A Review on Web Application Security

Tanvi Dhingra
Department of Computer Science and Engineering
PEC University of Technology
Chandigarh, India

***Abstract*— In this scientific era, web is an important part of our lives because it provides anytime, anywhere access to information and services. These services provided by web are called web applications. Everyday millions of users connect to these web applications for various activities like for business transactions, financial transactions, social communication and much more. Despite of such advantages, security is one of the major issue of web applications. Web applications have been split into browser side and server side components. In this paper some browser side and server side attacks have been discussed and some techniques like web application scanners and firewalls have also been discussed to prevent these attacks.**

***Keywords*— Web Application Security, Application Logic Vulnerability, Web Application Scanners, Firewalls**.

I. Introduction

Web Application is basically an application program that is stored on the server and delivered over the internet with the help of an interface called browser. Since the web applications completely rely on internet (which is highly unsafe from security point of view) for there working, so security of web applications is one of the major issue these days. Moreover ensuring web application security is a challenging task because web applications are associated with different layers like client layer, application layer and data layer. So security mechanism should be designed in such a way that it ensures security at all the layers of web application.

A. Structure of web application

Structure of web applications include following 3 layers:

- Client layer: Client is basically the user's web browser. The browser creates HTTP requests for specific URLs that maps its request to second layer i.e. application layer which maps the user's request to the corresponding resource on the web server.
- Application layer: It basically acts as an interface between client layer and data layer. It includes policies and rules in the form of code and expression which determines the workflow that is how data will pass from one participant to another.
- Data layer: It includes databases. It process the query sent by application layer and generate response to the query. Maintenance of databases is the major issue at this layer.

B. Working of Web Applications

Web application consists of a code on both client side and server side. Coding at client side can takes place in any language like DHTML (HTML, XHTML, HTML x.0), JavaScript, Java (Applets), VBScript, XML/XSL, CSS etc. which gets executed at the browser. Authentication and Authorization feature is usually associated with web applications. Consider the example of Gmail web application. When a user wants to access its Gmail account an authentication page is displayed which consists of fields like Login ID and password. This coding is basically in HTML with javascript usually or another scripting language embedded in it. When a user enters incorrect password or login id then a message gets displayed (invalid login id or password), this message is displayed because of the scripting language running at the client side. Now when the user enters the authentication information then this information via https goes to the server. Various coding languages can be used for coding at the server side (e.g asp .NET, PHP, JSP etc). For example that coding at the server side takes place in PHP. Server side languages communicate with the database (here MySQL) for storing, retrieving and manipulating data. PHP files are basically connection files that makes the HTML pages dynamic. Users request i.e request coming from clients browser is collected in a PHP file, processed and then sent to the MySQL database for accessibility to various resources.

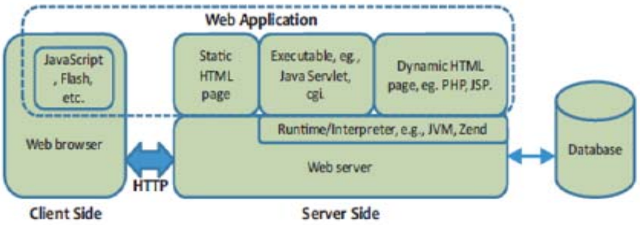

Figure.1:Overview of Web Application[5]

II. Security Breaches On Web Applications

A. Cross Site Scripting (XSS)

XSS is one of the most prevalent attack in web applications. Web Applications basically consists of HTML pages. So, the main focus of the attacker in this attack is to inject malicious code into the HTML pages. When user access the web application, then this HTML page with malicious script opens in the users web browser. This script is then executed in the users web browser leading to cross site scripting attack (XSS). With this attack the attacker can redirect the users browser, inject virus and can hijack the users web browser.

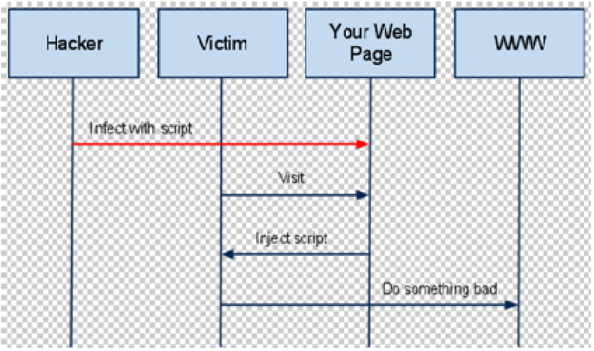

Figure2:Cross Site Scripting[1]

B. *SQL Injection*

It is the second most prevalent attack in web applications. This attack is possible when user input is improperly validated. In this attack the attacker illude the database by injecting SQL commands that gets executed by the database. This attack gives an attacker access to database which can allow the attacker to access secret data, modify contents of website, shut down the MySQL server etc. This is how this attack takes place:

Consider for example that the target website uses the following unsecured script:

```
<?php
$ sql= "SELECT_FROMusersWHERE
username='" .
$-POST[ ' username ' ] . " 'ANDpassword='" .
$POST [ ' password ' ] . " ' " ;
respons e = mysql query( $ sql ) ;
?>
```

After inserting malicious script, Let's take a simple username (mostly admin or administrator) and abc as password.

