International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

DETECTING AND PREVENTING WEB ATTACKS BY FILTERS

Ms. Ritu Royal[1]
Lecturer
GJIMT, Mohali

Dr. Pardeep Singh Walia[2]
Associate Professor
Govt. College For Girls, Chandigarh

ABSTRACT

The web is a vast and powerful attack surface that attackers can leverage to accomplish their goals of data and financial theft. Due to the positive economics available to attackers the level of sophistication and complexity they can employ is constantly rising. Attackers have been increasingly using the web and client side attacks in order to steal information from targets. The Web is playing a very important role in our lives, and is becoming an essential element of the computing infrastructure. With such a glory come the attacks–theWeb has become criminals' preferred targets.Web-based vulnerabilities now outnumber traditional computer security concerns. Although various security solutions have been proposed to address the problems on the Web, few have addressed the root causes of why web applications are so vulnerable to these many attacks. This paper will introduce and address web based attacks from attack to detection. Web attacks can be prevented using the filters developed to work on the server. Paper presents an effective detection method for the Web based attacks by different filters.

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

Keywords: Access control, Filters, Web Security

1. Introduction

Attacks on the Web are quite unique, compared to the attacks on the traditional computer systems and networks. The most common structure for web applications is three-tiered: presentation, application, and storage. The web browser belongs to the first tier, presentation. The web server, using technologies like PHP, ASP, ASP.NET, etc., is the middle tier, which controls the application logic. The database is in the storage tier. Therefore, a typical web application consists of three major components: contents (static and dynamic, such as, PHP, Javascript code) for the presentation tier, code for the application tier, and interactions with the database.

Web Attacks:

Attacks are the techniques that attackers use to exploit the vulnerabilities in applications. Attacks are often confused with vulnerabilities, to be sure that the attack you are describing is something that an attacker would do, rather than a weakness in an application. Web applications are commonly vulnerable to them, the major cause of attacks for web based applications in 2011 as estimated by OWASP. They are dangerous because they will frequently allow attackers to completely take over the software, steal data, or prevent the software from working at all. According to CWE/SANS ID-CWE-89 – 'SQL Injection' Ranked-1st and Score (93.8) - delivers the knockout punch of security weaknesses in 2011.

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

Every week hundreds of vulnerabilities are being reported in these web applications, and are being actively exploited. The number of attempted attacks every day for some of the large web hosting forms range from hundreds of thousands to even millions. The most common web application attacks are:

- SQL Injection
- Code Injection
- Remote code-Indclusion
- Cross-Site Scripting(CSS)

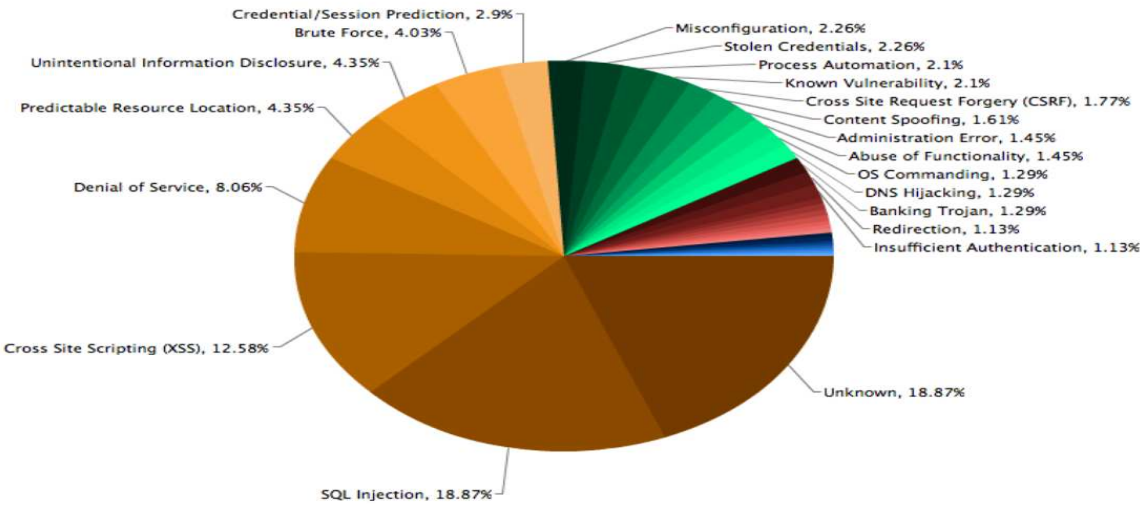


Figure 1. Web Hacking Incident Database Report for 2011

Web Attacker:

