ISSN:0975-9646

A Modified Efficient Log File Compression Mechanism for Digital Forensic in Web Environment

Rashmi Gupta, Raj Kumar Gupta
Computer Science Department,
Technocrats Institute of Technology, Bhopal (M.P.)

Abstracts -With the move towards global and multi-national companies, information technology infrastructure requirements are increasing. As the size of these computer networks increases, it becomes more and more difficult to monitor, control, and secure them. Networks consist of a number of diverse devices, sensors, and gateways which are often spread over large geographical areas. Each of these devices produces log files which need to be analyzed and monitored to provide network security and satisfy regulations. current information systems are replete with log files, created in multiple places (e.g., network servers, database management systems, user monitoring applications, system services and utilities) for multiple purposes (e.g., maintenance, security issues, traffic analysis, legal requirements, software debugging, customer management, user interface usability studies). Log files in complex systems may quickly grow to huge sizes. Often, they must be kept for long periods of time. For reasons of convenience and storage economy, log files should be compressed. However, most of the available log file compression tools use general-purpose algorithms (e.g., Deflate) which do not take advantage of redundancy specific for log files. In this dissertation a specialized log file compression scheme is described in five variants, differing in complexity and attained compression ratios. The proposed scheme introduces a log file transform whose output is much better compressible with general-purpose algorithms than original data.

KEYWORDS: **Digital Forensic, Log File, Compression, Suspicious User**

I. INTRODUCTION

Digital forensic is one of the crucial part of information technology. Digital Forensic is same as world of forensic [1]. The specialist uses this forensic in tracking the crime to know that what was happened. The other question is that why the criminal did it even by whom it happened. In Digital forensic the criminal is a hacker. In such cases the evidence are the trail left by the hacker. These trail are recorded by the different log files. Now the specialist or analyst focus on these log files then get the evidence against the suspicious user. But there is a need of accurate and trustworthy log files in order to excellent results [2].

Relevant data can be collected in multiple places such as network servers, database management systems, user monitoring applications, system services and utilities.

II. DIGITAL FORENSIC

Digital forensics is an significantly derived and proven technique towards the protection, compilation, justification, recognition, investigation, explanation and presentation of cyber evidence consequent from cyber sources for the purpose of facilitate or furthering the reconstruction of events found to be criminal or helping to predict the unconstitutional actions shown to be troublesome to planned operations [2].One important Element of digital forensics is the reliability of the digital evidence.

In digital forensic, log files are like the black box on an airplane that traces the events happened within an organization’s system and networks. Logs are collection of log entries that play a very significant responsibility in facts congregation and each entry contains information related to a precise event that has happened within a system or a network. Log files helps cyber forensic process in probing and seizing computer, obtaining electronic evidence for criminal investigations and maintaining computer records for the federal rules of evidence [3].

III. LOG FILES

Assuming that a log source offers configuration options, it is generally prudent to be conservative when selecting initial logging settings. A single setting could cause an enormous number of log entries to be recorded, or far too much information to be logged for each event. Excessive logging can cause loss of log data, as well as operational problems such as system slowdowns or even denial of service conditions. System-level administrators need to consider the likely effect of the log source configuration not only on the logging host, but also on other log management infrastructure components—for example, excessive logging can cause significantly more usage of network bandwidth and centralized log storage [4].

3.1. Various Types of Log

Various kinds of log files are available in our computer system; depending on their characteristics several of them are used for different purposes like security, data retrieval, analyzing, Authentication & etc. Several of them are following [4].

- **Network Device Logs**

Contain information on network traffic which is meeting an application rule or a packet filter rule. Traffic is not logged unless the Log communication to network log option is enabled. There are several network device logs,

- Router Logs- Router logs contain Network Traffic, Inbound and outbound packets and bandwidth Utilization information.
- L3 Switch Logs- L3 Switch Logs contain Network Traffic and SNMP trap information.

- **Firewall Logs**

Firewall logs provide useful information about the inbound and outbound packets, Information about particular servers e.g. Web Server, Packets which have been dropped, Alerts to the SA, Probing the system.

- **IDS Logs**

IDS logs Provides the information about Alerts on suspicious packet types, Attack statistics (Host / Network based). It Help in determining the probes and generating new attack signatures.

- **Web Server Logs**

A file that records every request and important information about the request from web server made by users known as web server log. For example, every time a browser requests a page, an entry is automatically made in this log by the web server, containing information such as the address of the computer on which the browser was running, the time at which the access was made, and the transfer time of the page, etc. There are several Web server log.

