engineering diploma with honors from the high school of computer sciences (ENSI), Tunisia, in June 1998, the Master degree in Telecommunication from the high school of communication (Sup"Com), Tunisia, in November 2002, and the Ph.D. degree in information and communications technologies form the high school of communication, Tunisia, in November 2007. His main areas of research include the design and analysis of all-optical network architectures and protocols, QoS support, VoIP deployment, network security, and digital forensics.

Thabet Slimani: got a PhD in Computer Science (2011) from the University of Tunisia. He is currently an Assistant Professor of Information Technology at the Department of Information Technology of Taif University at Saudia Arabia and a LARODEC Labo member (University of Tunisia), where he is involved both in research and teaching activities. His research interests are mainly related to Semantic Web, Data Mining, Business Intelligence, Knowledge Management and recently Web services. Thabet has published his research through international conferences, chapter in books and peer reviewed journals. He also serves as a reviewer for some conferences and journals.

How to cite this paper: Amor Lazzez, Thabet Slimani,"Forensics Investigation of Web Application Security Attacks", IJCNIS, vol.7, no.3, pp.10-17, 2015.DOI: 10.5815/ijcnis.2015.03.02