Analysis of Methodologies of Digital Data Collection in Web Servers

Mónica D. Tugnarelli [1], Mauro F. Fornaroli [1], Sonia R. Santana[1],
Eduardo Jacobo [1], Javier Díaz [2]

[1] Facultad de Ciencias de la Administración – Universidad Nacional de Entre Ríos.
Av. Tavella 1400, Concordia (3200), Entre Ríos, Argentina
[2] Facultad de Informática – Universidad Nacional de La Plata. 50 y 120, La Plata (1900),
Buenos Aires, Argentina.
[1] ✉ montug@fcad.uner.edu.ar, maufor@fcad.uner.edu.ar

Abstract. When an incident or security threat occurs, in which a system resource is compromised or potentially exposed to unauthorized access, computer forensics techniques and methodologies must ensure that it is possible to adequately determine what, who, when and how the incident occurred, as well as to ensure and preserve the evidence collected. This paper explore two methodologies of digital data collection, the first called Preventive Approach- Data Collection a priori or Forensic Readiness and the second called Reactive Approach - Post-Collection of a security event to comparatively analyze its performance based on certain criteria and control points established over HTTP and HTTP/2 web servers.

Keywords: Security, Incident, Forensia, Methodologies, HTTP.

1 Introduction

If an IT security architecture is correctly defined, it must provide a plan and set of policies that describe both the security services offered to users and the system components required to deploy those services. These security policies are applied to the information assets identified for their relevance to the organization's goals, knowing how they are managed and what are their risks to implement strategies and mechanisms that ensure confidentiality, integrity and availability of those information assets [1].

When a security incident or threat occurs, in which a system resource is compromised or potentially exposed to unauthorized access, this security architecture is violated. Generally speaking, as environmental threats, aspects ranging from administrative security, communications security, environmental security to physical security can be considered. Therefore, the security architecture must be able to deal with both intentional and accidental threats. The implementation of a systematic incident monitoring and management program, based on the use of methodologies, can provide a structured and organized approach to minimize the impact of the security incident and help to deliver a rapid and adequate response.

Every day hundreds of teams are exposed to potential incidents, consider as an example the advance of Internet of Things (IoT) and its working characteristics emission of possibilities and risks of an incident and its consequent impact. [2], [3]

Computer forensics methodologies must ensure that it is possible to determine adequately what, who, when and how happened in relation to that security incident, as well as to take care of the preservation and traceability of the collected data.

The definition offered by the first Digital Forensics Research Workshop (DFRWS), states that the digital forensic analysis or computer forensics is "*The use of scientifically proven and derived methods towards the preservation, collection, validation, identification, analysis, interpretation, documentation, and presentation of digital evidence derived from digital sources in order to facilitate or promote the reconstruction of facts, which may constitute legal evidence, or helping to anticipate unauthorized actions shown to be disruptive to planned operations*" [4]

The data sources are numerous, ranging from computers, cell phones, digital camera cards, embedded chips, drones, memories snapshots, to game consoles, that is to say, any device that produce digital data.

Computer forensics therefore requires a correct application of scientific methods, technique and tools to complete the stages related to the identification, preservation and analysis of digital evidence which, if necessary, can be considered legally in a judicial process, so in addition needs to ensure the quality and traceability of these data.

Given this panorama, this PID 7052 [5] seeks to advance in the comparative study of data collection methodologies related to security incidents and, particularly, to analyze the performance of these methodologies in web server environments.

2 Data collection methodologies

Currently, and generally speaking, data collection methodologies can be classified into two approaches:

2.1 Preventive Approach: Data Collection a priori from a security event. Also known as *Forensic Readiness* [6], [7], [8]. This approach introduces the concept of guarding the possible evidence before an incident occurs to primarily cover two objectives: maximize the environment's ability to gather reliable digital evidence and minimizing the forensic cost during the response to an incident. The premise is that such data can be used not only as an input for the analysis of potential security incidents and recovery for business continuity, but also as legal evidence which involves the assurance of the evidence as the data is actively collected. On the other hand, it is fundamental to have the ability to process data effectively and to have properly trained staff who know how to ensure that the potential digital evidence is rightly preserved. This approach further states that being prepared to gather and use evidence can also have benefits acting as a deterrent against the high rates of violation of internal security policies. Some of the key activities in Forensic Readiness planning are

