Detection of Web Application Attacks with Request Length Module and Regex Pattern Analysis

Ei Ei Han

Abstract Web application attack detection is one of the popular research areas during these years. Security for web application is necessary and it will be effective to study and analyze how malicious patterns occur in web server log. This system analyzes web server log file, which includes normal and malicious users' access patterns with their relevant links. This uses web server log file dataset for the detection of web application attacks. This system intends to analyze normal and attack behaviors from web server log and then classify attack types which are included in the dataset. In this system, three types of attacks are detected namely, SQL injection, XSS and directory traversal attacks. Attack analysis stage is done by request length module and regular expressions for various attack patterns.

1 Introduction

Web applications are becoming increasingly popular and complex in all sorts of environments, ranging from e-commerce applications to banking. The security of web applications has become increasingly important and a secure web environment has become a high priority for e-business communities. They are subject to all sorts of attacks. In today's times, the most critical issue for any web application is security. Web servers and web-based applications are popular attack targets. To detect web-based attacks, intrusion detection systems are configured with a number of signatures that support the detection of known attacks.

This system differentiates normal access patterns from malicious access patterns. It can detect how malicious users try to attack the web site. The system

E.E. Han(✉)
University of Computer Studies, Yangon, Myanmar
e-mail: eieihan.ucsy@gmail.com

© Springer International Publishing Switzerland 2016

T.T. Zin et al. (eds.), *Genetic and Evolutionary Computing*,
Advances in Intelligent Systems and Computing 388,
DOI: 10.1007/978-3-319-23207-2_16

can know which pages or links are most accessed and are tried by malicious users. It also describes successful attacked (attack gained) web pages and links. This system will be effective for the security of web application system and analysis on web server log. There are two fundamentally different attack detection methods – rule-based detection (static rules) and anomaly-based detection (dynamic rules).

Web server log analysis is a rule-based detection mode which concentrates on web attacks which are visible in default web server log files like Apache or IIS. This system combines traditional web usage mining system with security analysis.

2 Background Theory

2.1 Web Usage Mining

Web usage mining is the process of extracting useful information by analyzing web usage data from server logs. It is defined as an application of data mining techniques on the navigational traces of the users to extract knowledge about their preferences and behavior. Web usage mining involves three major phases namely, pre-processing, pattern discovery and pattern analysis. Some of the techniques used in Pattern discovery are Association rules, Classification, Clustering etc. Pattern Analysis filters out uninteresting rules or patterns found in the pattern discovery phase.

2.2 Web Application Attacks and DVWA

Malicious users try to attack a web site or web server by using various attack patterns. Web application attacks are occurred by performing Web application queries. They take the forms of well defined strings and parameters. These are recorded in the web server log file. By analyzing each record of server log file, malicious patterns can be detected. These patterns include some special and encoding characters. To classify the web based attacks, it is needed preparing for the input data like URL decoding and regular expression, etc.

One of the popular web application attack tools is DVWA. Damn Vulnerable Web App (DVWA) is a PHP/MySQL web application that is damn vulnerable. Its main goals are to be an aid for security professionals to test their skills and tools in a legal environment, help web developers better understand the processes of securing web applications. It will be used for launching web application attacks and logging them. With this tool, popular web based attacks can be created and stored in the database.

