ISSN:0975-9646

A Novel Approach to Detect Web Page Tampering

Ramniwas Kachhawa[#], Nikhil Kumar Singh[*], Deepak Singh Tomar[#]

[#] *Department of Computer Science and Engineering,*
Maulana Azad National Institute of Technology,
Bhopal (462051), Madhya Pradesh, India

[*]*Maulana Azad National Institute of Technology,*
Bhopal (462051), Madhya Pradesh, India

Abstract—**Recently it was observed that some contents of the web pages are tampered by malicious users for their own benefits such as injecting advertisement or vulgar materials. Malicious user also tries to alter the contents in order to misguide the end user. Maintaining the integrity of these web pages has become a tedious job for the law enforcement agencies. This paper gives bird eye over the web page tampering and its side effects, importance of log file in the detection of web page tampering. It also presents a framework to detect tampered web pages based on log file analysis approach effectively. To achieve this, first log files are converted into XML format and then tamper web pages are tagged.**

***Keywords*— cyber-crimes, web page, web page tampering, log file, XML.**

I. INTRODUCTION

With the development of the Internet, knowledge gathering through web pages has become popular. But malicious users may tamper contents of web sites which breaches the integrity of the web pages and break downs the security of sensitive data. Web page may be tampered through different attacks like spear phishing [1], SQL Injection [2, 3] etc. These attacks breach the privileges of the web administrator to perform web page tampering.

Web page tampering may be effectively analysed through log files of web server. Log file contains the entry related to each request and response of the web server. These entries contain the information such as IP address, URL, Time stamp, Byte transferred etc. Information of log file may be represented in different formats i.e. XML and plain text etc.

In this work, web page tampering is detected through the analysis of web server logs. Log files are extracted from the web server and pre-processed to retrieve the required information. This information is represented in XML format because it is a structured format and stores the information as objects. Searching is efficient in XML format in comparison with plain text format. After pre-processing, log files are analysed to find out the patterns that may be used for the detection of web page tampering.

Rest of the paper is organised as follows. Section II presents background work and Section III describes an overview of web page tampering. Log file and its format are explained in Section IV, followed by XML representation of log files in Section V. Section VI explains the proposed framework and Section VII describes conclusion and future work.

II. RELATED WORK

Various researchers have proposed different methodologies to detect the web page tampering. This section focuses on these approaches.

Xianzhong Long et al. [4] developed a tool based on Novel Fragile Water Marking scheme to protect web contents tampering and authenticate the users. This tool is based on two approaches. First is MAFW (Message Authentication Based Fragile Watermark) which provides security and detects alterations of web pages. Second approach is SCFW (Sparse Coding Based Fragile Watermark) to authenticate the end user.

Tushar kanti et al. [5] proposed an algorithm for defacement detection of web with spotting the exact location of defacement and also implemented a Web browser with inbuilt defacement detection techniques.

Eric Medvet et al. [6] proposed an approach based on Genetic Programming (GP); GP automatically generates algorithms those are capable of detecting almost all unauthorized modifications.

Giorgio Davanzo et al. [7] proposed a technology for defacement detection that build automatically a profile of the monitored web page and generate an alert to the relevant monitored organization whenever something “unusual” happens.

Xiang yang Liu et al. [8] proposed two novel fragile watermarking schemes for tamper proof web pages. One is Keyed-Hashing for Message Authentication (HMAC) and other is Nonnegative Sparse Coding (NNSC). These are used to generate a fragile watermark from a web page and then embedded it into the web page. When tampering is occurs, the watermark in the tampered web page will be destroyed or become inconsistent with the content.

III. WEB PAGE TAMPERING

Web page tampering is a common type of cyber-crime in which the web contents are altered through the malicious activities [7]. Web page tampering may misguide the client, breach the web security, change the visual appearance and enhance the vulnerability of web page [9].

Web page tampering may be performed either on static web pages or dynamic web pages [10]. Detection of tampering in static web pages is easier rather than the dynamic web pages because the static web pages does not modified frequently. On the other hand, in dynamic web pages the degree of dynamism changes widely across pages. If there is change, it is not sure whether the change has been done by the authorised user or by an attacker.