This is how the MySQL Query looks now:

```
SELECT*FROMusersWHEREuser=' admin '
ANDpassword=' ' OR ' a ' = ' a '
```

'a' = 'a is a true value, just like 1 = 1 i.e

```
SELECT*FROMusersWHEREuser='admin'
AND TRUE
```

Means attacker is logged in as an administrator by manipulating the SQL query.

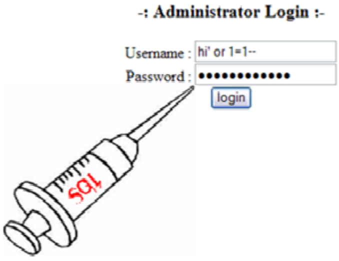

Fig.3:SQL Injection[1]

C.) Cookie Poisoning

A cookie is a piece of text that a web server can store on users hard disk. Cookies allow a Web site to store information on a user's machine and later retrieve it. Cookies consists of name-value pairs[10]. In this attack, the attacker modifies the content of the cookies to gain unauthorized access for purposes like identity theft, opening new accounts or gain access to existing accounts.

D). Password Guessing Attack

Authentication is one of the important step in accessing the web applications. In password guessing attacks, attacker gains access of a system or application with a guessed password. Nowadays passwords are easy to crack due to the availability of large number of automated tool like cain and abel, john the ripper, Hashcat, Hydra etc.

Password guessing attacks takes place in 2 ways:

- Brute force attack
- Dictionary attack

E) Remote File Inclusion

Inclusion means adding something. File inclusion basically means adding a common code into different files that are then referenced by main application module. In this attack, attacker gains full access of a website. This attack is mainly prominent with PHP i.e. it make use of PHP include() function. This is how the attack takes place, when the user sends its request to the web application then it gets collected by server side scripting language (PHP here) in the include() function. Here when the input is not properly validated then web applications might be fooled into including remote files with malicious code. This attack takes place at both the client side and server side.

Let us consider the following code:

```
$incfile = $-REQUEST[ " file " ] ;
include ( $incfile . " . php" ) ;
```

The first line of code extracts the value of the file parameter from the HTTP request. The second line of code dynamically sets the file name to be included using the extracted value. If the web application does not properly sanitize the value of the request parameter then this code can be exploited.

Some other vulnerabilities existing in web applications are:

- Buffer Overows
- Denial of service
- Path manipulation
- Session Hijacking

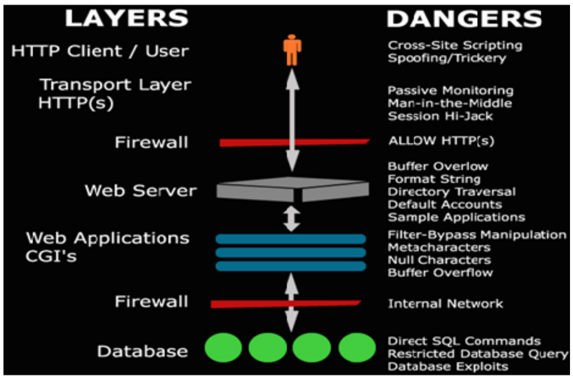

Figure4:Vulnerabilities existing at various layers in web application framework[9]

III. Security Mechanisms

Security is the major concern of Web Application Systems. Security goals like data confidentiality, authenticity, integrity, availability and non-repudiation have to be satisfied for the efficient working of these systems. Web applications are built from coding languages, so care must be taken during coding phase of a web application i.e. use of secure coding languages should be enhanced. Various security measures should be applied during the development phase of website i.e they should be checked for every possible loophole during their designing phase. Various security technologies have also been proposed to check for various security vulnerabilites in a website. Some of these security technologies are web application scanners, web application firewalls, static source code analyzer, fuzzing tools etc.

A. *Web Application Firewalls*

According to the Web Application Security Consortium, a web application firewall (WAF) is an intermediary device, sitting between a web-client and a web server, analyzing OSI Layer-7 messages for violations in the programmed security policy[3]. The main function of the web application firewalls is to protect the web server from various attacks like SQL Injection, cross site scripting, buffer overflows etc. Web application firewalls uses the following two approaches to detect an attack:

- Signature Based: In this approach, firewall maintains an attack signature file(e.g cross site scripting, SQL injection etc.).Whenever the signature of users input file matches with stored signature, then the firewall will block that attack from coming in.
- Abnormal behavior based: In this approach the firewall will check for abnormal traffic pattern behavior. Whenever the firewalls finds an abnormal behavior then it will block that traffic or attack.

