Log File Compression and its Security in Web Server

Sweta Singh
United Institute of Technology
Naini, Allahabad

Prashant Shukla
United Institute of Technology
Naini, Allahabad

ABSTRACT

The log file of any association may include sensitive data which must be protected properly for suitable working of that organization. Maintaining security of such log records is one of the important tasks. Also, over a long period of time maintaining authenticity of such log data is very important. However, deploying such a system for security of log records is a big task for any company and also it needs additional cost. There are many techniques have been proposed so far to secure log records. This paper presents a brief survey of optimal approaches for securing log files for forensic analysis. These techniques are reviewed considering its pros and cons.

General Terms

Security, Compression

Keywords

Log files, Forensic, Privacy, Confidentiality.

1. INTRODUCTION

All internet surfing activities performed by each user in an organization are record in the form of log files. Log files be track of all user behavior with various attributes like date, time web site etc. Also, log files are used to resolve different problems, to identify those users who violating the policies or performing malicious activities. It is also useful to identify intruder. Log file is the most desirable target for attacks [9]. The logic of purpose in background is that intruder with malafide intention would prefer to leave no traces outs of the activities executed at the time of intrusion. Hence, the target of attack is normally log files. Apart from this log files also contained information about private transactions performed in any organization. This susceptible data must be protected [6]. Also log file data can be used for unlawful access to the system [9]. From these scenarios, it is clear that security of log files is one of the most important tasks of an organization. The large no. of information is available on Web and database is still rapidly increasing.

For every website, normally thousands of users will be access a site. The administrator of a system has an access to the server log.

When web users interact with a site, data recording their performance is stored in server logs. Log files may contain very useful information characterizing the users’ experience in the website.

Since in a normal size site log files amount to several megabytes a day, there is a requirement of technique and tools to help take benefit of their matter. Usually log files are in the form of text files that can range from 1kb to 100mb, which depends on the congestion at a particular website.

In determining the amount of traffic a site receives during a particular period of time, to understand what exactly the log files are counting and tracking.

Computer forensics is one of the most concerns in IT. Computer forensics is similar to the forensic, Where Police acts like the forensics to clean a crime scene for evidence of what happened, to whom it happened, and who did the crime.

In the case of computer, the crime scene is the machine that was hacked, the injured party is the entity to which the computer belongs, and the hacker is the unlawful. The proof in the case of computer forensics is the track left by the hacker; all are recorded in the log files.

For computer forensics to be effective, it must be precise and truthful log files.

Log files are available for securing a network against intrusion. Log files also record network traffic, a note of the IP addresses is access to your network, on which IP port access was attempted, also date and time , whether or not the attempt was successful, etc.

If used to correct log files can be very helpful to maintain network security and integrity. However, for log files provide to determine security login must be activated and the files must be checked time to time. Log files will provide protection against beginner hackers.

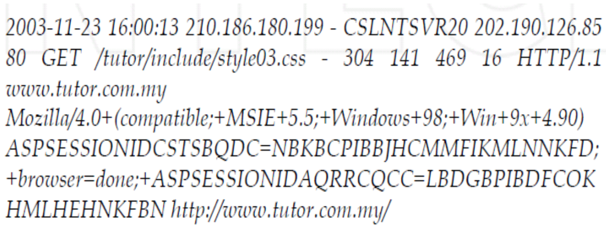
More experienced hackers are aware of log files and to hide their actions from the administrator by either deleting the log file altogether or replace the log file with another duplicate log file showing normal network activity.

In the primary instance, the network administrator will know that an intrusion was detected, but will have no clue as to the determine of the attacker or how the attacker entered the system.

In the second instance, the network manager will have no hints, and will have to switch on other techniques for detecting intrusion.

1.1 Structure of Log files

The log files consists of 19 attributes such as Date, Time, Client IP, AuthUser, ServerName, ServerIP, PortServer, MethodRequest, URI-Stem,query , Protocols Status, Time Taken, Bytes Sent, Bytes Received, Protocol Version, Host, User Agent, Cookies. One of the main problems encounter when dealing with the log files is the amount of data needs to be preprocessed.



```
2003-11-23 16:00:13 210.186.180.199 - CSLNTSVR20 202.190.126.85
80 GET /tutor/include/style03.css - 304 141 469 16 HTTP/1.1
www.tutor.com.my
Mozilla/4.0+(compatible;+MSIE+5.5;+Windows+98;+Win+9x+4.90)
ASPSESSIONIDCSTSBQDC=NBKBCPIBBJHCMMFIKMLNNKFD;
+browser=done;+ASPSESSIONIDAQRRCQCC=LBDGBPIBDFCOK
HMLHEHNKFBN http://www.tutor.com.my/
```

Figure 1. Example of log file

Before we will discuss various aspects of this topic. We start with describing the desirable properties that are needed for securing log files. Various properties are analyzed in this section.