Web page tampering may leads the victim organisation to following scathes i.e. reputation, customer trust, financially loss of business and integrity of the web page etc. So detection of the tampered web pages becomes necessary for victim organisations. In the detection of the tampered web page, log file plays an important role. Log file may be used to capture the footprint of web page whenever it is accessed by end user from web server.

IV. LOG FILE

Web Server Log files are the history book of digital information of web server [11]. It is simple plain text file which captures the entries related to the user's request and reply from the web server.

Each web server has its own log format such as JIGSAW server uses W3C log format, WEBSTAR server uses web Star Log Format and many more server have their own log format [12].

'Common Log Format' is famous log format for Apache HTTP Server Version 2.2.22. This format provides extensive and flexible logging capabilities at server side.

Common Log Format of apache 2.2.22 server is depicted in Fig. 1.

```
"%h %l %u %t %m %U%q %H %>s %b"
```

Fig. 1: Common Log Format

To understand the Common Log Format, a sample of log is captured from http://www.mytestingserver.com which is running on apache 2.2.22 server is depicted in Fig. 2.

```
192.168.5.46 - - [27/May/2014:14:33:11] "GET
/index.php?img=gifLogo HTTP/1.1" 200 4549
```

Fig. 2: Sample Log

Detail description of attributes of Common Log Format with the attributes value in sample log is represented in Table I.

Table I: Detail of the log format

Log Attributes	Sample Value	Interpretation
%h	192.168.5.46	IP Address of visitor
%l	-	RFC 1413 identity of the client
%u	-	user id of the person requesting the document
%t	27/May/2014:14:33:11	time that the request was received
%m	GET	Method used
%U%q	/index.php?img=gifLogo	client requested the resource
%H	HTTP/1.1	HTTP version
%>s	200	status code that the server sends back to the client
%b	4549	Bytes transferred

Each log entry contains attributes value i.e. URI, page size, IP address etc [13]. If attribute described in Table I, does not contains any value then the respective field assigned with hyphen. These attributes value may become the source of evidence for detection of tampered web page.

Bytes Transfer Field *'%b'* is a log attribute which represents the size of web page requested by the user. The size of static web page will be constant till the web administrator does not change it. For dynamic web page this value will be vary according to the user's requests. Value of this attribute *'%b'* may be taken into consideration when forensically detecting the tampering of static web page.

These values of log are stored into log file in plain text format and for searching the attribute value in the log file. As plain text format supports linear searching technique. The time complexity of linear searching technique is $O(n)$ [14]. To minimize the time of searching it is essential to convert the log of plain text format into another format. XML representation of log files may be used to minimise the time of searching. The complexity of searching in the XML data using keyword search is $O(kd|S_1|log|S|)$ where k is number of keyword in the query, d is depth of tree, $|S_1|$ and $|S|$ is the occurrence of least and most frequent keywords respectively in the query [15]. This paper converts the plain text format of log file into XML format. XML representation of log file is very much efficient as compare to traditional representation of log file.

V. XML REPRESENTATION OF LOG FILE

XML representation of log file is very important milestone where the issues of searching in the file, transport, storage of file and standard way of parsing comes arise. XML stores data in the form of a tree and supports keyword based searching i.e. nodes by node searching [16, 17]. First compare parent node and then child node. If parent node does not match then skips its entire child node and moves to their sibling node.

XML representation of log file shown in Fig. 2 is depicted in Fig. 3.

```
<?xml version="1.0"?>
  <root>
    <log IP="192.168.5.46">
          <identity> - </identity>
          <user_id> - </ user_id >
          <time>27/May/2014:14:33:11</time>
          <method>GET</method>
          <URI>/my_folder/modules/mod_superfishmenu/tm
          pl/js/jquery.event.hover.html</URI>
          <http_version>HTTP/1.1</http_version>
          <response_code>success 200</response_code>
          <bytes>3595</bytes>
    </log>
  </root>
```

Fig. 3: XML representation of sample log

. Data of the log comes with tags so searching in the XML represented log file is carried out according to tags. Indexing of keywords is based on these tags shown in Fig 3

VI. PROPOSED FRAMEWORK

In this paper, a framework has been proposed to analysis whether the requested web page has tampered or not. Usually log file contains several fields. But in this work, byte transferred field for requested web page is important evidence.