- define the scenarios or assets than may require digital evidences;
- identify the available sources and different types of possible evidence;
- establish a safe way of collecting evidence to meet the legal admissibility requirements;
- establish a policy for safe storage and safe handling of evidence
- ensure monitoring to detect and prevent major incidents;
- train staff so that everyone understand their role in the digital evidence process and the legal sensitivity evidence;
- ensure legal control to facilitate action for incident response

In summary, an organization's ability to exploit these data and anticipate the response to an incident is the focus of Forensic Availability.

2.2 Reactive Approach - Post-Collection of a security event. This approach attempts to recover the evidence after the security incident is detected in order to perform a forensic analysis to determine what happened. The examination must be conducted in such a way as to ensure the admissibility of evidence. Piccirilli [9] provides in his doctoral thesis a description of the stages that can be applied in cases recollection of digital evidence involving computer-related elements, which include:

- the study and analysis of the environment, to identify the digital evidence to be obtained;
- the analysis claims subject to expert examination, which establishes the objective that the digital evidence must meet
- the acquisition of digital evidence;
- the analysis of the evidence obtained, in accordance with the guidelines of the request forensic examination;
- the way of presenting the digital evidence obtained from the performed investigation;
- the preservation of the treated digital evidence (for eventual future stages of investigation, whose source would be the digital evidence itself).

Likewise, an exploratory analysis has been carried out on the most widely considered protocol standards and models, such as RFC 3227 [10] and ISO/IEC 27037 [11], where a set of common points are observed for correct forensic analysis. Among them we can summarize the importance of preserving the testing environment, how and where evidence is stored, how it is analyzed for maximum outcome, and finally the importance of reports that are clear and concise [12], [13].

In this paper, both methodological approaches applied to web servers will be analyzed, specifically analyzing HTTP protocol information in versions 1.1 and 2.

3 HTTP Protocol

Hypertext Transfer Protocol (HTTP) is an application level protocol with defined characteristics for use in distributed, collaborative and hypermedia information systems. It is characterized by being a simple and widely accepted client/server protocol, which defines the structure of the request/response messages as well as the way in which these messages are exchanged between clients and web servers. Improvements have been introduced in its different versions, mainly aimed at improving its performance, reducing resource consumption and latency, and resolving some of the problems of communication through TCP [14],[15].

The latest version HTTP/2 [16], presents a binary protocol that incorporates multiplexing and the mandatory use of TLS keeping the same semantics and compatibility with versions 1.0 and 1.1.1. The protocol is implemented if the client and server have support and if either of them do not have it, in the protocol negotiation, it is agreed to use the previous versions. Currently, most browsers and server environments have official implementations for the new version.

The following table summarizes the main differences between the versions:

Table 1. Main differences between HTTP versions.

	HTTP/1.0 (1996) RFC 1945	**HTTP/1.1 (2000) RFC 2616**	**HTTP/2 (2015) RFC 7540**
Requirements management	A requirement delivered at a time, over a connection.	HTTP Keep Alive Mechanism: several requirements can use multiple connections with the server to reduce latency.	Multiple request / response messages on the same connection. Allows you to assign priorities to the requirements.
Header Field	Text format. HTTP messages: ASCII encoding sent as plain text over the connection.	Text format. HTTP messages: ASCII encoding sent as plain text over the connection.	Binary format. HTTP messages are converted into encrypted binary frames Compression of header (HPACK Algorithm).
Multiplexing	Does not allow simultaneous connections using the same TCP connection.	Does not allow simultaneous connections using the same TCP connection.	Allows multiple requests and responses in parallel using the same (a single) TCP connection, sending each request in a different stream.
Server side	Download of resources at the client's request (first HTML, then CSS, JS, images, links)	Download of resources at the client's request (first HTML, then CSS, JS, images, links)	Server Push Technology: allows files (CSS, JS, images) to be uploaded from the server to the client without the client asking for it.