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

A web attacker is a malicious principal who owns one or more machines on the network. In order to study the security of browsers when rendering malicious content, we assume that the browser gets and renders content from the attacker's web site.

"Every year the Web security community produces a stunning amount of new hacking techniques published in various white papers, blog posts, magazine articles, mailing list emails, etc. Within the thousands of pages are the latest ways to attack websites, Web browsers, Web proxies, and so on."

2. Related Work

Reis et al (2009) describe sandboxing as the process by which different scripts run by websites such as JavaScript, Java and flash, are executed in a virtual environment without being able to access the system or the hard drive directly. The fact that a number of sandboxes are put into one another, makes it harder for scripts to access the system and run a malicious code. According to Barth et al (2009) sandboxes themselves do not communicate with each other, but interact with the browser through messages, limiting the damage that can be caused by an attacker. They analyse two techniques for communication between sandboxes, which ensure higher levels of authenticity. Firstly, fragment identifier messaging which enables confidential messaging between sandboxes and secondly postMessage, which ensures authenticity, but lacks confidentiality (Barth et al, 2009). According to Jackson et al (2009) another security feature which protects sensitive information such as cookies, passwords and history, is that in modern browsers scripts from websites can access browser content only if they have the same origin. This reduces the risk of theft of personal information, and is particularly applicable for popular websites with a lot of advertising, as adverts displayed on them actually connect the

user with potentially malicious third party servers (Barth et al, 2009). Additionally, Jackson et al (2009) argue that web browsers today incorporate a number of security mechanisms to protect users from false identification of malicious websites, also known as DNS Rebinding attacks.

3. Research Methodology

Research is the pursuit of truth with the help of study, observation, comparison and experiment. In research methodology we not only talk of the research methods but also consider the logic behind the methods we use in the context of our research study and explain why we are using a particular method or technique and why we are not using others so that research results are capable of being evaluated either by the researcher himself or by others.

Research Problem: Defining the research problem is an important step in a research. The research problem of this proposal is:

A) To discover potential Web Attacks, their root causes and associated risks.

B) To identify vulnerable locations in a web application that may lead to web attacks.

4. Web Based Attacks

Web based attacks are considered by security experts to be the greatest and oftentimes the least understood of all risks related to confidentiality, availability, and integrity. The purpose of a web based attack is significantly different then other attacks; in most traditional penetration

testing exercises a network or host is the target of attack. Application vulnerabilities could provide the means for malicious end users to breach a system's protection mechanisms typically to take advantage or gain access to private information or system resources.

There are some fundamental categories of web application attacks:

- Spoofing: is the act of mimicking another user or process to perform a task or retrieve information that would normally not be allowed. An attacker could use a crafted HTTP request containing the session id information from another user and retrieve the targeted users account information.

- Repudiation: Aggregating and correlating logs from multiple sources (web application, middleware, and database) can prevent repudiation attacks.

- Information Disclosure: Information disclosure is one of the biggest threats to large organizations who maintain private information about their customer base.

- Denial of Service: attacks are likely the most well known of all application attacks, often generated by malicious users, competitors or script kiddies.

- Elevation of Privileges: Escalation of privileges requires a malicious user to either already possess or gain through unlawful methods authorization privileges of a regular user.

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

Web applications, on average, are probed or attacked about 27 times per hour or about once every two minutes. At the apex of an attack, web applications experience nearly 25,000 attacks per hour or 7 per second.