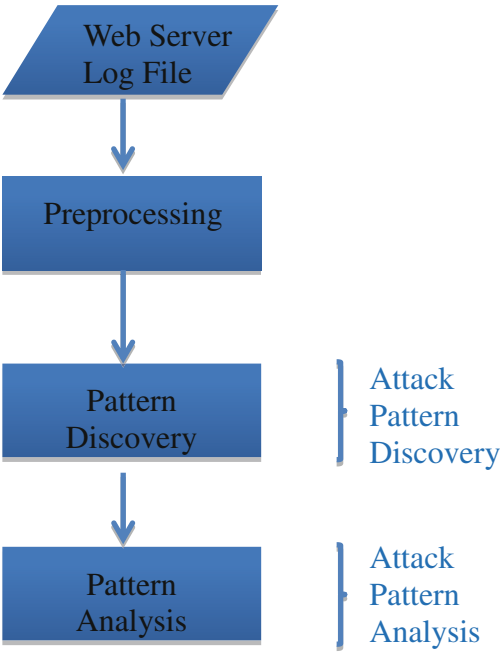


Fig. 1 Web Usage Mining Framework Combined with Intrusion Detection

Examples of web server log file by testing with DVWA are as follows:

127.0.0.1 - [23/Apr/2014 12:55:31 +0630] "GET /DVWA-1.0.8/ HTTP/1.1" 200 4618 "-" "Mozilla/5.0 (Windows NT 6.1; rv:27.0) Gecko/20100101 Firefox/27.0"
127.0.0.1 - [23/Apr/2014 12:55:35 +0630] "GET /DVWA-1.0.8/vulnerabilities/xss_r/ HTTP/1.1" 200 4456 "http://localhost/DVWA-1.0.8/" "Mozilla/5.0 (Windows NT 6.1; rv:27.0) Gecko/20100101 Firefox/27.0"
127.0.0.1 - [23/Apr/2014 12:55:50 +0630] "GET /DVWA-1.0.8/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%2Fxss%2F%29%3C%2Fscript%3E HTTP/1.1" 200 4514 "http://localhost/DVWA-1.0.8/vulnerabilities/xss_r/" "Mozilla/5.0 (Windows NT 6.1; rv:27.0) Gecko/20100101 Firefox/27.0"
127.0.0.1 - [23/Apr/2014 12:56:20 +0630] "GET /DVWA-1.0.8/vulnerabilities/xss_r/?name=%3CScriPt%3Ealert%28%2Fxss%2F%29%3C%2FScripT%3E HTTP/1.1" 200 4514 "http://localhost/DVWA-1.0.8/vulnerabilities/xss_r/?name=%3Cscript%3Ealert%28%2Fxss%2F%29%3C%2Fscript%3E" "Mozilla/5.0 (Windows NT 6.1; rv:27.0) Gecko/20100101 Firefox/27.0"

The web server log file which contains attacks includes URL encoding characters. So, it is needed to decode these characters to get attack patterns. Percent ("%") character serves as the indicator for percent-encoded octets. It is the practice of translating unprintable characters or characters with special meaning within URLs to a representation that is unambiguous and universally accepted by web browsers and servers. When you pass information through a URL, you need to make sure it only uses specific allowed characters that have meaning in the URL string.

2.3 Keyword Removal (Signature Based Detection)

Input filtering describes the process of validating all incoming data. “Suspicious” input that might contain a code injection payload is either rejected, encoded, or the “offensive” parts are removed using so called “removal filters”. The protection approach implemented by these filters relies on removing predefined keywords.

Different kinds of attacks have different keywords. For example, SQL injection attack has the keywords like SELECT, INSERT, UPDATE, DELETE, UNION, etc.XSS attack has the keywords like<script, javascript, or document. Directory Traversal attack has the keywords like “dir, cmd, windows, ../”, etc.

Firstly, attack types are detected by their respective keywords in the system. This process is also known as keyword detection for each attack type.