1.2 Various Properties

The various properties to safe log files are provided below:

1) Correctness**:** The log data stored in log files must be correct. It means that the stored data must be similar with new generated data.

2) Tamper Resistance: Only log file administrator can edit the valid log entries[3]. And if any manipulation is done, it should be noticed.

3) Verification: Every log entry must have sufficient information to check its reliability and authenticity to make sure that the log entries are unchanged.

4) Confidentiality: As an attacker can take vulnerable information from log files, therefore log records should be stored in such a way so that they should not be traceable to anyone in the network.

5) Privacy: Log records should not be detected by the attacker during its transfer and its storage [5].

1.3 Threat Model

In this section we discuss the threats available in present situation while to secure log files [4][7]. Different types of attacks that we need to defend against are given below.

1) Integrity of Log Records during transfer: The attacker can obtain illegal entry to communication medium. And he can not only have access to the data but also he can change the data which is move to the log server for safe storage. \

2) Authenticity of Log Record creator: The attacker may impersonate to be valid network consumer and begin to transmit log records from someone else's identity [8].

3) Confidentiality of Log files: The attacker may try to associate the log records over traffic to find the information about perceptive transactions of an organization.

2. PROPOSED APPROACH

Proposed approach consists of four steps as shown in figure 1.

1) Log file compression
2) Authentication bit generation and log file to image conversion
3) Tamper detection
4) Log file decompression

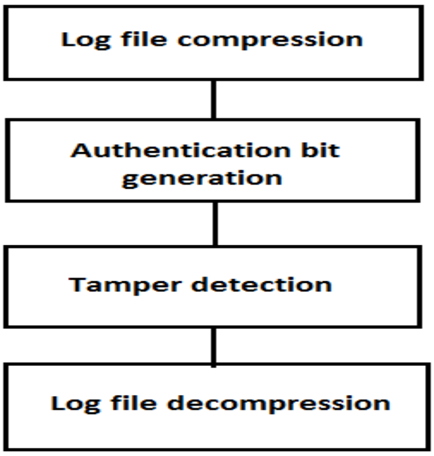


Figure 1: Flow of the proposed approach

As we know that log file is nothing but the combination of various alphanumeric characters which includes number, characters and special symbols. All given four steps are hierarchal. It means output of previous step will be the input of next steps.

2.1 Log file compression

Since log files are very huge in size hence before proceeding we need to compress it. There are following steps are provided to compress the log file.
Step1- Take a log file as an input and read it's all characters row wise.
Step2- Convert all characters into it's ASCII form, which will be decimal number .
Step3- All decimal numbers are converted into eight bit binary form.
Step4- Append the frequency count with each bit. For example if there is a bit stream 1111 it will be written like 14. But if there will be only single 0 or 1, no frequency count will be written beside it.

Step5- Output of step 4 will be represented as compressed log file which will be no more readable.

2.2 Authentication bit generation and log file to image conversion

Once we get the compressed log file it will be embedded with authentication bit in order to make self authenticating log files. These authentication bits are nothing but fragile watermark, which will be destroyed if any alteration is done to log file, hence tamper can be easily localized. Block diagram is shown in figure 2. The detail steps are shown below:

Step1- Take the compressed log file as an input.

Step2- Convert the compressed log file into binary vector.

Step3- Create clusters of five bits for all bits of log file binary vector.

Step4- Generate three authentication bits for each cluster of five bits by following way.

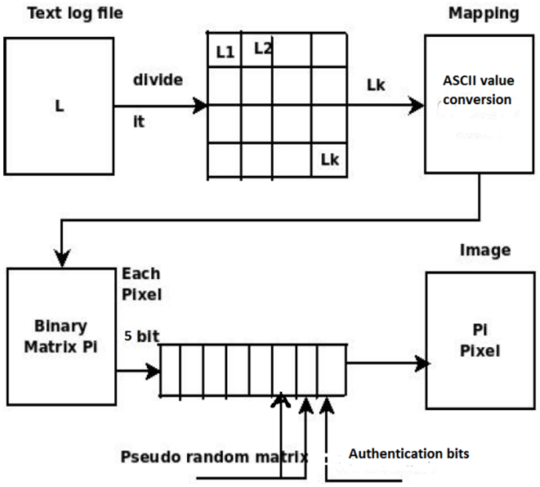


Figure 2: Authentication bit generation for image log file

For 1st bit generation-

Step 5- Take bit wise XOR of five bits of each cluster and calculate the modulo 2 of the sum of them.

$$1\text{st bit} = \left(\sum_{i=1}^{4} b_i \oplus b_{i+1}\right) \bmod 2 \qquad (1)$$

Where b_i represents the i^{th} bit of vector.