Four dominant attack types comprise the vast majority of attacks targeting web applications: SQL injection, Directory Traversal, Cross-Site Scripting, and Remote File Inclusion.

According to Cisco Systems report, In March 2011 with a series of GIF injection attacks targeted at popular Pakistani news sites. The second largest attack in 1Q11 involved website compromises designed to deliver the Hiloti trojan. This particular wave of attacks, breaking in January 2011 before resuming in February, is part of an ongoing series. Though the Lizamoon series of SQL injection attacks were highly publicized in March 2011, both the actual numbers of compromised websites and the live encounter rates were far fewer than had been reported. In reality, only a few thousand websites were actually compromised and live encounters represented only 0.15% of all Web malware encountered for the quarter.

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

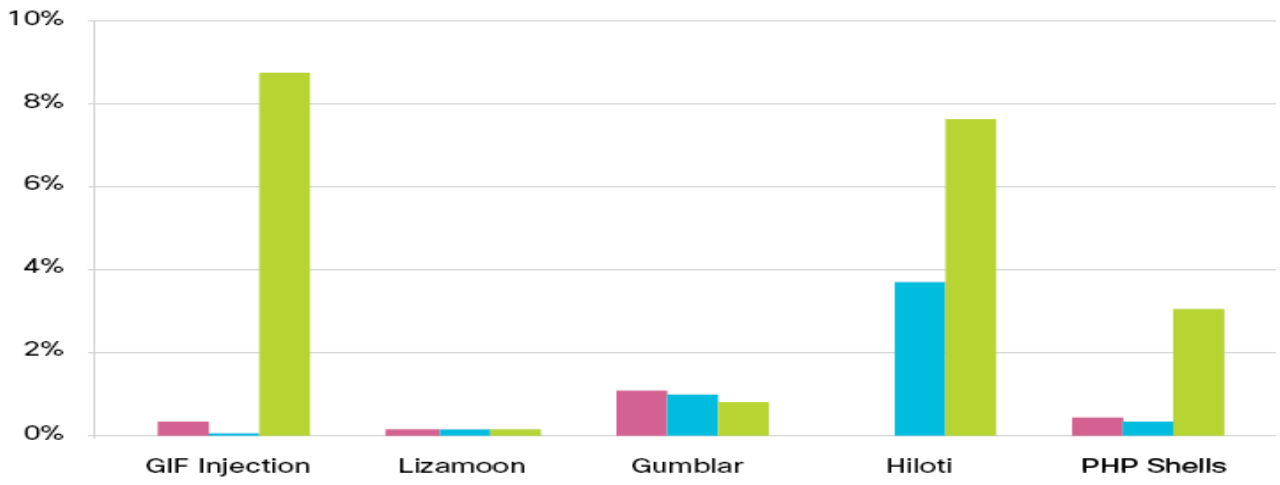


Figure 2. High Profile Web Attacks Report by Cisco Systems, 2011

5. Detecting Attacks in a Network

Web applications are running on the OSI [OSI, 1994] layer 7 the application layer. To detect attacks against web applications, the detection mechanisms have to be application layer aware and see the relevant traffic. Attacks can be detected at different zones and devices in the network infrastructure. Each place has a different view of the traffic and has its advantages and disadvantages. We are now going to explore each of these places in the network.