3 Implementation of the System

3.1 Request Length Module

The length of the input requests to the web server can be used to detect the occurred attacks. If μ is considered to be the average length of n requests received by an application with the parameters of L1,L2,L3,……..,Ln in which Li represents the length of the received requests of i, and σ2 will be the variance of these requests, then according to the equation 1, the possibility of P for a request with the length of L will be as the following.

$$p = \frac{\sigma^2}{(l-\mu)^2} \tag{1}$$

The values of μ and σ2 are calculated separately in the education phase according to the received requests. After calculating these values in the test phase and while considering the pre-defined values and the size of the newly received request, the value of P will be calculated and if it’s higher than a threshold, that request will be considered as an anomaly request. In figure 2, you can see an example of the normal and abnormal requests which can be detected by this module. This method can easily detect attacks like Directory Traversal and Buffer

overflow. Because these attack inherently, have request sizes larger than the normal size. [1]

3.2 Regular Expression Patterns

The goal of a regular expression is to match a certain expression within a lump of text. A regular expression pattern is usually enclosed within slashes ('/').Regular expressions enable a powerful, flexible, and efficient text processing. This system can analyze how attack log file occurred by using DVWA web server. By inputting some attack patterns from input box and by POST method, we can analyze how certain types of attacks occurred in web server log file.

Compute Possibility

id	URL	mean	sigma	possibility	Result
1	/	65.93	5215.44	1.24	is normal
2	icons/poweredby.png	65.93	5215.44	2.37	is normal
3	icons/apache_pb.gif	65.93	5215.44	2.37	is normal
4	favicon.ico	65.93	5215.44	1.73	is normal
5	/	65.93	5215.44	1.24	is normal
6	adduser.php	65.93	5215.44	1.73	is normal
7	/adduser_submit.php	65.93	5215.44	2.37	is normal
8	/adduser.php	65.93	5215.44	1.79	is normal
9	/login_submit.php	65.93	5215.44	2.18	is normal
10	/login.php	65.93	5215.44	1.67	is normal
11	/login.php?op=login	65.93	5215.44	2.37	is normal
12	register.php	65.93	5215.44	1.79	is normal
13	register.php?op=reg	65.93	5215.44	2.37	is normal
14	/?_test1=c:\\windows\...	65.93	5215.44	.32	is normal
15	/register.php?regna...	65.93	5215.44	1.60	is normal
16	/register.php?regna...	65.93	5215.44	26.35	is normal
17	/register.php?regna...	65.93	5215.44	7.67	is normal
18	/?mode=a)/)/)/)/)/)/)/)...	65.93	5215.44	.26	is normal
19	/register.php?regna...	65.93	5215.44	1.09	is normal
20	/?mode=RihEE</->Ri...	65.93	5215.44	82.91	is attack
21	/register.php?regna...	65.93	5215.44	1218.39	is attack
22	/index.php?view=../../.....	65.93	5215.44	141.60	is attack
23	/?mode=<!--#exec c...	65.93	5215.44	36.64	is normal
24	/index.php?view=Oel...	65.93	5215.44	.12	is normal
25	/index.php?view=%n...	65.93	5215.44	.23	is normal
26	/index.php?view=../../.....	65.93	5215.44	315.01	is attack
27	/index.php?view=39" ...	65.93	5215.44	4.27	is normal
28	vulnerabilities/xss_s/ ...	65.93	5215.44	4.27	is normal
29	/index.php?view=%S...	65.93	5215.44	6.23	is normal

