Detection of Web-Based Attacks by Analyzing Web Server Log Files

Nanhay Singh, Achin Jain, Ram Shringar Raw and Rahul Raman

Abstract In today's scenario, Web traffic is increasing everyday in the world and has overtaken P2P traffic. The Websites are getting hacked on daily basis. These rises in hacking activity pose a greater threat than the network attacks as they threaten to steal crucial and important information from Website. This information can be related to the users, employee, and other important data stored in applications and database linked to the Website. Increase in Web network traffic has opened new and more efficient attack vectors for the hackers and attackers to work with. Attackers take advantage of the vulnerability in traditional firewalls deployed on Website. These firewalls are not designed to protect Web applications; lots of Websites are getting attacked by malicious scripts and users. In this paper, many Web attacks are carried out on Web applications hosted on local server to analyze the log file created after the attacks. A Web application log file allows a detailed analysis of a user action. We have simulated some Web attacks using MATLAB. Results extracted from this process helps in the recognition of majority of the attacks and helps in prevention from further exploitation.

Keywords Web attacks · Web server log file · Buffer overflow attack · iFrame injection attack

N. Singh (✉) · A. Jain · R. S. Raw
Ambedkar Institute of Advanced communication Technologies and Research, Delhi, India
e-mail: nsingh1973@gmail.com

A. Jain
e-mail: achin_jain25@yahoo.com

R. S. Raw
e-mail: rsrao08@yahoo.in

R. Raman
National Institute of Technology, Rourkela, India
e-mail: rahulraman2@gmail.com

D. P. Mohapatra and S. Patnaik (eds.), *Intelligent Computing, Networking, and Informatics*, Advances in Intelligent Systems and Computing 243,
DOI: 10.1007/978-81-322-1665-0_10, © Springer India 2014

1 Introduction and Background

Web applications are becoming increasingly popular and complex in all sorts of environments ranging from e-commerce to banking applications. As a consequence, Web applications are subject to all sorts of attacks. The consequences of attacks can be very severe, like identity supplanting, sensitive data hijacking, access to unauthorized information, modification of Web page content, malicious script execution, etc. Therefore, it is a very necessary and important task to protect Web application and adopt suitable security methods.

In this paper, we describe the various Web attacks with experimental results and analysis of the attacks. The rest of the paper is organized as follows: Sect. 2 gives brief overview of the attack on Web applications. In Sect. 3, descriptions of Web server log files is given in detail, which plays an important role in the work carried out in this article. Section 4 explains the problem formulation of the work. The experimental work and result analysis of the problem are carried out in Sect. 5 with comparative analysis between iframe injection and buffer overflow attack. Finally, Sect. 6 concludes the paper.

2 Web-Based Attacks

A Web attack is defined as the unwanted intrusion to Website resources. Cross-site scripting (XSS) attack refers to a range of attacks in which the attacker injects malicious code mostly JavaScript into a Web application [1, 2]. According to [3], more than 60 % of Websites are vulnerable to XSS attacks. SQL injection attack is considered to be one of the most critical cyber attacks and vulnerabilities related to *SQL* injection have been described as one of the most serious threats for Web applications [4, 5]. In this attack, the attacker tries to gain control over Web application database by exploiting vulnerability present. There are numerous approaches to launch SQL injection attack discussed in [6]. CSRF vulnerability occurs when a Website has inadequate mechanism to check whether a valid request has been sent intentionally or unintentionally by a logged-in user [7]. In CSRF attack, the attacker forces victim Web browser to perform an unwanted action on a trusted Website without user's interaction in that action. CSRF attack has been identified to be among the top four most common vulnerabilities present in today's Web-based programs [8].

There are many solutions available to counter these attacks such as firewall, but these features are not always enough to protect the users from being attacked. As a result, users are vulnerable to exploitations while performing basic functionalities (e.g., logging in) [9, 10]. In this paper, we restrict our research work to five well-known vulnerabilities explained in The Open Web Application Security Project (OWASP) [8].