IV. Compression

In computer science and information theory, data compression, source coding or bit-rate reduction is the process of encoding information using fewer bits than the original representation would use. Compression is useful because it helps reduce the consumption of expensive resources, such as hard disk space or transmission bandwidth. On the downside, compressed data must be decompressed to be used, and this extra processing may be detrimental to some applications. For instance, a compression scheme for video may require expensive hardware for the video to be decompressed fast enough to be viewed as it is being decompressed (the option of decompressing the video in full before watching it may be inconvenient, and requires storage space for the decompressed video). The design of data compression schemes therefore involves trade-offs among various factors, including the degree of compression, the amount of distortion introduced (if using a lossy compression scheme), and the computational resources required to compress and uncompress the data.

V. Problem Statement

Digital investigations are becoming more time consuming and complex as the volumes of data required to analysis is large in size therefore lossless compression log file in lossless manner. There is possibility of manipulation or deletion of log information. Because log files are incriminating evidence against attackers, these files are at risk of attacks. Therefore, a mechanism is needed to prevent the manipulation and deletion of log info and log files by attackers and maintain the contents of log files that are created at the time of outbreak.

The aim of this paper is to propose an effective and efficient scheme for digital forensics to resolve following problems that arise during log file analysis for forensic investigation

VI. Proposed Solution

Log files are plain text files in which every line corresponds to a single logged event description. The lines are separated by end-of-line marks. Each event description consists of at least several tokens, separated by spaces. A token may be a date, hour, IP address, or any string of characters being a part of event description. In typical log files the neighboring lines are very similar, not only in their structure, but also in their content.

The proposed transform takes advantage of this fact by replacing the tokens of a new line with references to the previous line. There may be a row of more than one token that appears in two successive lines, and the tokens in the neighboring lines can have common prefixes but different suffixes (the opposite is possible, but far less frequent in typical log files).

6.1. Architecture

The Three tier architecture with various log files gives the solution of this problem. This three tier architecture divides the different reasonability to different layer. The first tier is used to remove ambiguity from the log files. The second tier generates the token from the log file. Finally the third tier takes care of all the redundancy left by the last executed phase.