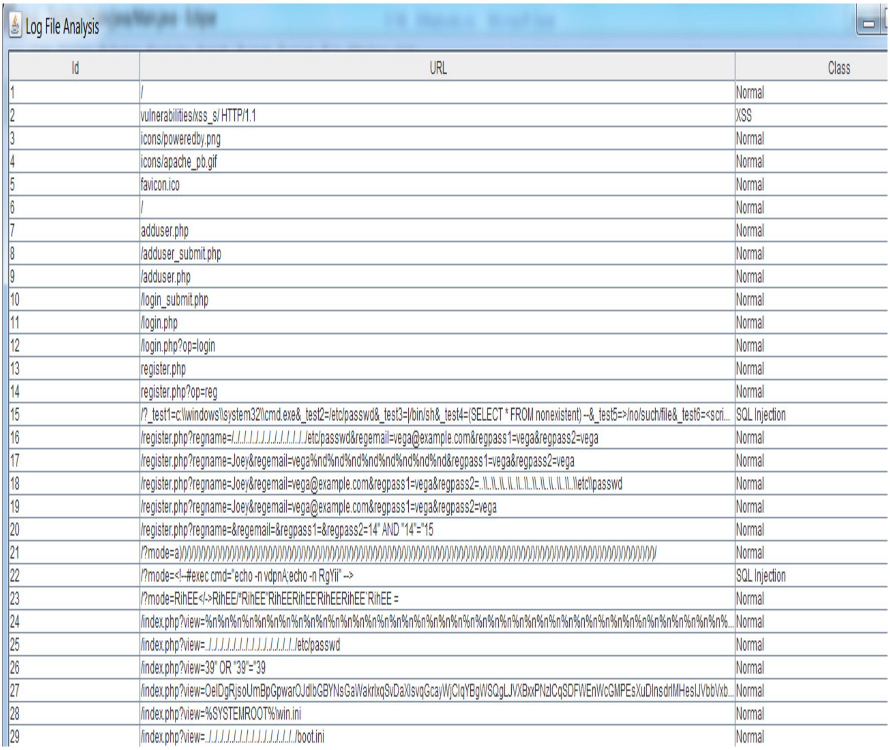
Fig. 2 Attack Detection with Request Length Module

In this system regex patterns for three web attacks and normal (attack free) patterns are predefined. Some patterns of regular expression used in this system are as follows:

[a-z A-Z]*/[a-z]+_s/ [A-Z]+/1.1=XSS
/((\%3C)|<)((\%2F)|\/)*[a-z0-9\%]+((\%3E)|>)/ix =XSS
/((\%3C)|<)((\%69)|i|(\%49))((\%6D)|m|(\%4D))((\%67)|g|(\%47))[^\n]
+((\%3E)|>)/I=XSS

/(javascript|vbscript|expression|applet|script|embed|object|iframe|frame|
frameset)/i=XSS
/(java)?script/i=XSS
/(\.|(%|%25)2E)(\.|(%|%25)2E)(\/|(%|%25)2F|\\|(%|%25)2C)/=Directory
Traversal
/exec(\s|\+)+(s|x)p\w+/ix=DirectoryTraversal
/(\||%00|system\(|eval\(|'|\\)/i=Directory Traversal
/(\')|(\%27)|(\-\-)|(#)|(\%23)/ix=SQL Injection
/((\%3D)|(=))[^\n]*((\%27)|(\')|(\-\-)|(\%3B)|(;))/i=SQL Injection
/\w*((\%27)|(\'))(\s|\+|\%20)*((\%6F)|o|(\%4F))((\%72)|r|(\%52))/ix
=SQLInjection
/((\%27)|(\'))(select|union|insert|update|delete|replace|truncate)/ix=SQL
Injection

By computing regex pattern analysis, the system can show the result as in figure 3.


