Implementation of Web Defacement Detection Technique

Rashmi K. Verma
M.E. Student, Department of Computer Engineering
Shah and Anchor Kutchhi Engineering College, Mumbai, Maharashtra, India

Shahzia Sayyad
Professor, Department of Computer Engineering
Shah and Anchor Kutchhi Engineering College, Mumbai, Maharashtra, India

Abstract- Websites are no longer merely about having an "Internet presence" today, but are also used for commercial transactions and to transfer sensitive data like personal information, credit card number, etc. This rapid proliferation of website has also spawned new threat to business and other organizations. Hackers are developing new techniques to deface website and steal the data from Web server. One kind of such attack on website is Website defacement attack. The term 'Website defacement' refers to unauthorized change of the content made either on a single web page (usually default web page) or on entire web site. The content of a defaced web page may be partially changed or it may be fully replaced by another page. Detection of website defacement automatically is very difficult because today web pages are highly dynamic and their degree of dynamism may vary widely across different pages. This paper proposes a hash code based web defacement detection mechanism. The proposed system includes the development of a module for an Apache web server for defacement detection in web pages, and configured it so that defaced web page should not be served by web server to the legitimate user. The proposed system prevents the legitimate user from accessing defaced web pages.

Keywords – Website Defacement, Hash Code, Apache Module.

I. INTRODUCTION

In today's era of Internet, the websites has an important role to fulfill user requirements. These websites are maintained to be secure. The attacker may hack a website without knowledge of developer and they may do any fraudulent activities on the website.

In our global business environment if a company does not have a Web site they can be viewed as not participant in today's economy. Website defacement is a common type of cyber attack. In the web defacement attack the invader changes the visual appearance of the webpage. It often refers to the unauthorized change of the content (usually the default page) of a website by an intruder. Website defacement is typically the work of system crackers, who break into a web server and replace the hosted website with one of their own [14]. In recent years hackers defaced several web sites by using techniques such as phishing, code injection, domain hack, XSS etc [15].

Web site defacement attacks were done to violate web integrity (correctness) by one of the following violations [13]:

- Change the content of a web page.
- Change any part of content of web page.
- Replace a web page entirely.
- Change the apparent of source of a web page.
- Redirect a web page.
- Destroy or delete a web page.

Due to wide usage of internet helps crackers gain more knowledge on vulnerabilities and exploitation techniques. As cyber criminals continue to develop and advance their techniques, they are also shifting their targets focusing less on theft of financial information and more on business espionage and accessing government information. Various security studies show that attacking websites to gain fame or money is definitely on the rise nowadays.

Common targets of defacement are religious, government web sites and bank web sites. Corporations are also targeted more often than other sites on the Internet [14]. Web sites represent the image of a company or organization and these are therefore suffering significant losses due to defacement. Visitors may lose faith in sites that cannot promise security and will become wary of performing online transactions. After defacement, sites have to be shut down for repairs, sometimes for an extended period of time, causing expenses and loss of profit and value.

There's an overwhelming need for a solution to fight against Web page defacement attack on web pages. The Web server would never present a defaced page to a user.
In order to circumvent this problem, we developed a module for Apache web server (by most measures, the leading server on the web today) which can detect defacement in a web page, and generate an error page if it is defaced otherwise send a web page to legitimate user.

The rest of the paper is organized as follows. Section II presents related work and Section III describes Proposed System. Methodology and design are explained in section IV. Implementation Details of proposed system presented in section V, followed by Results and Discussion in Section VI. Concluding remarks are given in section VII.

II. RELATED WORK

Alberto Bartoli et al. [1] proposed anomaly detection techniques for a web defacement monitoring service. Authors thought there should be systematic approach to fight against web defacement attack. Attractive options to these problems consist in augmenting availability and performance monitoring services with defacement detection capabilities. Author accesses the performance of earlier approach and found that earlier approaches based on anomaly detection are incapable of detecting web defacement automatically. An anomaly detection system constructs a profile of the monitored page automatically in learning phase, and raise an alert when the page content does not fit the profile. But previous approaches have flaws they do not cope with the problem of dynamism of web pages and also raise an excessive false alarm. Alberto Bartoli et al. [1] developed a feasible approach based on a dataset composed of 300 highly dynamic web pages that they observed periodically for 3 months and on a sample of 320 defacement extracted from ZoneH. In their approach , each detection algorithm is tested against its ability in not raising false alarm or missing defacement. There are some flaws it generates too many False Positive Ratio and False Negative Ratio.