Proposed framework is based on the concept of detecting the web page tampering through matching bytes transferred attribute with the value of last log entry of requested web page.

This paper proposed a three tier framework consist of database section, registry section and decision section as depicted in Fig. 4.

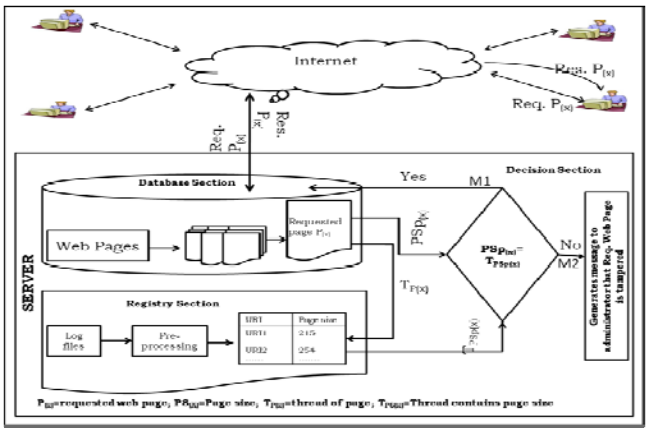

Fig. 4: Proposed framework

Database section- Database section contains all the web pages of server. In this work, when user request for web page $P_{(x)}$ to server then before responds to the request, Server generates two threads i.e. $T_{P(X)}$ to registry section and $PS_{(X)}$ to decision section.

1. $\mathbf{T_{P(x)}}$ contains URI of the requested page $P_{(x)}$.
   Example:-
   /my_folder/modules/mod_superfishmenu/tmpl/js/jquery.event.hover.html
2. $\mathbf{PS_{P(x)}}$ contains the size of the requested web page $P_{(x)}$.
   Example:-
   Bytes transferred- 3595

Registry section- Registry section stores and maintains the log files of the web server. Log files contain present and past information of client-server transaction. In this work, when server generates thread $T_{P(X)}$ to registry section then it search the respected size of URI in $T_{P(x)}$ in XML representation of log file and generates a thread $T_{PSP(x)}$ to decision section contains size of the page $P_{(x)}$.

Decision section- Decision section generates the decision regarding web page tampering through matching techniques. Decision section performs the match of the incoming threads $PS_{P(x)}$ from database section and $T_{PSP(x)}$ from registry section.

$PS_{P(x)}$ is the size of the requested page P(x) taken from database section and $T_{PSP(x)}$ is the size of the same page taken from log file of registry section.

Then decision section matches both the sizes of the page $P_{(x)}$.

$$T_{PSP(x)} == PS_{P(x)}? \text{ M1: M2}$$

If match is positive it means requested page is not tampered. Generates message M1 for database section to response the requested page to respected client.

Otherwise if the match is negative its means requested page is tampered. And generate message M2 to web administrator for correction of the page $P_{(x)}$.

Currently proposed framework for detection of tampered web page is in the developing stage.

VII. CONCLUSION AND FUTURE WORK

The security experts have suggested various approaches to maintain the integrity of web page contents. But a number of web contents integrity related issues are still unsolved.

In this paper, the existing approaches to handle web content tampering are explored along with it the importance of log file in the detection of web content tampering is also presented. The size of requested web page from registry section and bytes transferred during the fetching of the page are key parameters. The proposed framework is very effective for static web page tampering and it will be very helpful for administrator to develop security policies. The future work will be focus on to enhance this approach to handle dynamic web page tampering.

REFERENCES

[1] Debarr D. , Ramanathan V. , Wechsler H., "Phishing detection using traffic behavior,spectral clustering and random forest ",Intelligence and Security Informatics (ISI),IEEE International conference on, seattle, WA, 4-7 July 2013,pp.67-72

[2] Sadeghian A. ,zamani M., Abdullah S.M. , "A Taxonomy of SQL Injection Attacks", Informatics and Creative Multimedia (ICICM) International conference on, kualalumpur, 4-7 Sept. 2013, pp. 269-273.