Log File Analysis

Id	URL	Class
1	/	Normal
2	vulnerabilities/xss_s/ HTTP/1.1	XSS
3	icons/poweredby.png	Normal
4	icons/apache_pb.gif	Normal
5	favicon.ico	Normal
6	/	Normal
7	adduser.php	Normal
8	/adduser_submit.php	Normal
9	/adduser.php	Normal
10	/login_submit.php	Normal
11	/login.php	Normal
12	/login.php?op=login	Normal
13	register.php	Normal
14	register.php?op=reg	Normal
15	/?_test1=c:\\windows\\system32\\cmd.exe&_test2=/etc/passwd&_test3=|/bin/sh&_test4=(SELECT * FROM nonexistent) --&_test5=>/no/such/file&_test6=<scri...	SQL Injection
16	/register.php?regname=/../../../../../../../../../../../../../etc/passwd&regemail=vega@example.com&regpass1=vega&regpass2=vega	Normal
17	/register.php?regname=Joey&regemail=vega%nd%nd%nd%nd%nd%nd%nd%nd&regpass1=vega&regpass2=vega	Normal
18	/register.php?regname=Joey&regemail=vega@example.com&regpass1=vega&regpass2=..\\..\\..\\..\\..\\..\\..\\..\\..\\..\\..\\..\\..\\etc\\passwd	Normal
19	/register.php?regname=Joey&regemail=vega@example.com&regpass1=vega&regpass2=vega	Normal
20	/register.php?regname=&regemail=&regpass1=&regpass2=14" AND "14"="15	Normal
21	/?mode=a))/	Normal
22	/?mode=<!--#exec cmd="echo -n vdpnA;echo -n RgYii" -->	SQL Injection
23	/?mode=RihEE</->RihEE/*RihEE"RihEERihEE'RihEERihEE`RihEE =	Normal
24	/index.php?view=%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%n%...	Normal
25	/index.php?view=../../../../../../../../../../../../../../../../etc/passwd	Normal
26	/index.php?view=39" OR "39"="39	Normal
27	/index.php?view=OeIDgRjsoUmBpGpwarOJdIbGBYNsGaWakrlxqSvDaXlsvqGcayWjClqYBgWSQgLJVXBxxPNzlCqSDFWEnWcGMPEsXuDInsdrlMHesIJVbbVxb...	Normal
28	/index.php?view=%SYSTEMROOT%\win.ini	Normal
29	/index.php?view=../../../../../../../../../../../../../../../../boot.ini	Normal

Fig. 3 Attack Classification with Regex Pattern Analysis

3.3 System Flow Diagram

Input to the system is web server log file. After preprocessing, web log data are stored in the database. URL decoding is performed on the data. First step of request length module is computed. The next step, regex pattern analysis is performed and attack detection results are produced.

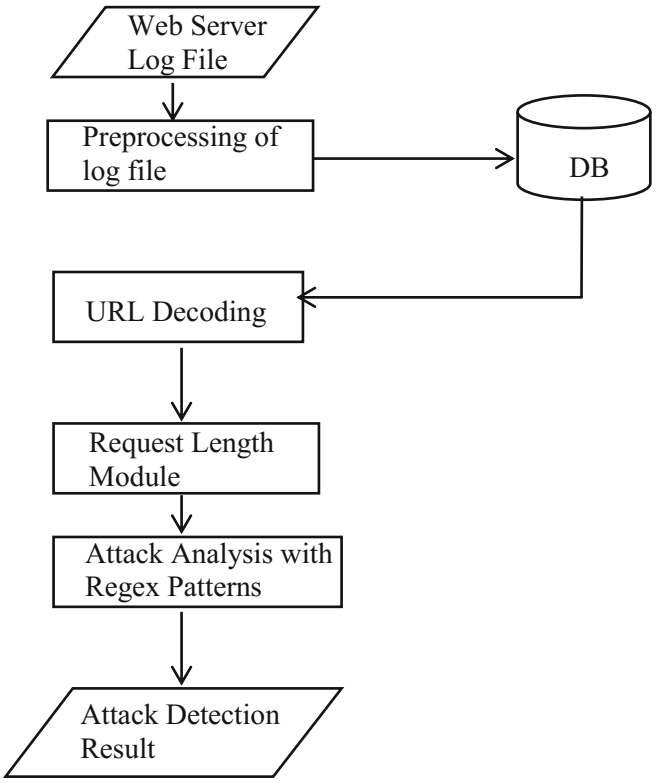


Fig. 4 System Flow Diagram

4 Conclusion and Experimental Results

This system presents about analyzing and classifying web application attacks. Combination of request length module and regular expression patterns are used in this system. Three types of attacks namely; SQL injection, XSS and directory traversal can be effectively classified by this system. Other attacks in the dataset that are not covered by this system will be resulted as unknown attacks. By computing request length module, unseen attacks can be detected. Predefined regex pattern analysis cannot be covered in some cases. For these condition, the system results as unknown attacks.