Tushar Kanti et al. [2] enforced a website defacement detection mechanism and spot exact defacement location using diff algorithm. They also implemented a Web browser with inbuilt defacement detection techniques. In this approach, they calculated the hash code for web pages and compared with the saved hash code for the same page. If the hash code is found to be same then it is not defaced otherwise it will be marked as defaced. Authors provided a facility to the user so that user can spot exact lotion of defaced web page, for this purpose authors implemented a diff algorithm to show difference between original web page and defaced web page. Diff algorithm computes the comparison between two states of a page in order to spot the defacement. Authors also evaluated their approach against existing approach in terms of four parameters: space complexity, time complexity, number of defacement detection. Cost.

Ramniwas Kachhawa et al. [3] proposed a novel approach to detect web page tampering and stated the importance of log file in detection of web page tampering. Authors also presents a framework to detect tampered web pages based on log file analysis approach. Log files are extracted from web server and pre-processed to retrieve the required information. This information is represented in XML format because 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.

Ebot Ebot Enaw, Djoursoubo Pagou Prosper [4] proposed an approach to detect web defacement through Artificial Intelligence concept like anomaly detection, inference and machine learning. Earlier approaches to detect web defacements, have some flaws, many of the approaches generate many false positive, false negative especially in the present context where changes in web pages occur regularly and dynamically. Author intended to provide an intelligent approach to efficiently detect web defacement, identify the signature of web defacement attacks and self-improve the detection based on new types web defacement attack and even legitimate updates of websites. Authors designed a new architecture which identifies relevant criteria that can characterize the specificity of a web page and developed a module which will learn normal behavior of web pages based on those criteria. Web defacement trainer module aimed at assessing previous web defacement examples in an effort to identify the signature of an attacker and confirm web defacement. Authors also described a web crawler module that will crawl web pages and extract relevant data that will be used to identify attack. Analyzer component assesses information provided by the web crawler and compare them with data obtained from normal behavior trainer. If web defacement occurs authors designed a module which generates an alert message and sends it through SMS and Email to the web site administrator.

III. PROPOSED SYSTEM

The proposed system framework for protection against Web Defacement attack is depicted in figure 1.

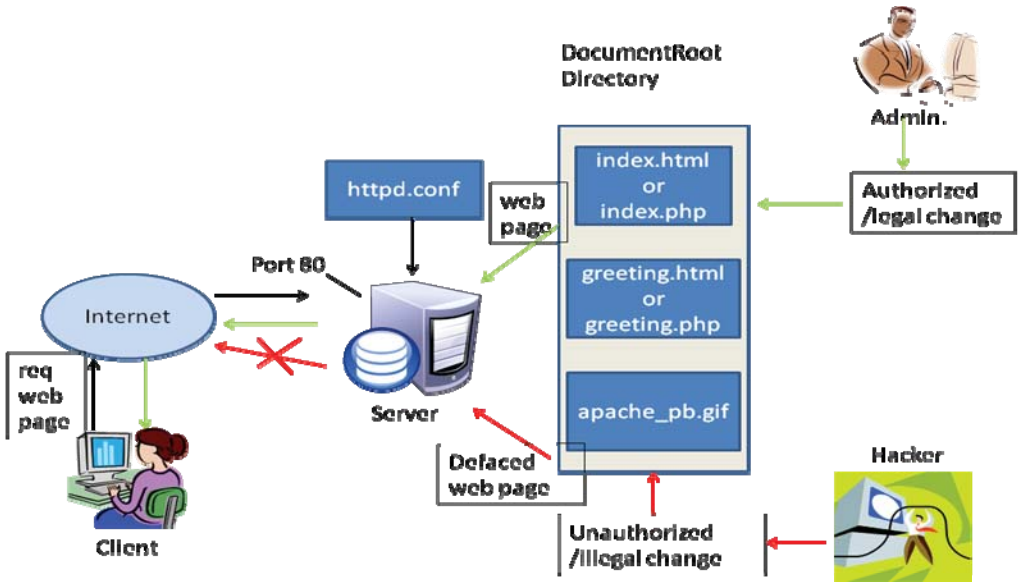


Figure 1. Proposed System Framework for protection against Web Defacement attack

When client request a web page request goes to the web server and it dispenses the requested web page to the client. Contents of the web page altered only by the person having authorized privileges which is an administrator but if someone (hacker) unauthorized way changes the contents of default web page (index page) or any other web pages it should not be delivered by web server to client.