3 Web Server Log Files

The Web logs are used to track the end-user behavior. Log files are those files that list the actions that have been occurred on the Web applications [11]. Web server creates and maintains log files for the purpose of getting feedback about the activity and performance of the server and the problems occurring in the Web server [12]. Log files play a very important role in the detection of attack on the Web application as analysis of log files helps in identifying anomalies in the request to the server and difference in normal request response from malicious request response. By studying Web server log files, it is possible to create rules based on regular expression as in case of various attacks such as XSS attack, iframe injection attack, and *SQL* injection attack. In other attacks, Web server log file analysis helps in creating rules based on parameter value anomaly such as "referrer" field in CSRF attack and "bytes" field in buffer overflow and iframe injection attack.

4 Problem Formulations

Web server log analysis is a rule-based detection method which is used for the analysis of Web attacks which are visible in default Web server log file like Apache or *Internet Information Server* (IIS). Detecting Web attacks is not a very simple process as there are a lot of attack vectors which should be known to make efficient detection rules, and it is very important to identify as many attack vectors as possible. Another problem is standardization of encoding method used in log files. A well-defined set of regular expressions and rules allows the identification of many of the critical Web application attacks.

In this paper, we have carried out study and analysis of Web attacks and their impact on Web server log files. For analyzing attack's effect, we have discussed two different attacks i.e., iframe injection attacks and buffer overflow attack on vulnerable Web application hosted on local server. Buffer overflow and iframe injection attacks are stimulated on Web applications using various scripts. After study of Web attacks, we accessed Web server log files (Apache server in this work) and then analyzed the difference in normal and malicious requests. In this work, we have also carried out comparative analysis of buffer overflow and iframe attack using result graphs generated with the help of MATLAB. In comparative work, we analyzed the effect of transferred bytes in both the attacks. Before introducing the experimental and result analysis work, we briefly describe all the five Web-based attacks one by one given in the subsections.

4.1 iFrame Injection Attack

The iframe stands for in-line frame, and this tag is used to insert contents from other Websites or server. This tag can be used by the attacker to inject malware

containing Websites or links using XSS attack. The attack is stimulated to show the difference between the sizes of bytes returned for the same request. In normal request, the URL looks as shown in the box below.

http://localhost/Test.php?yourname=Achin+Jain

When iframe injection was injected in the request, the *URL* looks as shown in the box below.

http://localhost/Test.php?yourname=Achin+Jain%3Ciframe+src%3D%94%3A%2F%
2Fdisney.com%2F%94+width%3D1+height%3D1+style%3D%94visibility%3Ahidden%
3Bposition%3Aabsolute%94%3E%3C%2Fiframe%3E

4.2 Buffer Overflow Attack

In buffer overflow attack, the attacker tries to exploit the very common vulnerability in the Web server of not validating the input properly. A buffer overflow occurs during program execution when a fixed-size buffer has had too much data copied into it. This causes the data to overwrite into adjacent memory locations and depending on what is stored there. The behavior of the program itself might be affected [13]. In this paper, the buffer overflow attack is tried on "*php*" page with a field to accept the name of the user and to stimulate the work. The length of the field is set to accept only 5 characters. In the first scenario, the legitimate input is passed to the "*php*" file by entering value through HTML form, and in the second scenario, the value of the parameter is directly passed from the *URL*. In the first attempt, the *URL* with correct input from HTML file looks as shown in the box below.

http://localhost/Test.php?yourname=achin

In the second attempt, the parameter value is directly passed to the "Test.php" from *URL* and *URL* looks like as shown in the box below.

http://localhost/Test.php?yourname=achinachinachinachinachinachinachinachina
chinachinachinachinachinachinachinachinachinachinachinachinachinachina
chinachinachinachinachinachinachinachinachinachinachinachachinachina
chinachinachinachinachinachinachinachinachinachinachinachinachinachina
chachinachinachinachinachinachinachinachinachinachinachinachinachina
chinachinachinachinachinachinachinachinachinachinachinachinachinachina
chinachinachinachinachinachinachachinachinachinachina

5 Experimental Works and Result Analysis

5.1 Experimental Works

In this section, the experimental work has been carried out to study the effects on Web server log file due to various attacks on the vulnerable Web application hosted in local Web server.

5.1.1 iFrame Injection Attack

In this work, our main focus is to analyze the log files for both the requests and observe what is the change in the amount of bytes that returned from the server. Log record showing normal request is given below in Fig. 1, which clearly shows that the amount of transferred bytes while serving the request is 64.