Detection rate of request length module and regex pattern analysis for each attack type is shown in figure 5. By analyzing this result, directory traversal attack can be more effectively detected than the other two attacks in request length module. By regex pattern analysis, SQL injection attack detection rate is higher than the others. Request length module is effective for unknown attacks. For known attacks and with certain patterns, regex pattern analysis is an effective method. Regex patterns in this system are for three types of attacks and normal access patterns. The experimental results are computed based on the dataset received by DVWA.

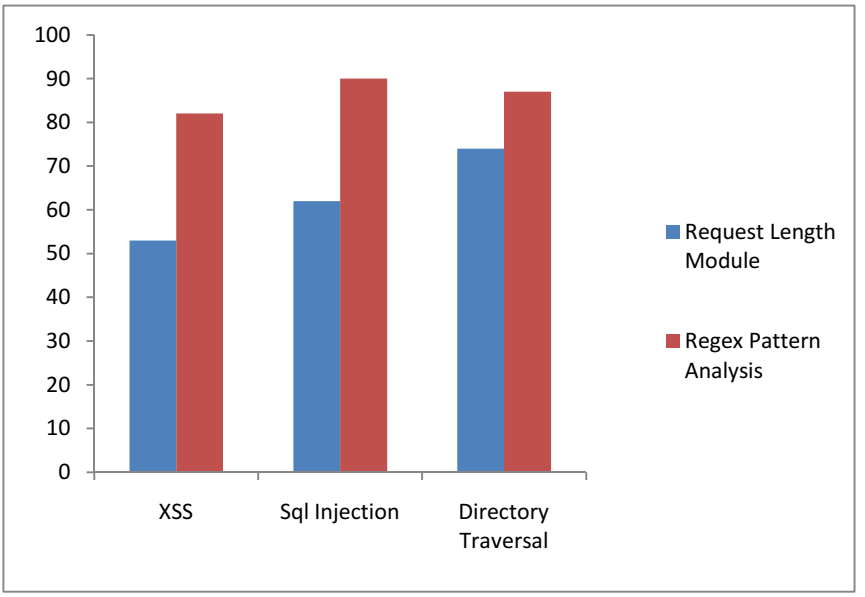


Fig. 5 Detection Rate of Three Attacks by Request Length Module and Regex Pattern Analysis

References

1. Vamsidhar, T., Ashok, R., Venkat, R.: Intrusion Detection System For Web Applications With Attack Classification. Journal of Global Research in Computer Science (2012)
2. Kruegel, C., Vigna, G., Robertson, W.: A multi-model approach to the detection of web-based attacks. Reliable Software Group. University of California, Santa Barbara (2005)
3. Meyer, R., Cid, C.: Detecting Attacks on Web Applications from Log Files. SANS Institute (2008)
4. Kruegel, C., Vigna, G.: Anomaly detection of Web-based attacks. In: Proceedings of the 10th ACM Conference on Computer and Communication Security(CCS 2003), Washington, DC, October 2003, pp. 251–261. ACM Press, New York (2003)

5. Mookhey, K.K., Burghate, N.: Detection of SQLInjection and CrosssiteScriptingAttacks (2004). http://www.blackhat.com/presentations/bhusa04/bhus04mookhey/old/bhus04mookhey_whitepaper.pdf
6. Robertson, W., Vigna, G., Kruegel, C., Kemmerer, R.: Using generalization and characterization techniques in the anomaly based detection of web attacks. In: 13th Annual Network and Distributed System Security Symposium, San Diego (2006)
7. Gallagher, B., Eliassi-Rad, T.: Classification of http attacks: A studyon the ecml/pkdd 2007 discovery challenge (2009)
8. Faradzhullaev, R.: Analysis of Web Server Log Files and Attack Detection. Institute of Information Technologies, Academy of Sciences of Azerbaijan (2007)