4 Referring to testing and control points

As it has been expressed in previous paragraphs, the analysis proposed in this work is based on the need to reach general and comparative conclusions about the performance of the two methodological approaches considering aspects such as: the quality of the collected data, the traceability of the data, the level of responsibility of the data, analysis of incidents response answer times, conservation of the evidence and volume of the collected data.

In order to develop the activities and tests, a work group has been configured in a LAN Ethernet web which is made of an Operative Ubuntu Server version 15.10 and a Web Apache Server version 2.4.12. This net is completed by six workstations connected to both, the wired and wireless net.

For the execution of tests and the acquisition of data, free distribution computing forensics tools have been analyzed [17], [18], such as CAINE [19], Black Arch Linux [20] and Kali Linux [21]. Kali Linux version 64 bit 2017.1, which has more than 300 tools and applications related with the audit and the computer forensics was chosen.

The following documents have been selected as general guides for the proofs and the frame of the work:

- RFC 3227: published by the Internet Engineering Task Force (IETF) which set guidelines to collect and store evidences without putting them on risk.
- ISO/IEC 27037:2012 which gives guidelines for the proper handling of the digital evidence governed by three essential principles: the relevance, the reliability and the sufficiency.
- OSSTMM (Open Source Security Testing Methodology Manual) [22]. It is one of the most complete professional standards used to audit systems security.

The project's scheduled first activities originated some results, which are detailed as follows. The first activity consisted on identifying the control points in the protocols HTTP 1.1 and HTTP/2. In order to do this, the capture, analysis and guard of the following have been selected.

a. Incoming and outgoing traffic of the ports 80 and 443 TCP, to get information about the possible kinds of attacks and the origin of them.
b. Condition of the established connections in the 80 and 443 ports.
c. Log files (/var/log/):
    - *messages.log*: general system of messages record.
    - *auth.log*: authentication record.
    - *secure*: authentication record.
    - *utmp/wtmp*: logins record.
d. *httpd*: log record of Apache: error.log and access.log. The former gives diagnostic information and records any that could occur in the processing requirements. The latter stores all the processed requirements by the server.
e. configuration files from Apache server: with the purpose of determining no authorized modifications in the server configuration altering its function.

The second activity focuses on the identification of comparative points among the gathering methodologies of digital evidence, for which data are being collected. These data will be analyzed in the following stages taking into account the particularities of each approaching.

a. Preventive approach
   - Monitoring and recompilation of data according to the established points detailed in the activity 1.
   - Two daily copies of the collected data are stored on external storage with integrity protection of hash (MD5).

b. Reactive approach
   - Standard monitoring and recompilation a data according to the points detailed in the activity 1.

These data will allow to build a comparative matrix with quantitative and qualitative data, which will try to answer about some main issues such as:

- ○ Which methodology does offer a better answer in case of a security incident?
- ○ Which will be the infrastructure cost of any of them?
- ○ Which approach does offer better times of operative recovery?
- ○ Which approach does offer the most suitable environment to realize computer forensics after an incident?
- ○ Can the quality, traceability and inviolability of the collected data be secured in the preventive approach?

5 Conclusions and future work

This article presents the first findings of PID 7052-UNER called Analysis of methodologies of digital data collection.

With regard to the collection methodologies proposed for the analysis, an exploratory study of related material and publications was carried out, identifying the main characteristics, objectives and reasons why an organization can adopt a preventive or reactive approach to security incidents. In this sense, the team intends to build a comparative matrix that presents the relevant aspects in terms of quality, traceability and data availability, the incident’s response time in both cases and last but not least, the volume and way of storing the gathered information.

The new version of the HTTP protocol called HTTP/2 has been analyzed in order to know its implications for data traffic capture. From the point of view of interest to this work, no major changes were detected at the application level, but progress should be made in the future in the analysis of its relationship with TCP and security restrictions with the use of TLS.

First results show that: a) The Forensic Readiness methodology provides an active mechanism for anticipating incidents in contrast to security incident response methodologies; b) Digital continuity, risk management and forensic preparedness support each other; c) Maximizing the exploitation of potential evidence: preserved, unpolluted or damaged evidence; d) The integrity of the data is ensured with a digital hash strip.

