Survey on Web Application Vulnerabilities

Yogesh Kumari

Assistant Professor, Department of Computer Science, Shivaji College, Delhi University, New Delhi, India

---

Abstract - *As in the last few years discovery of World Wide Web has grown rapidly. This has made web applications an important aspect to provide end users all the necessary features to access server functionality. Today, web application is one of the most widely used platforms to deliver information and services over Internet. Many World Wide Web applications are used in many security critical environments including medical, financial, e-commerce, military systems. A web application is the one that provides server functionality to its end users through a number of web pages. These web pages contain code that executes dynamically resulting client's queries.*

Although many techniques have been developed to fortify web applications and mitigate attacks towards web applications. Web applications are prone to attacks like browser hijacking, session riding, and cookie theft. This paper explains the working of web applications and the existing security vulnerabilities possible in web applications.

***Key Words*: Web application vulnerabilities, SQL injection, cross-side scripting**

1. INTRODUCTION

Today Internet has turned out to be an effective way of communication all over the world. Millions of businesses use Internet to exchange information with target market. It is also true that effective marketing is only possible when business is able to capture all the necessary information and store it in an effective way so that later on, information can be processed with the accurate result to the end user. Web application utilizes web browsers and various web technologies to carry out task over the Internet. Web application includes shopping carts, online forms, spreadsheets, file conversion, photo and video editing, word processing, file processing and more.

Web application can be accessed with the same version without any compatibility issues. Web applications run on multiple platforms regarding of device or the operating system.

How Web Application Works?

Web application uses client side script to present information to end users, using JavaScript and HTML. While server side scripts are used to handle storage and retrieval of information using PHP and ASP. Through this only, end users are capable to interact with any website which includes online forms, content management systems, shopping carts. Every web application requires a web server, application server and database. Web server is required to manage requests from clients. Application server is required to perform the query submitted by end user and database is used to store the information.

Whenever a web server receives a request from an end user for any web page, the web server sends the page to the requesting browser and it passes the page to application server. The application server reads code on the page, performed the requested task like querying the database or processing any data and generates the accurate results. The result is passes back to web server by application server as static HTML page.

At first, user sends the request to web server through application's user interface or web browser. The same request is forwarded to web application server through web browser. Application server performs the task by requesting database server for any query, as soon as the query is processed, application server gets the results. Further, application server sends result to web server and web server respond back to the client with the results.

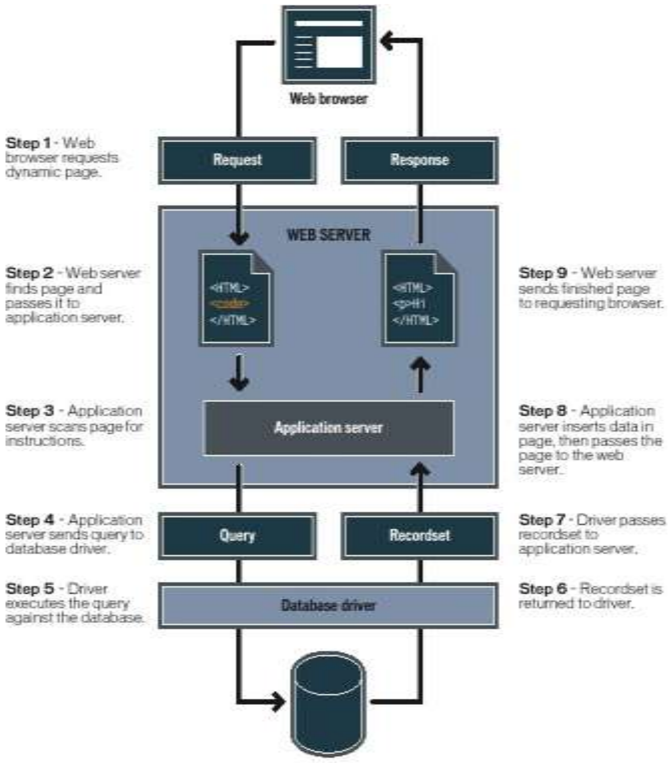

Web browser
Step 1 - Web browser requests dynamic page.
Request
Response
WEB SERVER
Step 2 - Web server finds page and passes it to application server.
<HTML>
<code>
</HTML>
<HTML>
<p>Hi
</HTML>
Step 9 - Web server sends finished page to requesting browser.
Step 3 - Application server scans page for instructions.
Application server
Step 8 - Application server inserts data in page, then passes the page to the web server.
Step 4 - Application server sends query to database driver.
Query
Recordset
Step 7 - Driver passes recordset to application server.
Step 5 - Driver executes the query against the database.
Database driver
Step 6 - Recordset is returned to driver.