For 2nd bit generation-

Step 6- Using a secret key, generate a random matrix Rm of same size r x c as image whose values ranges from 0 to 31. Do bit wise XOR between corresponding cluster's bits and bits of Rm.

$$2\text{nd bit} = \left(\sum_{i=1}^{5} b_i(\mathrm{Im}) \oplus b_i(\mathrm{Rm})\right) \bmod 2 \qquad (2)$$

For 3rd bit generation-

Step7- Calculate the decimal value of five bit cluster.

Step 8- Calculate the complement of the each decimal value of cluster and take the bit wise XOR with its original value.

$$3rd\ bit = \left(\sum_{i=1}^{5} b_i(Im) \oplus b_i(31 - Im)\right) mod\ 2 \qquad (3)$$

Step 9- Append all generated three bits to five bits of it's corresponding cluster as shown in figure 3.

Step10- Now take the decimal value of each eight bit binary cluster.

Step11- The resultant matrix will be represented as image.

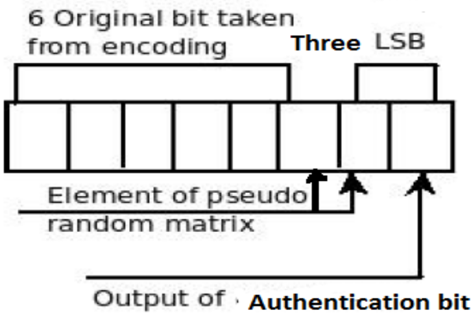


Figure 3: Bit arrangement for a pixel

2.3 Tamper detection

It may be possible that log file is tampered intentionally or unintentionally. Sometimes attacker tries to alter the log files to remove their suspicious activity. But proposed scheme as created a self authenticating log file which itself is sufficient enough to detect the alteration as shown in figure 4. The detailed procedure is as follows:

Step1- Take the altered log file image as an input.

Step2- Take a tamper localization matrix with all initially assigned values as 0.

Step3- Extract the last three bits of all pixels of log file image.

Step4- Recalculate the three bits for all cluster using eq. 1, 2 and 3.

Step5- Compare the extracted and recalculated three bits for all corresponding pixels.

Step6- If any mismatch is found then mark the corresponding location in tamper localization matrix.

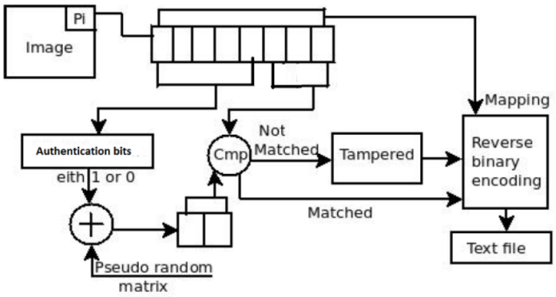


Figure 4: Tamper localization procedure

2.4 Log file decompression

Once image log file is checked for tamper, we need to decompress it in readable format. Following steps explain the decompression approach.

Step1- Extract five MSBs of each pixels of image log file and make a vector.

Step2- Now creates clusters of eight bits from given vector.

Step3- All eight bit clusters are converted into decimal format.

Step4- All 1 and 0 bits are expand with its adjacent frequency count for example 14 is written as 1111.

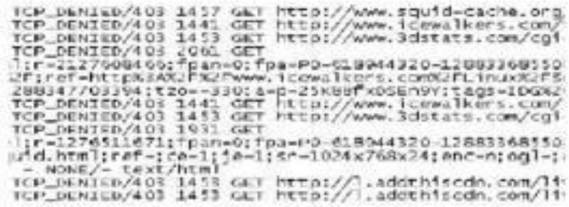
Step5- Now again creates clusters of eight bits from given vector.

Step6- All eight bit clusters are converted into decimal format.

Step7- Now replace all decimal value with its ASCII values and the resultant text in copied in output file which is nothing but the extracted decompressed log file.