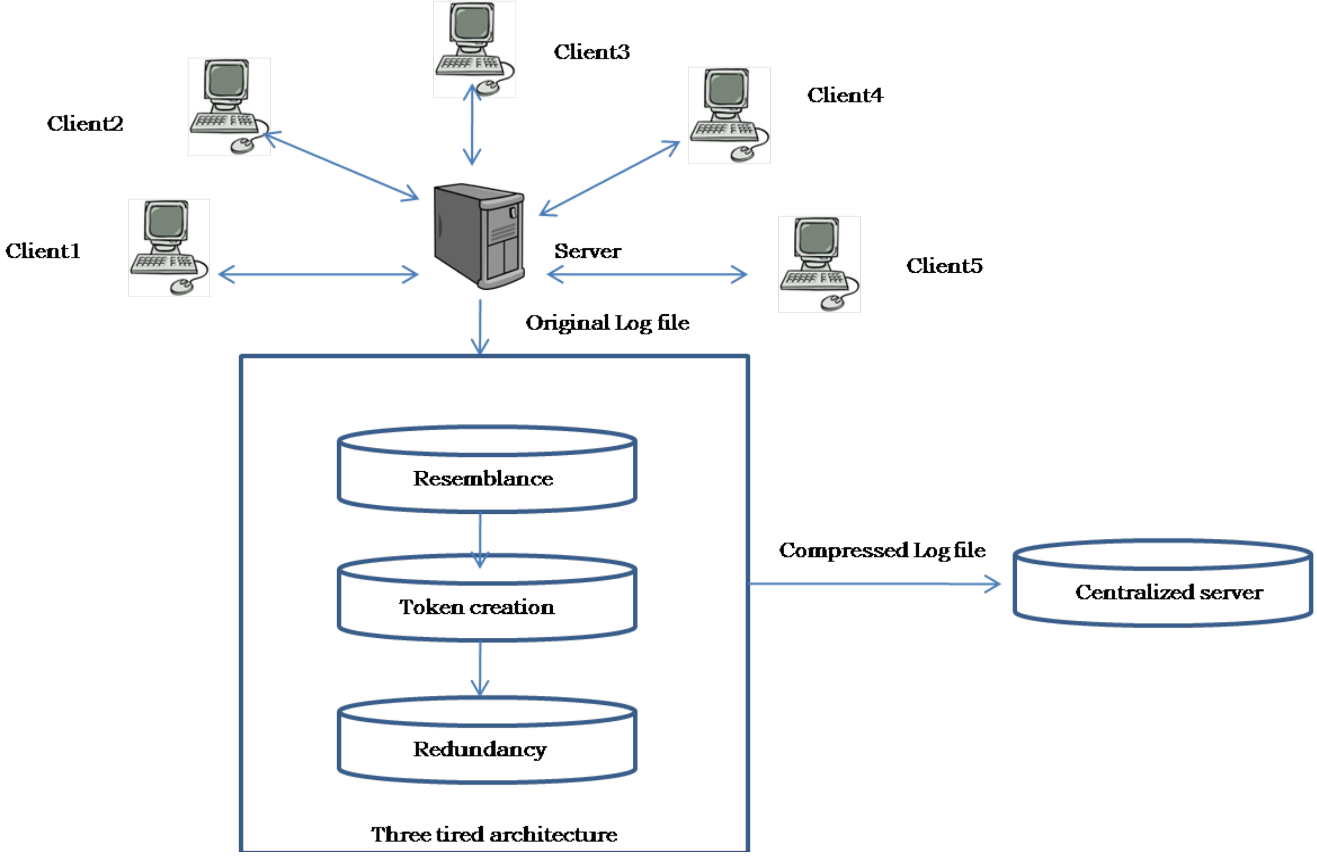
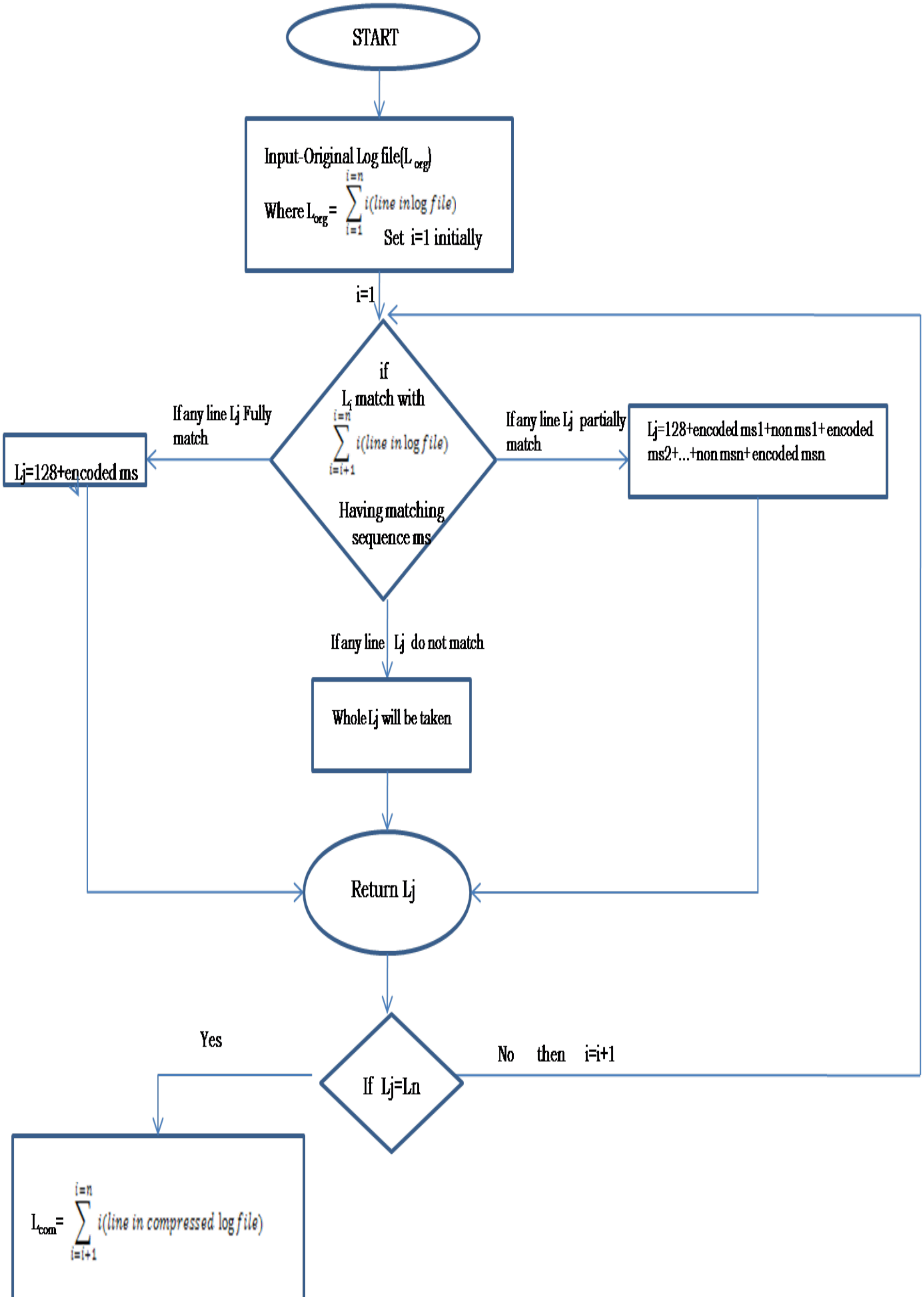