The main objective of this work is detection of website defacement attack on the sites hosted on Apache web server and defaced webpage should not be delivered to the legitimate user. In order to achieve this we developed module for Apache Web Server which will calculates the MD5 hash code for the default web page and all other web pages stored at Document Root Directory and compares it with stored referenced hash code for the same page. If the result of comparison is matched means no web defacement so Apache web server will send requested web page to client and if a conflict occurs in both hash values ie. not matched means the web page is changed by hacker and Apache web server should not serve the defaced web page to the legitimate user.

IV. METHODOLOGY AND DESIGN

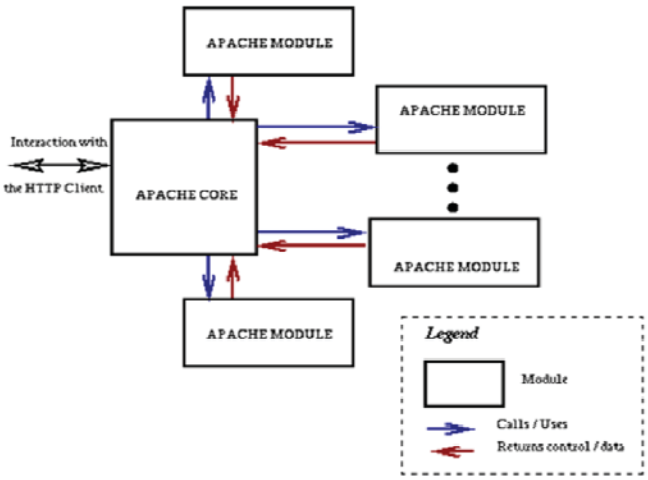
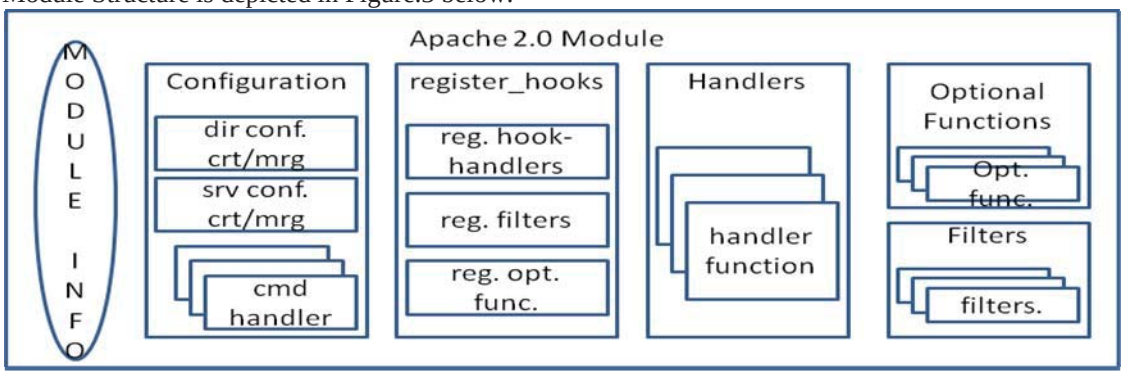


Figure 2. Apache 2.0 Architecture [7]

Apache 2.0 Architecture [7] comprises of an Apache core and number of Apache modules. Apache core which handles basic functionality of the server such as allocating a requests and maintaining and pooling all the connections . Once the request arrives to Apache server it forward that request to one of the Apache module which will serves that request. Apache web server is the only server having modular architecture, so that anyone can add to the basic functionality of the server without disturbing the basic core implementation. Due to its modularity, many features that are necessary within special application domains can be implemented as add–on modules and plugged into the server. Modules are pieces of code which can be used to provide or extend functionality of the Apache HTTP Server. Modules can either be statically or dynamically included with the core. For static inclusion, the module's source code has to be added to the server's source distribution and to compile the whole server. Dynamically included modules add functionality to the server by being loading as shared libraries during start–up or restart of the server. In this case the module mod_so provides the functionality to add modules dynamically.