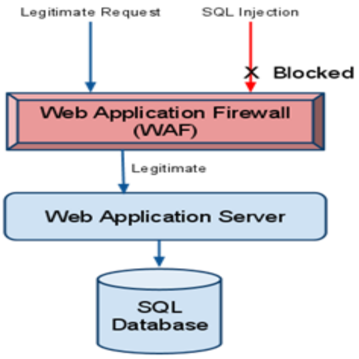

Fig.5:Working of web application firewalls[7]

B. *Web application scanners*

A web application security scanner is a program which communicates with a web application through the web front-end in order to identify potential security vulnerabilities in the web application and architectural weaknesses[4]. Scanners are basically used for testing of vulnerabilities in a website. Wireshark, Nmap, Nessus, accunetix are some of the examples of web application scanner tools. In order to test web applications, a scanner must support following features:

- Protocols Support: Scanner should support various protocols like HTTP 1.1, HTTP 1.0, SSL, HTTP Keepalive, HTTP Compression etc. in order to perform various functions like handling sensitive information, reusability of existing connection for multiple requests etc.
- Authentication: Since Web Applications are usually associated with authentication and authorization so, scanner should support various authentication schemes like basic, digest, HTML form based, single sign on etc.
- Crawling: It is a process used to describe the action taken by a program as it browses from page to page on a website. Crawler starts from the starting page and traverse provided links until it reaches the destination. Crawling is required to ensure that scanner is aware of all the pages associated with a website.
- Session Management: It is required basically to perform web crawling. Session Management helps the scanner to identify various web elements like cookies, forms, links etc.
- Parsing: Scanner should be able to parse various programming languages like HTML, javascript, VBscript, XML etc in order to extract information from the web content.
- Reporting: After performing all the above steps the scanner should prepare a report of the collected information so that the user can understand various loopholes in the required application. Report should be clear enough so that the vulnerabilities can be mitigated from the web application.

1. Working of Web application scanners

- Vulnerability database: Scanner maintains a database for various known vulnerabilities. When the scanning process gets started, then user input is compared with the database to determine the known vulnerability.
- Knowledge base: In knowledge base complete information like type of protocol, proxy, scripting language etc. is maintained. This information is used by the scanner for deriving the result.
- Results: After scanning the targets, the scanning engine will generate a result report which gives the complete description of the attack.

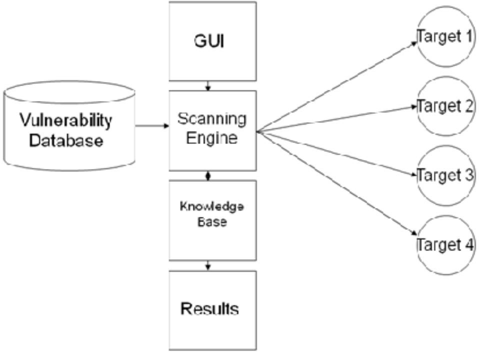

Fig.5:Working of Web Application Scanner[8]

IV. CONCLUSION

This paper provides an exhaustive survey of architecture, various security attacks existing in the web applications and the impact of these attacks on various layers of web application. Various security technologies like web application firewalls and web application scanners have been discussed i.e. how these technologies work to prevent and detect various security attacks. Although these security technologies can greatly prevent from various vulnerabilities but still more focus should be given during the development of a web applications, so that various vulnerabilities can be prevented during the initial development of website.

REFERENCES

[1] Abhishek Kumar Baranwal, Approaches to detect SQL injection and XSS in web applications;April 2012

[2] Issac Museong Kim, Using Web Application Firewall to detect and block common web application attacks;November17,2011

[3] Web Application Security,February 2008

[4] "Web Application Security Scanner Evaluation Criteria" Web Application Security Consortium, 2009.

[5] Xiaowei Li and Yuan Xue, A Survey on Web Application Security

[6] Elizabeth Fong and Vadim Okun, Web Application Scanners: Definitions and Functions

[7] http://www.acunetix.com/blog/news/ barracuda-networks-breached/

[8] https://users.cs.jmu.edu/tjadenbc/Bootcamp/ 7-VulnScanning.ppt/

[9] http://www.blackhat.com/ presentations/bh-europe-01/ jeremiah-grossman/ bheurope-01-grossman.ppt

[10] http://www.imperva.com/resources/glossary/cookie_poisoning.html