3. EXPERIMENTAL RESULT

A proposed algorithm for log file security is simulated in Matlab 2010 and has been taken as the text log files of variable sizes from available web server logs. Depending upon scheme, text log files are taken first as algorithm input as shown in figure 5.



```
TCP_DENIED/403 1457 GET http://www.squid-cache.org
TCP_DENIED/403 1441 GET http://www.icewalkers.com/
TCP_DENIED/403 1453 GET http://www.3dstats.com/cgi
TCP_DENIED/403 2061 GET
l;r=2127900466;fpan=0;fpa=P0-618944320-12083368;50
2F;ref=http%3A%2F%2Fwww.icewalkers.com%2FLinux%2F5
288347/03394;tzo=-330;a=p-25KB8fxDSEn9Y;tags=1DG%2
TCP_DENIED/403 1441 GET http://www.icewalkers.com/
TCP_DENIED/403 1453 GET http://www.3dstats.com/cgi
TCP_DENIED/403 1931 GET
l;r=1276511671;fpan=0;fpa=P0-618944320-12883368550
uid.html;ref=;ce=1;je=1;sr=1024x768x24;enc=n;dgl=;
 - NONE/- text/html
TCP_DENIED/403 1453 GET http://l.addthiscdn.com/l1
TCP_DENIED/403 1453 GET http://l.addthiscdn.com/l1
```

Figure 5: Example of text log file

First of all text log file is compressed by proposed compression technique. After compression it will be no more readable in nature as shown in figure 6.

We can see that image log file consist few gray colors. Where the valuable information is occupied by non black color of text log file, whereas the absence of information is shown as black portion or remaining space in image log file as shown in figure 6. Text log file size and black colour present in image log file are inversely proportional to each other. It means when we increases the size of text log file then the area in image log file which contains black portion will decrease.

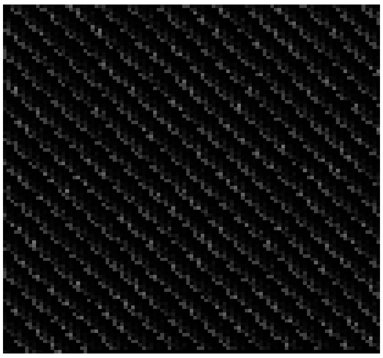


Figure 6: Compressed image log file

Now this compressed log file again embedded with some authentication bits in order to make itself authenticable log files as shown in figure 7.

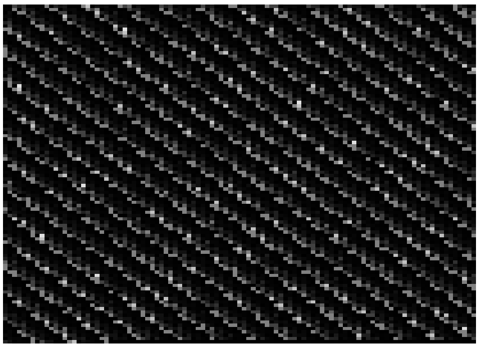


Figure 7: Encrypted image log file

Suppose an attacker has done some alteration in image log files like shown in figure 8. If we are not having the real image log file shown in figure 7, then one cannot say that given image log file shown in figure 8 is altered and where alteration is done. Since the original one is lost, there is no mean for comparison left. But now we pass this changed image log file into our image to text log file conversion algorithm then we get the result as shown in figure 9.

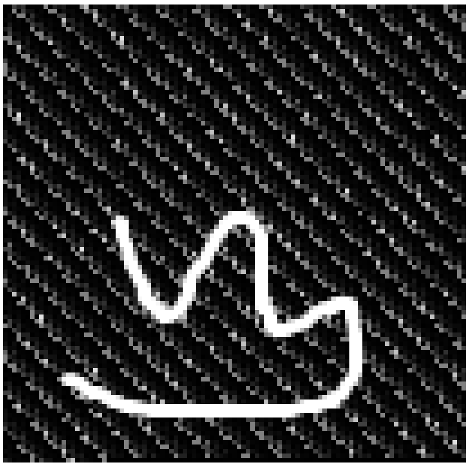


Figure 8: Altered log file.

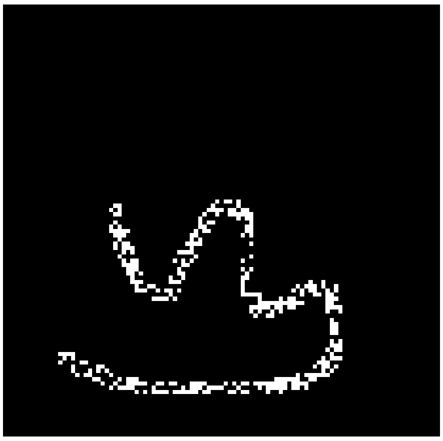
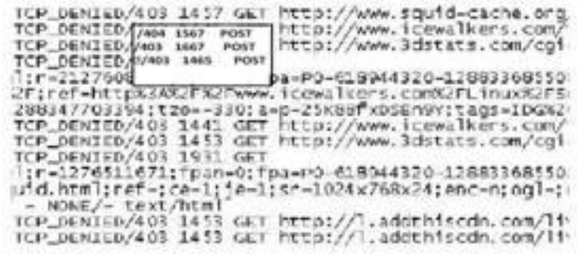


Figure 9: Alteration detected log file.