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

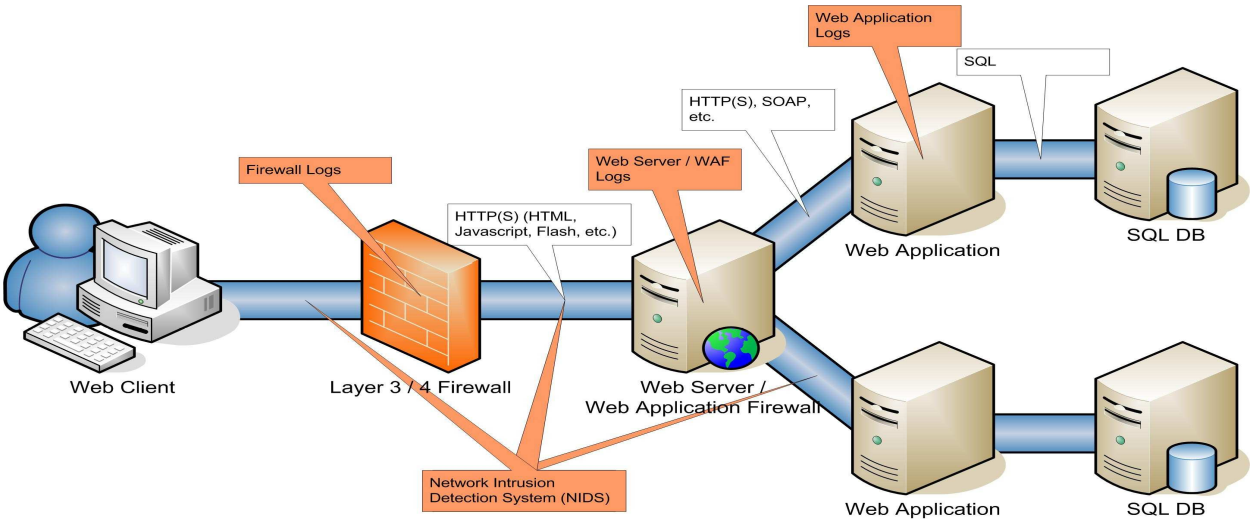


Figure 3. Detecting Attacks in a Network

Firewalls can detect anomalies in the protocols they are aware of like fragmented IP traffic, but they are generally not the best place to detect attacks on the application layer. Firewall log files usually do not contain application layer data like HTTP data, only layer 3 and 4 information, so they are not very helpful in detecting what is going on higher layers.

- Application layer firewall / web application firewall (WAF)

Web application firewalls are designed to work on the OSI layer 7 (the application layer). They are fully aware of application layer protocols such as HTTP(S) and SOAP and can analyze those requests in great detail. Compared to a layer 3/4 firewall, rules can be defined to allow/disallow certain HTTP requests like POST, PUSH, OPTIONS, etc., set limits in file

transfer size or URL parameter argument length. WAF log files contain as much information as those from a web server plus the policy decisions of the filter rules (e.g. HTTP request blocked; file transfer size limit reached, etc.). A WAF provides a wealth of information for filtering and detection purposes and is thus a good place for the detection of attacks.

- Web server

The web server is the end device of an HTTP request. Standard web servers like Apache and IIS are logging by default in the Common Log Format (CLF) specification. Web server logs do not contain any data sent in the HTTP header, like POST parameters. The HTTP header can contain valuable data, as most forms and their parameters are submitted by POST requests. This comes as a big deficiency for web server log files.