In Fig. 2, the log showing record with iframe injection attack is shown. Through this record file, it is clear that bytes transferred while serving same request is 167 and additional 226 due to error in injecting Website. Therefore, a total of 393 bytes is transferred from the server.

5.1.2 Buffer Overflow Attack

The main focus is to analyze the log file and identify the changes in the Web server log file. Figure 3 shows the log file of the normal request, and Fig. 4 shows the log file with buffer overflow attack.

It is clearly visible from both the log files that a parameter value of any length can be sent via URL. The value used is very short, but the bytes that server has to send as response increases from 59 to 564 which is approximately 9.5 times larger. If the input value increases to a larger extent, then it is very likely that Web server will crash down.

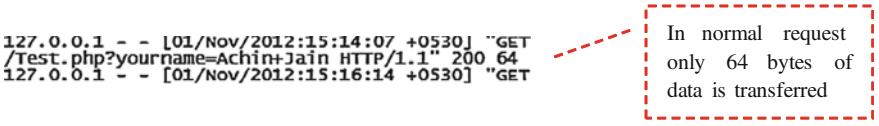

Fig. 1 Log file for normal request

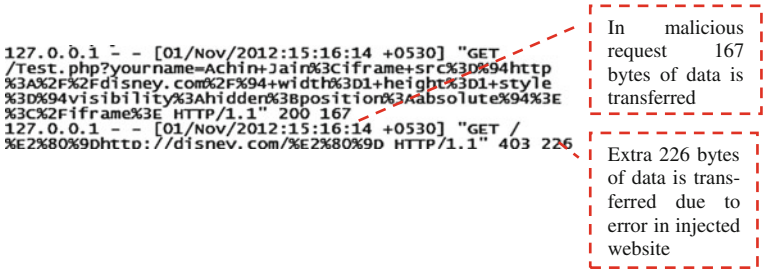

Fig. 2 Log file showing iframe injection attack

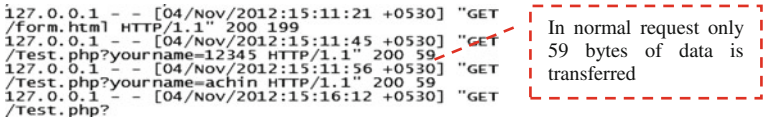

Fig. 3 Log file record with input from HTML file

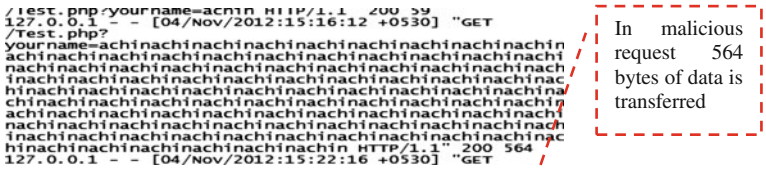

Fig. 4 Log file record in buffer overflow attack

5.2 Result Analysis

In this section, we have carried out the comparative analysis between buffer overflow and iframe injection attack to distinguish which attack is more severe. The parameter that we have considered in this work is the amount of bytes transferred. In the first step, we have simulated iframe injection attack, and various inputs that are tested are listed in Table 1. First five input parameters passed are normal, and the amount of bytes transferred from the server while responding back to this normal request is 59. The next input contains script for iframe injection attack, and for these attacks, bytes transferred increased accordingly as shown in the form of graph in Fig. 5.

In the second step, we have simulated buffer overflow attack, and inputs that are tested are listed in Table 2. First five input parameters passed are normal, and the amount of bytes transferred from the server while responding back to this normal request is 59. The next input contains input having length more than the expected length (5 in our work), and for these attacks, bytes transferred increased exponentially as shown in the form of graph in Fig. 6.

For the comparative analysis, we have simulated both the attacks together, and the result is shown in Fig. 7. In the figure, it is clear that impact of iframe injection attack depends on the Website injected in the script, and the impact of buffer overflow injection attack is directly proportional to the length of the injected input.