Figure 1: Architecture of Proposed Method

6.2. Algorithm

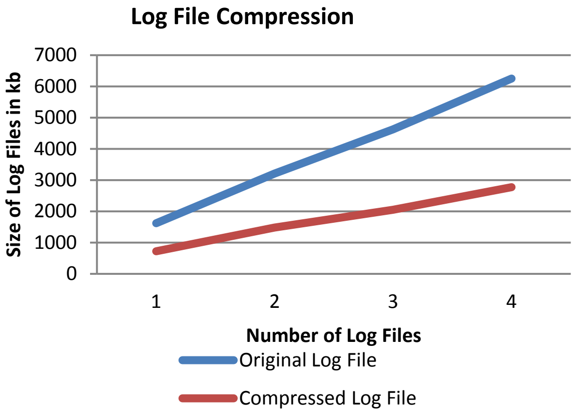

START
Input-Original Log file(L_{org})
Where $L_{org} = \sum_{i=1}^{i=n} i(line\ in\ log\ file)$
Set i=1 initially
i=1
if L_i match with $\sum_{i=i+1}^{i=n} i(line\ in\ log\ file)$ Having matching sequence ms
If any line Lj Fully match
Lj=128+encoded ms
If any line Lj partially match
Lj=128+encoded ms1+non ms1+ encoded ms2+...+non msn+ encoded msn
If any line Lj do not match
Whole Lj will be taken
Return Lj
If Lj=Ln
Yes
No then i=i+1
$L_{com} = \sum_{i=i+1}^{i=n} i(line\ in\ compressed\ log\ file)$

VII. Results

The result shows the two major indispensible properties of Log Files. The first one is the compressed file size and the other one is compaction ratio. Here the graphs show both of the properties.

On one hand first graph shows the difference between original log files and compressed log files On other hand second graph displays the compaction ratio.

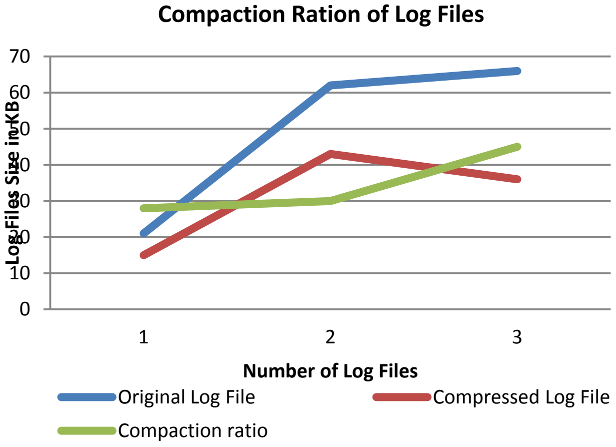
The third graph compares the existing method and the proposed method which shows that the proposed method is better than the existing method.