- Web application

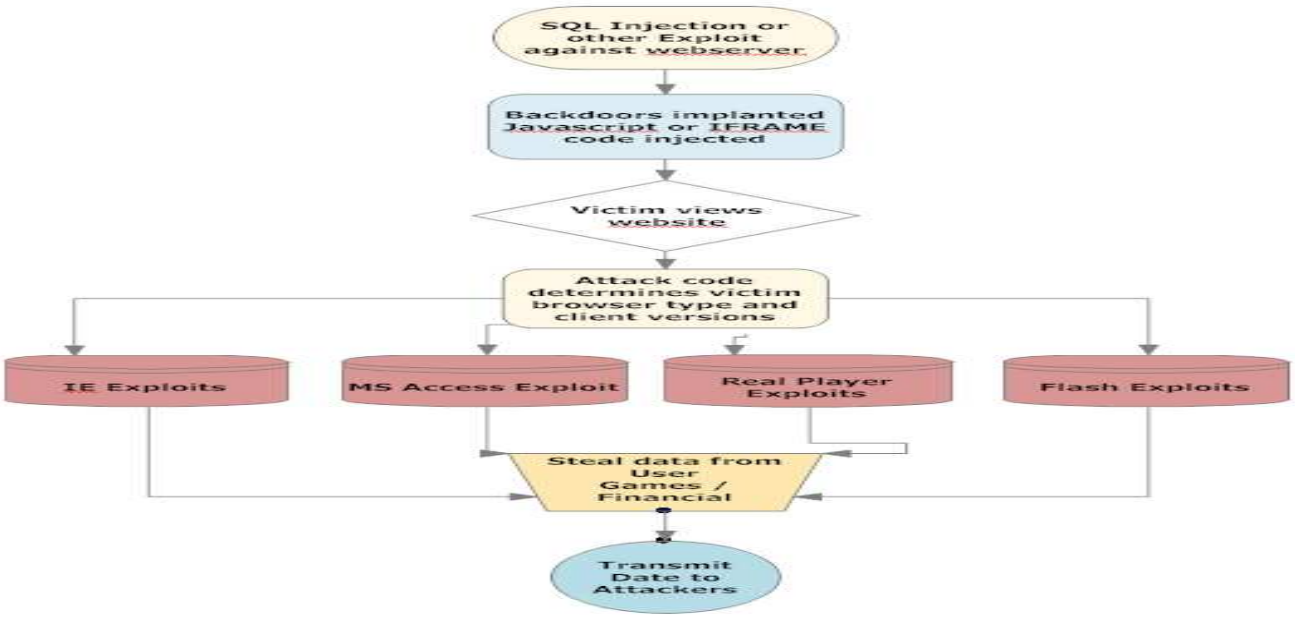
A web application consists of a framework (PHP, ASP, J2EE, etc.) which implements the business logic. It is considered to be best practice to perform input/output validation in this tier. A strong input validation policy will detect malformed and malicious input and can log security related information to a log file. The application has access to the full user trail each step a user takes (logging in, making a transfer, logging out, etc.). A comprehensive logging at the application tier enables the detection of misuse and fraud and allows a full reconstruction of a user's steps.

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012



Attack Flowchart

6. Prevention Methods

Most of the vulnerabilities appear to be caused by the mistakes in the programs, but, when we look deeper and think about why the developers make such mistakes, we realize that the real problem is the underlying access control architecture: because of the inadequacy of the access control support from the underlying architecture, developers are forced to implement additional access control in their programs.

Web Filters:

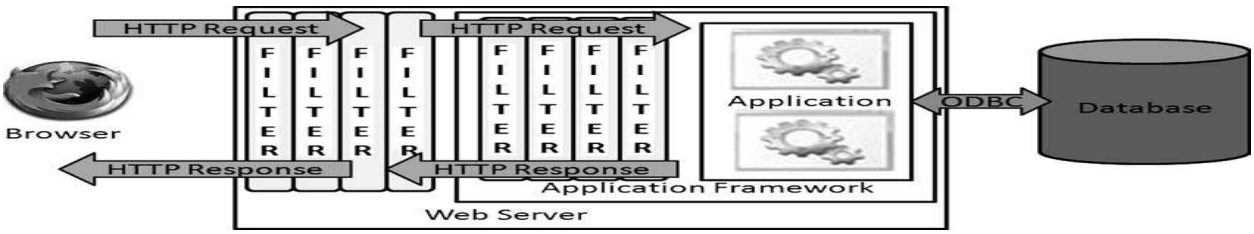
International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

A Web filter is a program that can screen an incoming Web page to determine whether some or all of it should not be displayed to the user. The filter checks the origin or content of a Web page against a set of rules provided by company or person who has installed the Web filter. A Web filter allows an enterprise or individual user to block out pages from Web sites that are likely to include objectionable advertising, pornographic content, spyware, viruses, and other objectionable content. Vendors of Web filters claim that their products will reduce recreational Internet surfing among employees and secure networks from Web-based threats.