After analysis of iframe injection and buffer overflow attack, we can see in figure that both the attacks can be used against the Website and applications hosted on Web server to increase illegitimate traffic. From the graph, it is clear that the

Table 1 Input parameter and bytes transferred for iframe injection attack

S. No	Input	Bytes transferred
1	abcde	59
2	12345	59
3	@#$ %^	59
4	ABCDE	59
5	A!2C#	59
6	Achin Jain < iframe src = http://www.disney.com/>	101
7	Achin Jain < iframe src = http://www.w3schools.org/>	104
8	Achin Jain < iframe src = http://www.imdb.com/>	99

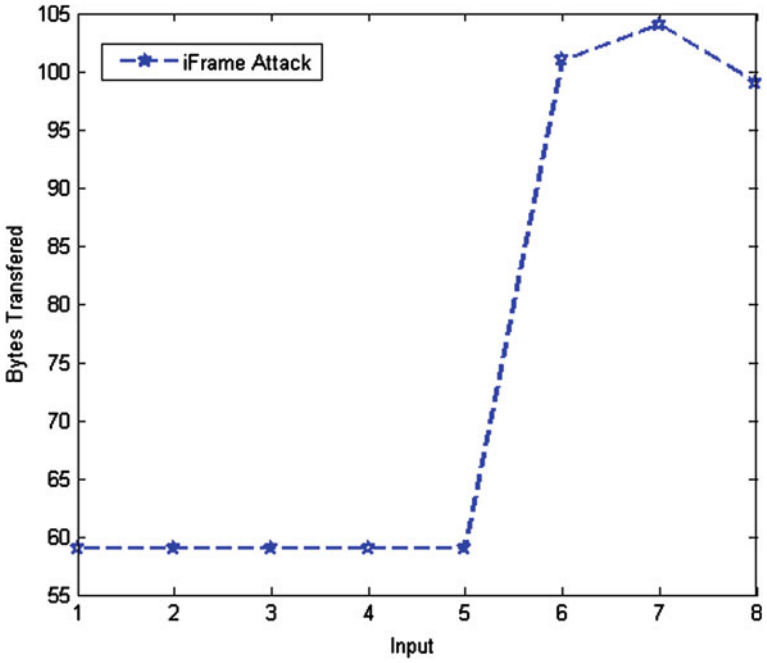

Fig. 5 Graph for iframe injection attack

Table 2 Input parameter and bytes transferred for buffer overflow attack

S. no	Input	Bytes transferred
1	abcde	59
2	12345	59
3	@#$ %^	59
4	ABCDE	59
5	A!2C#	59
6	Achin Jain	64
7	Achin Jain M. Tech IS student	82
8	Achin Jain M. Tech IS student studying in AIACTR Geeta colony Delhi	120
9	Achin Jain M. Tech IS student studying in AIACTR Geeta colony Delhi has done B. Tech from MSIT Delhi in year 2007	164

impact of buffer overflow attack is much severe than iframe injection attack, and to prevent from such conditions, input parameter length should be monitored efficiently and proper mechanisms need to be employed on Web server to defend against Web attacks.