Figure1: Working of Web Application

2. BACKGROUND

Over the last few years, web applications have turned out be a favorite subject where attacks tends to attack. Cross-site scripting, cookie poisoning came in picture. For instance, when a website is vulnerable to cross-site scripting, it means user cookie can be theft. And if cookie has been compromised, attacker can perform session hijacking and can have access onto user's account. Thus, it is very important to understand what are the ways through which attacker can attack any website. What could be the possible vulnerabilities in any web application?

3. CLASSIFICATION OF WEB APPLICATION VULNERABILITIES

There are various types of vulnerabilities that have been presented as a thread to any web application. The following classes of attacks are as follows:

3.1. Authentication

The Authentication talks about the attack that targets the method of validating the identity of a user, any application or any service. Authentication can be performed by any of the mechanism like something you have, something you know or something you are.

1. Brute Force

Brute force attack is a hit and trial method to guess the username or password. In this attacker can generate millions of thousands of incorrect guess. And if the brute force is successful, the attacker is able to access the account.

2. Insufficient Authentication

Insufficient Authentication attack takes place whenever a web application provides access to get sensitive information or any functionality without properly authentication. If a resource is unknown to an attacker, it still remains accessible through a specific URL.

For instance, Most of the websites have been designed with administrative functionality provided in root directory /admin/. Though the directory is not linked to anywhere but it can still be accessed through a standard web browser. If somehow attacker visits this page, entire administrative access of the web application would be gone.

3. Weak Password Recovery Validation

Any website tends to have weak password recovery validation when it permits anyone to illegally obtain, change or recover anyone's password. It happens when any information which is required to validate user's identity can be circumvented. There are various ways through which password recovery systems can be compromised, such as brute force attacks, easily guessed secret questions.

For instance, in many cases, website provides hint to remind users of their password. Some websites request users to provide email address with the combination of the phone number. Such information can be easily gained through online white pages.

3.2 Authorization

Authorization section includes attacks that target web application's method of determining if the user has permissions in order to perform a specific task or request. There are various techniques through which an attacker can fool a web site into increasing their privileges to gain protected information.

1. Session Prediction

Websites are designed in such a way that a user connection is established by providing correct credentials in order to authenticate and track a user. Websites generate unique session ID to identify user session as authenticated. So, session ID is the proof and if this session ID is somehow gained by attacker, fraudulent activates are possible.

2. Insufficient Session Expiration

If any web application provides access via old session credentials or session ID, attacker can gain access. Insufficient session expiration increases web application exposure through which attacker can attack or steal any user's personal information. A stolen session ID can be used to perform any fraudulent transaction. If the expiration time of session is more, more is the chances that attacker will guess the valid session ID. Such expiration can be exploited to view user's activity over Internet.

3. Session Fixation

Session Fixation is another technique of attack in which user's session ID is given an explicit value. A number of techniques are used to fix the session ID value depending on the functionality of target website. Once the value is fixed, attacker wait for legitimate user to login. Once login credentials are given to successful login, the attacker uses predefined session ID value.

4. Insufficient Authorization

When a website permits an end user to access sensitive data that should require increased access control restrictions. Even if a user is authenticated, full access of sensitive content and functionality must not be granted. For instance, many websites store administrative content and functionality in directory /admin and /log. If an attacker gets the access of these directories, the entire information can be compromised.

3.3 Client-Side Attacks

The client side attacks basically focuses on the exploitation of website's users.

1. Content Spoofing

Content Spoofing is an attack technique that permits an attacker to inject malicious code which later can be misrepresented as the legitimate content of any web application. For instance,

```
<HTML>

<FRAMESET COLS=”100, *”>

<FRAME NAME=”pr_menu” SRC=”menu.html”>

<FRAME NAME=”pr_content

SRC=”http://foo.example/pr/01012003.html>

</FRAMSET>

</HTML>
```

In this code, if an attacker altered the URL to http://foo.example/pr?pg=http://attacker.example/spoofed_press_release.html?

```
<HTML>

<FRAMESET COLS=”100, *”>

<FRAME NAME=”pr_menu” SRC=”menu.html”>

<FRAME NAME=”pr_content

SRC=”

http://attacker.example/spoofed_press_release.html”>

</FRAMESET>

</HTML>
```

I. Cross-Site Scripting

Cross-site scripting is an attack technique in which attacker’s supplied executable code is echoed through any web application. Persistent and non persistent attacks are the two cross-site scripting attack. In non-persistent attack, end user is required to visit crafted link laced with malicious code. Once the link is clicked, the code is embedded in the URL and executed within user’s web browser. Whereas, in case of persistent attack, malicious code is submitted to website where it is stored for a period of time.

2. Command Execution

In this section, attacks are covered which are designed to execute remote commands on any web application.

I. Buffer Overflow

Buffer overflow occurs when data written to a memory exceeds the allocated size of the buffer. Once the buffer is overflow, it can cause the software to crash or fault. Buffer overflow also used as denial of service attack when memory is corrupted resulting as software failure.