Here the white colored area shows the tampered portion in image log file which was very difficult to spot in text log file because it can easily be modified like one can change the IP address, date, time of crime and we cannot locate the changed area. After ensuring that image log file is altered somewhere, that log file as a digital evidence. Now when conversion is done between image log file and text log file then it will not be similar like original one as shown in figure 10. Now we can conclude that in only presence of the figure 10, no one can spot the altered portion in log file which are shown by box but if it will be converted into image log by proposed algorithm then it is easily identifiable that integrity of log file is compromised.



```
TCP_DENIED/403 1457 GET http://www.squid-cache.org
TCP_DENIED/404 1567 POST http://www.icewalkers.com/
TCP_DENIED/403 1667 POST http://www.3dstats.com/cgi
TCP_DENIED/403 1465 POST
1;r=2127608              ba=P0-61894320-12883368550
2F;ref=http%3A%2F%2Fwww.icewalkers.com%2FLinux%2F5
2883477033394;t20=-330;a=p-25K88fxDSEn9Y;tags=IDG%2
TCP_DENIED/403 1441 GET http://www.icewalkers.com/
TCP_DENIED/403 1453 GET http://www.3dstats.com/cgi
TCP_DENIED/403 1931 GET
1;r=1276511671;fpan=0;fpa=P0-618044320-12883368550
uid.html;ref=;ca=1;je=1;sr=1024x768x24;enc=n;ogl=;
 - NONE/- text/html
TCP_DENIED/403 1453 GET http://1.addthiscdn.com/1
TCP_DENIED/403 1453 GET http://1.addthiscdn.com/1
```

Figure 10: Altered text log file

4. CONCLUSION

This thesis proposes a new log file security technique as well as discusses numerous existing states of art approaches for log files security. For each technique a detailed contributions along with its advantages and limitations are provided. This exhaustive analysis highlights a number of future scopes of research in this topic.

Efficient preservation technique for log file ensures all security conditions such as integrity, confidentiality, authenticity. Another advantage of efficient preservation technique is self alteration detectable capability and also recovers the altered data. According to proposed approach we first compress the log file by proposed compression technique after that convert the compressed log file into image log file by means of bit encoding technique and tamper detection capability is achieved by self embedding fragile watermark scheme. In case of any alteration on image log file then due to nature of fragile watermark, one can easily locate that tampered region. Image log file can be used as digital evidence in network forensics.

5. REFERENCES

[1] M. Bellare and B. S. Yee, ―Forward integrity for secure audit logs,‖ Dept.Comput. Sci., Univ. California, San Diego, Tech. Rep., Nov. 1997.

[2] C. Lonvick, The BSD Syslog Protocol, Request for Comment RFC 3164,Internet Engineering Task Force, Network Working Group, Aug. 2001.

[3] D. New and M. Rose, Reliable Delivery for Syslog, Request for Comment RFC 3195, Internet Engineering Task Force, Network Working Group, Nov. 2001.

[4] U. Flegel, ―Pseudonymizing unix log file,‖ in Proc. Int. Conf. Infrastruture Security, LNCS 2437. Oct. 2002, pp. 162–179.

[5] J. E. Holt, ―Logcrypt: Forward security and public verification for secure audit logs,‖ in Proc. 4th Australasian Inform. Security Workshop, 2006, pp. 203–211.

[6] D. Ma and G. Tsudik, ―A new approach to secure logging,‖ ACM Trans. Storage, vol. 5, no. 1, pp. 2:1–2:21, Mar. 2009.

[7] J. Kelsey, J. Callas, and A. Clemm, Signed Syslog Messages, Request for Comment RFC 5848, Internet Engineering Task Force, Network Working Group, May 2010.

[8] BalaBit IT Security (2011, Sep.). Syslog-ng—Multiplatform Syslog Server and Logging Daemon [Online]. Available: http://www.balabit.com/network-security/syslog-ng.

[9] Indrajit Ray,K.Belyaev,‖Secure Logging As A Service-Delegating log management to the cloud ‖, IEEE Systems Journal,June 2013.