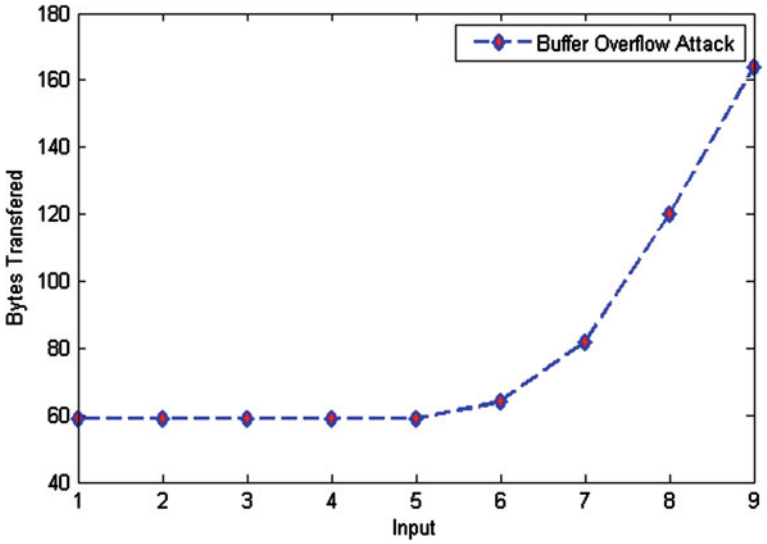

Fig. 6 Graph for buffer overflow attack

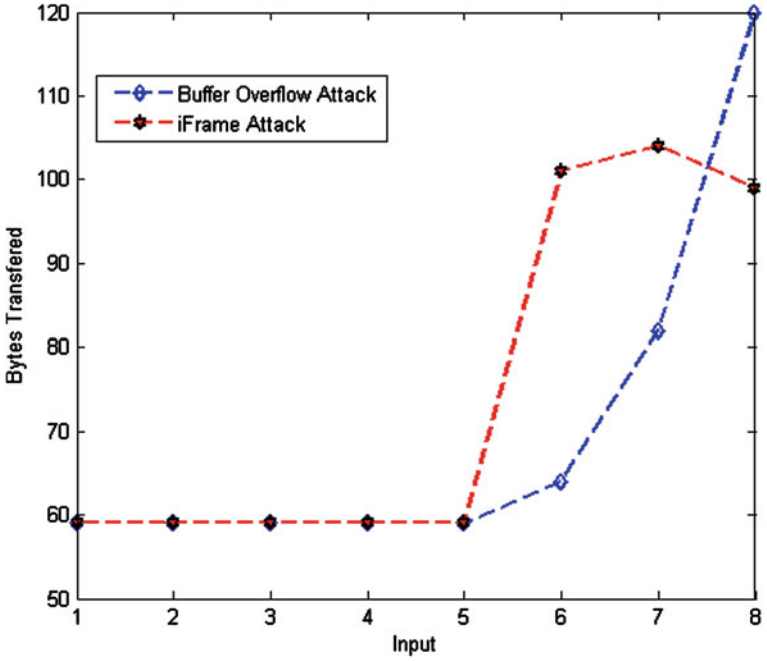

Fig. 7 Comparative analysis of buffer overflow and iframe injection attack

6 Conclusion

In this paper, we have analyzed two Web-based attacks using Web server log files through experimental work. Analysis of log file tells us that while serving normal request, the referrer field contains the host site URL, whereas in case of malicious request, this field is blank, which means that request is coming from other host than legitimate Website. For *iframe* injection and buffer overflow attack, we used "bytes" transferred field for the analysis purpose. After carrying out comparison between *iframe* and buffer overflow attack, we found out that the effect of buffer overflow attack is directly proportional to the length of the malicious input.

References

1. CERT. Advisory CA-2000-02: Malicious HTML tags embedded in client Web requests. Accessed from http://www.cert.org/advisories/CA-2000-02.html (2000)
2. Endler, D.: The evolution of cross site scripting attacks. Technical report, iDEFENSE Labs, (2002)
3. Berinato, S.: Software vulnerability disclosure: The chilling effect. Accessed from http://www.csoonline.com/article/221113/software-vulnerability-disclosure-the-chilling-effect (2007)
4. Aucsmith, D.: Creating and maintaining software that resists malicious attack. http://www.gtisc.gatech.edu/bioaucsmith.html. Accessed on Sept 2004. Distinguished Lecture Series (2004)
5. T. O. Foundation: Top ten most critical Web application vulnerabilities 2005. Accessed from http://www.owasp.org/documentation/topten.html (2005)
6. Singh, N, Singh, K, Raw, R.S.: Analysis of detection and prevention of various SQL injection attacks on Web applications. IJAIS **2**(7), (2012)
7. Cross-Site Request Forgery: http://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF). Accessed on Nov 2011
8. OWASP Top 10 Application Security Risks: http://www.owasp.org/index.php/Top_10_2010-Main. Accessed on Nov 2011
9. Open Source Vulnerability Database (OSVDB): http://osvdb.org. Accessed on Nov 2011
10. Common Vulnerabilities and Exposures (CVE): http://cve.mitre.org. Accessed on Nov 2011
11. Joshila Grace, L.K., Maheswari, V., Nagamalai, D.: Analysis of Weblogs and Web user in Web mining. Int. J. Netw. Secur. Appl. (IJNSA) **3**(1), (2011)
12. Pamnani, R., Chawan, P.: Web Usage Mining: A Research Area in Web Mining. Department of Computer Technology, VJTI University, Mumbai (2010)
13. Kuperman, B.A., Brodley, C.E., Ozdoganoglu, H., Vijaykumar, T.N., Jalote, A.: Detecting and prevention of stack buffer overflow attacks. Commun. ACM **48**(11), 50–56 (2005)