Modules interact with the Apache server via a common interface. They register handlers for hooks in the Apache core or other modules. The Apache core calls all registered hooks when applicable, that means when triggering a hook. Modules on the other hand can interact with the server core via the Apache API. Using that API each module can access the server's data structures, for example for sending data or allocating memory. Each module contains a module–info, which contains information about the handlers provided by the module and which configuration directives the module can process. The module info is essential for module registration by the core [6]. Apache 2.0 Module Structure is depicted in Figure.3 below.



Apache 2.0 Module Structure

Figure 3. Apache 2.0 Module structure[6]

There are many modules are inbuilt module but some are listed below that come as part of the Apache HTTP Server distribution.

Apache Module mod_allowmethods
This module makes it easy to restrict what HTTP methods can used on an server.

Apache Module mod_auth_basic
This module allows the use of HTTP Basic Authentication to restrict access by looking up users in the given providers.

Apache Module mod_authz_user
This module provides authorization capabilities so that authenticated users can be allowed or denied access to portions of the web site.

Apache Module mod_so
On selected operating systems this module can be used to load modules into Apache HTTP Server at runtime via the Dynamic Shared Object (DSO) mechanism, rather than requiring a recompilation. On Unix, the loaded code typically comes from shared object files (usually with .so extension).

Apache Module mod_status
Provides information on server activity and performance [12].

APR MD5 Routines

For calculating the md5 of a file in apache module the APU module apr_md5 is used. The Apache Portable Runtime supports MD5 Routines having

Data Structure : struct apr_md5_ctx_t
Functions: apr_status_t apr_md5_init (apr_md5_ctx_t* context)
MD5 Initialize. begins an MD5 operation, writing a new context [8].

First the web page will be given as an input. The Apache web server calculates a hash code for requested web page at runtime and compared with the saved hash code for the same page. If the hash code is found to be same then it is not defaced otherwise it is will be marked as defaced. Apache web server will not send such defaced web page to the legitimate user.

V. Implementation Details

First Install latest version of Apache HTTP Server 2.4 on Linux operating system. For the development of an apache module a Content Generator Module and Filter module was implemented. An Apache module after development compiled using APXS tool.
In the first phase, we developed a content generator module [5] for Apache by building a Handler function (A handler is essentially a function that receives a callback when a request to the server is made) the purpose of this module will be to calculate and print out digest value MD5 for existing files on Apache web server.
In the second phase we created a database which holds translated file name and its hash code.
Last phase we developed a filter module which compares the stored hash code with hash code calculated at runtime.
Once a Apache module is compiled successfully. Perform all required changes in apache configuration file. The main configuration file of an Apache web server apache2.conf, it should be edited to include some directives such as AddHandler and AddOutputFilter in it. Also replaces entry LogLevel warn (all warning messages are logged here) with LogLevel info (Various information messages are logged here) in main configuration file. After making all required changes in configuration file, restart the Apache server. Check error log to see results.

VI. Result and Discussions

Suppose a web page (sampl.html) is requested by client through a web browser, the web page (sampl.html) is hosted on Apache web server in its Document Root Directory (/var/www/html/). Once the request is arrives at Apache web

server, in backend, the developed apache module calculates the hash code of the requested web page at runtime and compares with stored hash code if result of comparison is matched then apache web server send web page to client.