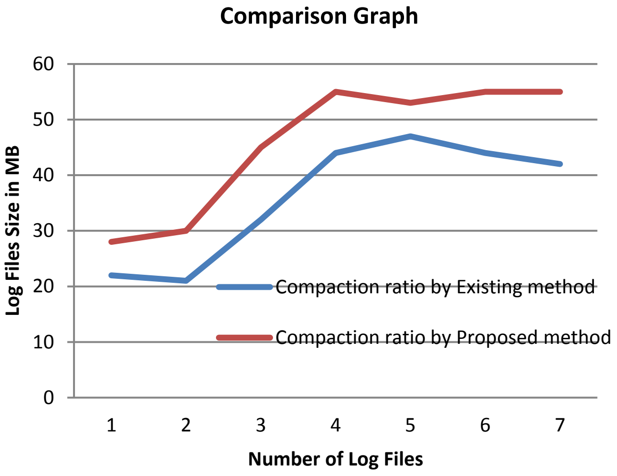
Graph 1

7.1. Comparison Table

Original Log File	Compressed Log File	Compaction ratio by Existing method	Compaction ratio by Proposed method
21	15	22	28
62	43	21	30
66	36	32	45
1622	722	44	55
3213	1482	47	53
4628	2052	44	55
6250	2775	42	55



Graph 2



Graph 3

VIII. Conclusion

In this paper a fully reversible log file transform is describe that capable of significantly reducing the amount of space required to store the compressed log. The transform has been presented in different variants aimed at a wide range of possible applications, starting from a fast variant for on-line compression of current logs (allowing incremental extension of the compressed file) to a highly effective variant for off-line compression of archival logs. The transform was not tuned for any particular type of logs, its set of features was designed for different types of logs, and the obtained test results show it manages to improve compression of different types of log files. It is lossless, fully automatic (it requires no human assistance before or during the compression process), and it does not impose any constraints on the log file size.

Acknowledgement

The research presented in this paper would not have been possible without our college, at TIT, Bhopal. We wish to express our gratitude to all the people who helped turn the World-Wide Web into the useful and popular distributed hypertext. We also wish to thank the anonymous reviewers for their valuable suggestions

References

[1] Kessler, M. G. (2006). Kessler's Corner: The growing field of computer forensics. The Kessler Report, 9(1), 7.

[2] Gary L Palmer "A Road Map for Digital Forensic Research". Technical ReportDTR-T0010-01, DFRWS. Report for the First Digital Forensic Research Workshop (DFRWS), 2001

[3] Rafael Accorsi, "Safekeeping Digital Evidence with Secure Logging Protocols:State of the Art and Challenges" University of Freiburg, Germany.

[4] M. Bishop, "A Standard Audit Trail Format", National Information Systems Security Conference, Baltimore, MD , 1995

[5] T. C. Bell, J. G. Clearly, and I. H. Witten. Text compression. Prentice Hall, Upper Saddle River, New Jersey, USA, 1990.

[6] T. C. Bell, J. G. Cleary, and I. H. Witten. "Modeling for text compression". ACM Computer Surveys (CSUR), 21(4):557–591, December 1989.

[7] News Limited. Phishers hit monster jobs site. Available Online: http://www.australianit.news.com.au/story/0,24897,2229308215306,00.html (LastAccessed 29 August 2007), August 2007.

[8] Pavel Gladyshev "Formalising Event Reconstruction in Digital Investigations" Ph.D. dissertation Department of Computer Science, University College Dublin, 2004.

[9] Bernie Lantz,Rob Hall, Jason Couraud, "Locking Down Log Files: Enhancing Network Security By Protecting Log Files" Issues in Information Systems Volume VII, No. 2, 2006

[10] ZhiyongWu , Bin ZhuGe Weiming Wang ,"Tamper Resistance Protection Of Logs Based On Forward-Secure" Institute of Network and Communication Engineering, Zhejiang Gongshang University , 2010 IEEE

[11] L. Volonino, "Electronic evidence and computer forensics,"Communications of the Association for Information Systems,vol. 12, Article 27, 2003.

[12] E. Casey, "Error, uncertainty and loss in digital evidence,"IJDE, vol. 1, no. 2, 2002.

[13] Benjamin Boeck, David Huemer, A Min Tjoa,Towards more Trustable Log Files for Digital Forensics by Means of "Trusted Computing" in 24th IEEE International Conference on Advanced Information Networking and Applications, 2010

[14] Nobutaka Kawaguchi, Shintaro Ueda, Naohiro Obata, Reina Miya ji, Shinichiro Kaneko, Hiroshi Shigeno, Kenichi Okada," A Secure Logging Scheme for Forensic Computing" Proceedings of the 2004 IEEE Workshop on Infonnation Assurance United States Military Academy , West Point, NY 10-11 June

[15] Muhammad Kamran Ahmed, Mukhtar Hussain and Asad Raza "An Automated User Transparent Approach to log Web URLs for Forensic Analysis" Fifth International Conference on IT Security Incident Management and IT Forensics 2009.