II. OS Commands

In this type of attack, Operating system commands are used to exploit website through manipulation of application input. For instance,

```
exec("ls -la $dir",$lines,$rc);
```

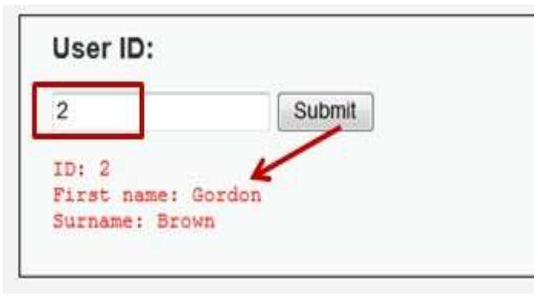
While executing Shell command, appending (;) semicolon can force the web application into executing second command

```
http://example/directory.php?dir=%3Bcat%20/etc/pass
```

Retrieving the content of /etc/passwd file.

III. SQL Injections

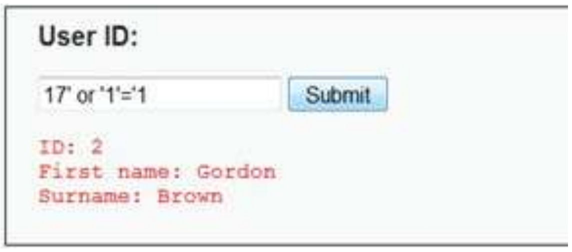
In SQL injection attack, attacker sends the malicious input through application interface and queries are executed in the database giving attacker the control over the database. For instance, in this entering the right query i.e. 2 , the correct result will be shown from the database.



```
ID: 2
First name: Gordon
Surname: Brown
```

Figure2

But in SQL injection attack, the attacker has given wrong query i.e.17 OR with equality symbol, OR is always going to give the answer “1” and ultimately providing access to the database.



```
ID: 2
First name: Gordon
Surname: Brown
```

Figure3

Even an attacker submits a login and password that looks like

Login: ' OR ''=''

Password: ' OR ''=''

Select Username from users where Username='' OR ''='' AND Password= ''OR ''=''

This query compares empty string with an empty string and resulting TRUE results and providing attacker to be logged in.

IV. Path Traversal

The Path Traversal attack forces access to file and directories that reside outside the web document root directory. Any device which exposes HTTP based interface makes it potentially vulnerable to Path Traversal attack.

The most basic path traversal attack uses "…/" character sequence to alter resource location requested in the URL.

Path Traversal against a web server

Attack: http://example/../../../../../some/file

Attack:http://example/..%255c..%255c..%255csome/file

Attack: http://example/..%u2216..%u2216some/file

3. Logical Attacks

The logical attacks focus on the exploitation of web application's logical flow. Few examples of application logics are password recovery, auction bidding, account registration and e-commerce purchases.

I. Denial of Service

Denial of Service attack prevents a website from serving normal user activities. Denial of Service attack is used to target any specific user, database server and the web server. An intruder can lock out any specific user by repeatedly login attempts, use of SQL injections to modify database system to make it unstable and buffer overflow technique to crash or fault the web server.

4. CONCLUSIONS

Web applications have become most common and widely used communication medium for day-to-day activities. An attacker can easily attack any web application if there is any fault in its design, implementation or deployment. There are many ways through which an intruder can break into a system and weaken it. This paper discussed most widely used web application vulnerabilities which intrude web application security. An increasing amount of logic and application code is moving towards client side as a result, it is bringing new security challenges. The attacker is able to gain more knowledge about the application results in comprise of server side application state.

REFERENCES

[1] Web Application Security Consortium: Thread Conclusion version 1.0.0 by webappsec.org.

[2] Classification of Web Application Vulnerabilities, Savita B. Chavan and Dr. B. B. Meshram, V.J.T.I, Mumbai, India V.J.T.I, Mumbai, India

[3] A Survey on Server-side Approaches to Securing Web Applications XIAOWEI LI Vanderbilt University YUAN XUE Vanderbilt University

[4] Web Application Vulnerabilities: A Survey, Vandana Dwivedi, Himanshu Yadav and Anurag Jain RITS, Bhopal (India)

[5] OWASP Top 10 Web Application Vulnerabilities.http:// www.applicure.com/blog/owasp-top-10-2010

[6] A. Klein. "Cross Site Scripting Explained" Technical report, Sanctum Inc., June 2002.

[7] Open Web Application Security Project. The ten most critical Web application security vulnerabilities. http://umn.dl.sourceforge.net/sourceforge/ owasp/OWASPTopTen2004.pdf, 2004.

[8] Z. Su and G. Wassermann. The essence of command injection attacks in Web applications. In Proc. POPL, 2006

[9] Security Code Review- Identifying Web Vulnerabilities by Kiran Maraju.