Web Server and Application Filters

We are working on developing such a system support:

- Server-side access control:

The server side, access control is primarily based on sessions. When a user logs into a web application, the server creates a dedicated session for this user, separating him/her from the other users. Sessions are implemented using session cookies; as long as a request carries a session cookie, it will be given all the privileges associated with that session. Namely, within

each session, all requests are given the same privileges, regardless of whether they are initiated by first-party or third-party contents, from client-side or server-side extensions, or from another origin. We refer to this access control as the “same-session” policy.

- Parameterized Statements in PHP

PHP has a number of frameworks that you can use to access a database. The mysqli package for accessing MySQL databases, the PEAR::MDB2 package (which superseded the popular PEAR::DB package), and the new PHP Data Objects (PDO) framework, all of which provide facilities for using parameterized statements to secure web network.

- Encryption Filters, Evading Input Filters

Using a filter in between the Web application server and database server to filter out the abnormal or bad SQL injection queries. Web applications frequently employ input filters that are designed to defend against common attacks, including SQL injection. These filters may exist within the application’s own code, in the form of custom input validation, or may be implemented outside the application, in the form of Web application firewalls (WAFs) or intrusion prevention systems (IPSs).

- Nesting Stripped Expressions

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

Some sanitizing filters strip certain characters or expressions from user input, and then process the remaining data in the usual way. If an expression that is being stripped contains two or more characters, and the filter is not applied recursively, you can normally defeat the filter by nesting the banned expression inside itself.

- Using URL Encoding

URL encoding is a versatile technique that you can use to defeat many kinds of input filters. In its most basic form, this involves replacing problematic characters with their ASCII code in hexadecimal form, preceded by the % character. A vulnerability discovered in 2007 in the PHP-Nuke application employed a filter which blocked both whitespace and the inlinecomment sequence.

7. Conclusions

Today, more and more applications are being hosted on the Internet. As organizations have moved their applications to this environment, the threats have only increased. To avoid becoming a victim, understand what you face, how attacks are carried out, and how you can employ proper defensive measures. A little research and understanding go a long way. It is time to think about whether we can design a better and backward-compatible access control system, instead of developing fixes to patch the existing one in order to defeat certain specific attacks. The web technology is still evolving, so a good design should not only be able to satisfy today's needs, it should also be extensible to satisfy the unknown protection needs that will inevitably come up during the technology evolution. In this paper, we have summarized our pursuit in building a better access control system for the Web. We strongly believe that the access control

systems in the current Web infrastructure is fundamentally inadequate to satisfy the protection needs of today's Web, and they have, directly and indirectly, contributed to the dire situation in web applications.

8. References

[1] Wenliang Du, Xi Tan, Tongbo Luo, Karthick Jayaraman, and Zutao Zhu, "Re-designing the Web's Access Control System (Extended Abstract)?," Data and Application Security and Privacy XXV, LNCS 6818, page 4-11,2011.

[2] A. Yip, X. Wang, N. Zeldovich, and M. F. Kaashoek. Improving application security with data flow assertions. In Proceedings of the 22nd ACM Symposium on Operating Systems Principles, Big Sky, MT, October 11-14 2009.

[3] "Top 25 most critical web application vulnerabilities", OWASP Foundation, http://www.owasp.org/documentation/topten.html, 2011.

[4] Justin Crist, "Web Based Attacks", SANS Institute, As part of the Information Security Reading Room, http://www.sans.org/reading_room/whitepapers/application/web-based-attacks_2053

[5] Barth, A., et al, (2009) "Secure Frame Communication in Browsers", USENIX Security Conference 2008, www.usenix.org/events/sec08/tech/full_papers/barth/barth.pdf

International Journal of Enterprise Computing and Business Systems

ISSN (Online) : 2230-8849

http://www.ijecbs.com

Vol. 2 Issue 1 January 2012

[6] BARTH, A., JACKSON, C., MITCHELL, J. 2009. Securing Frame Communication in Browsers. Communications of the ACM, 52 (6), pp. 83-91.

[7] JACKSON, C., BARTH, A., BORTZ, A., SHAO, W., BONEH, D. 2009. Protecting Browsers from DNS Rebinding Attacks. ACM Transactions on the Web, 3 (1), pp. 2:2-28.