[3] Deepak Singh Tomar, "Web Forensics System on the Basis of Evidence Gathering with Code Injection Attack", International Journal of Computer Science & Communication, vol.1,issue 2, July 2010, pp. 313-315.

[4] Xianzhong Long; Hong Peng; Changle Zhang; Zheng Pan; Ying Wu, "A Fragile Watermarking Scheme for Tamper-Proof of Web Pages," Information Engineering, 2009. ICIE '09. WASE International Conference on , vol.2, no., pp.155-158, 10-11 July 2009.

[5] Tushar kanti; Vineet Richariya; Vivek Richariya;"Implementation of an Efficient Web Defacement Detection technique and Spotting Exact Defacement Location using Diff Algorithm," IJETAE, vol. 2, Issue 3, pp. 252-256, March 2012.

[6] Eric Medvet; Cyril Fillon; Alberto Bartoli;"Detection of Web Defacements by means of Genetic Programming," IEEE Third International Symposium on Information Assurance and Security, IAS 2007, pp. 227- 234, 29-31 Aug. 2007.

[7] Giorgio Davanzo; Eric Medvet ; Alberto Bartoli; "A Comparative Study of Anomaly Detection Techniques in Web Site Defacement Detection" Proceedings of the IFIP Tc 11 23rd International Information Security Conference on, vol. 278, pp. 711-716, 7-10 Sept. 2008.

[8] Xiang yang Liu; Hongtao Lu; "Fragile Watermarking Schemes for Tamper proof Web Pages" Springer Berlin Heidelberg 5th International Symposium on neural Networks, vol. 5264, pp.552-559, 24-28 Sept. 2008.

[9] Gurjwar R. K. ; Sahu D. R. ; Tomar D. S. ; "An approach to reveal website defacement", Int. J Computer Science and Information Security, vol.11, issue 6, pp. 73-79, 2013.

[10] Shadi Aljawarneh; Christopher Laing; Paul Vickers; "Verification of Web Content Integrity: A new approach to protecting servers against tampering," 8th Annual Post Graduate Symposium on the Convergence of Telecommunications Networking and Broadcasting, 28-29 June 2007.

[11] ShaimaaEzzat; Salama; Mohamed I. Marie; LailaM. El-Fangary; Yehia K. Helmy; "Web Server Logs Pre processing for Web Intrusion Detection,"Computer and Information Science, vol. 4, No 4, 19 June 2011,ISSN 1913-8989 (Print) ISSN 1913-8997 (Online).

[12] Aditi Shrivastava; Nitin Shukla; "Extracting Knowledge from User Access Log" International Journal of Scientific and Research Publications, Volume 2, Issue 4, April 2012, ISSN 2250-3153.

[13] Nikhil Kumar Singh; Deepak Singh Tomar; Bhola Nath Roy; "An Approach to Understand the End User Behavior through Log Analysis,"
International Journal of Computer Applications, 5(11), pp. 27–34, August 2010.

[14] Francis,A.; Ramachandran, R.; "Modulo Ten Search- An Alternative to Linear Search," IEEE International conference on Process Automation, Control and Computing (PACC), coinbatore, India, pp. 1-4, 20-22 July 2011.

[15] Yu Xu; Yannis Papakonstantinou; "Efficient LCA base Keyword Search in XML Data," CIKM '07 Proceeding of the sixteenth ACM conference on information and knowledge management, pp. 1007-1010, 06 Nov. 2007.

[16] Xingyuan Li;"A Search Algorithm Oriented to XML Keywords," International Journal of Computer Science Issues, Vol. 10, Issue 2, No 2, March 2013, ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784.

[17] Guoliang Li; Jianhua Feng; Lizhu Zhou; "Interactive Search in XML Data," Proceeding WWW '09 Proceeding of the 18th International conference on World wide web, pp. 1063,1064, 2009.