The next stages of the PID, which are under development, include the simulation of a Denial of Service (DoS) attack for the purpose of analyzing the various aspects already mentioned related to the performance of the methodologies.

References

1. ISACA Homepage, Incident Management and Response, http://www.isaca.org/, last accessed 2017/07/23.
2. Internet Crime Complaint Center (IC3) Homepage, Annual Report 2015, http://www.ic3.gov/media/annualreports.aspx, last accessed 2017/03/14.
3. Ministerio Público Fiscal de la Ciudad Autónoma de Buenos Aires Homepage. CyberCrime Informe Final 2013 - Delitos Informáticos, http://delitosinformaticos.fiscalias.gob.ar/wp-content/uploads/2014/02/CyberCrime-Informe-Final-2013-flip.pdf, last accessed 2017/02/14.
4. Digital Forensic Research Workshop (DFRWS) Homepage, http://www.dfrws.org/, last accessed 2017/02/14.
5. Tugnarelli, M., Fornaroli, M., Santana, S., Jacobo, E., Díaz, J.: Análisis de Metodologías de Recolección de Datos Digitales. In: Libro de Actas Workshop de Investigadores en Ciencias de la Computación 2017, pp. 1000-1004. ISBN 978-987-42-5143-5.
6. TAN, John: Forensic Readiness. http://isis.poly.edu/kulesh/forensics/forensic_readiness.pdf, last accessed 2016/09/30.
7. Rowlingson, Robert: A Ten Step for Forensic Readiness. International Journal of Digital Evidence Volume 2, 1-28 (2004).
8. Pooe, A., Labuschagne, L: A conceptual model for digital forensic readiness, http://ieeexplore.ieee.org/document/6320452/, last accessed 2017/07/10.
9. Piccirilli, Dario: Protocolos a aplicar en la forensia informática en el marco de las nuevas tecnologías (pericia – forensia y cibercrimen). Tesis de doctorado. Facultad de Informática. Universidad Nacional de La Plata. http://hdl.handle.net/10915/52212. (2016).
10. IETF Homepage, RFC 3227 Guidelines for Evidence Collection and Archiving. https://www.ietf.org/rfc/rfc3227.txt , last accessed 2016/08/30.
11. Guidelines for identification, collection, acquisition and preservation of digital evidence ISO/IEC 27037:2012.
12. . U.S. Department of Justice. Electronic Crime Scene Investigation: A Guide for First Responders, Second Edition. https://www.ncjrs.gov/pdffiles1/nij/219941.pdf, last accessed 2017/08/30.
13. Forte, D: Principles of digital evidence Collection. ElSevier, Network Security, Volume 2003, Issue 12, 6-7 (2003).

14. IETF Homepage, RFC 1945 Hypertext Transfer Protocol - HTTP/1.0 http://tools.ietf.org/html/rfc1945, last accessed 2017/06/30.
15. IETF Homepage, RFC 2616 Hypertext Transfer Protocol - HTTP/1.1 http://tools.ietf.org/html/rfc2616, last accessed 2017/06/30.
16. IETF Homepage, RFC 7540 Hypertext Transfer Protocol Version 2 (HTTP/2). https://tools.ietf.org/html/rfc7540, last accessed 2017/07/30.
17. Altheide, Cory, Carvey, Harlan: Digital Forensics with Open Source Tools. 10.1016/B978-1-59749-586-8.00001-7, p.p 1-8 (2011).
18. Tugnarelli, M.; Fornaroli, M.; Pacifico, C.: Análisis de prestaciones de herramientas de software libre para la recolección a priori de evidencia digital en servidores web. In: Libro de Actas Workshop de Investigadores en Ciencias de la Computación 2015, pp. 985-990. ISBN 978-987-633-134-0.
19. Computer Aided Investigative Environment Homepage, http://www.caine-live.net/, last accessed 2017/09/30.
20. BlackArch Linux Homepage, https://blackarch.org/, last accessed 2017/08/30.
21. KALI Linux Homepage, https://www.kali.org, last accessed 2017/08/30.
22. ISECOM Homepage, Open Source Security Testing Methodology Manual (OSSTMM), http://www.isecom.org/mirror/OSSTMM.3.pdf, last accessed 2017/03/30.