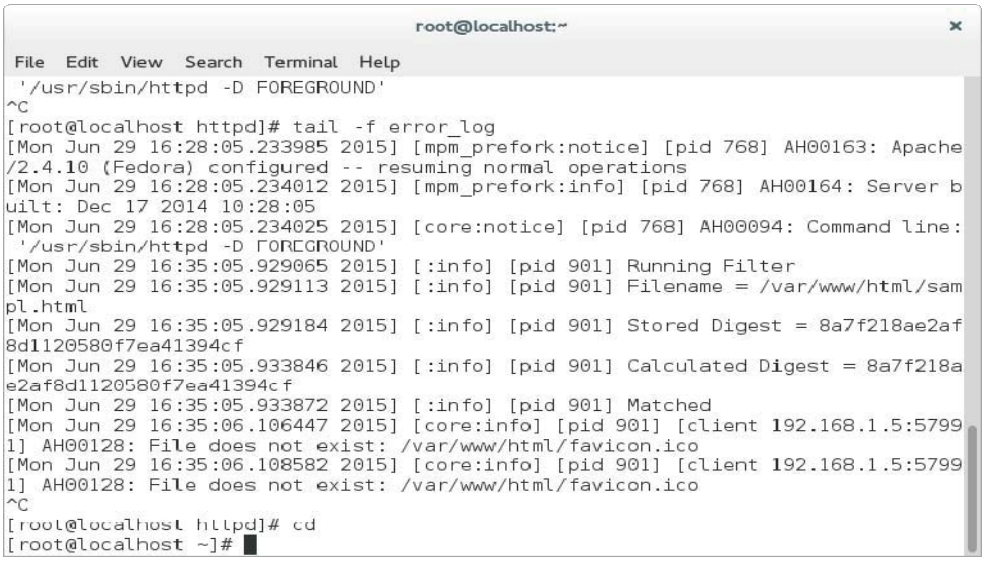


Figure 4. Error Log file contains result of comparison for web page (sampl.html)

VII.CONCLUSION

Web defacement problem is growing very aggressively and demands initiative on the part of security professionals in terms of action and education of others to address this significant threat. The detection of web defacement has been a great concern over the past couples of years for engineers and scientists which led to the publication of some article related web defacement detection approaches. To fight fast-spreading cybercrime, businesses and governments must collaborate globally to develop an effective model that can control the threat. Our approach consists of development of an apache module and configuring it to detect web defacement attack. We developed module for apache web server because apache web server is most popular web server and almost 56.2% of web sites were using it [16]. It was widely used all over the world because of open source and many other interesting features like it had many implemented and compiled modules which extend the functionality of its core. The proposed method of web page defacement detection is applicable to static web pages. Future work will include developing a module for apache web server which can detect web page defacement in dynamic web pages.

REFERENCES

[1] G. Davanzo; E. Medvet; A. Bartoli; “Anomaly detection techniques for a web defacement monitoring service”, ELSEVIER, Expert System with Application 38, 2011, 12521-12530

[2] 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.

[3] Ramniwas Kachhawa, Nikhil Kumar Singh, Deepak Singh Tomar “A Novel Approach to Detect Web Page Tampering” (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (3) , 2014, 4604-4607

[4] Ebot Ebot Enaw, Djoursoubo Pagou Prosper, “A Conceptual Approach to Detect Web Defacement Through Artificial Intelligence”, International Journal of Advanced Computer Technology (IJACT), Vol. 3, No. 6, 2319-7900.

[5] Nick kew, “The Apache Modules Book”, Application Development with Apache, Prentice Hall; 1 edition (26 January 2007)

[6] The Apache Modeling Project Documentation – FMC, Available: http://www.fmc-modelling.org/download/projects/apache/the_apache_modelling_project.pdf, January 2008

[7] Apache Server Architecture – ResearchGate. Available: http://www.researchgate.net

[8] Apache Portable Runtime : MD5 Routines, Available: http://apr.apache.org/docs/apr/2.0/group_apr_md5.html

[9] Title of PowerPoint Slide Presentation – UFMG, Available: http://homepages.dcc.ufmg.br

[10] Log Files – Apache HTTP Server Version 2.4, (2011, Sep 9), Available: http://httpd.apache.org/docs/2.4/logs.html

[11] Configuration Files – Apache HTTP Server Version 2.4, Available: http://httpd.apache.org/docs/2.4/configuring.html

[12] Apache HTTP server version 2.2 Module index, Available: http://httpd.apache.org/docs/2.2/mod/

[13] Charles P. Pfleeger and Shari Lawrence “Security in Computing”, 3rd Edition, , Prentice Hall – 2003

[14] Website Defacement, Available: https://en.wikipedia.org/wiki/Website_defacement
[15] Website Defacement and Domain Hacking, Available: http://rajstudy.weebly.com/website-defacement--domain-hacking.html
[16] Usage statistics and market share of Apache for websites, Available: http://w3techs.com/technologies/details/ws-apache/all/all, oct 2015