Web Application Hacking

A web application is an application that is accessed over a network such as Internet or an intranet. The term may also mean a computer software application that is coded in a browser-supported language and reliant on a common web browser to render the application executable. Although the World-Wide Web was initially conceived as a vehicle for delivering documents, it is now being used as a platform for sophisticated interactive applications, displacing the traditional mechanism of installable binaries. Web-based applications offer numerous advantages, such as instant access, automatic upgrades, and opportunities for collaboration on a massive scale. However, creating Web applications requires different approaches than traditional applications and involves the integration of numerous technologies.

A web application is any application that uses a web browser as a client. The application can be as simple as a message board or a guest sign-in book on a website, or as complex as a word processor or a spreadsheet. The 'client' is used in client-server environment to refer to the program the person uses to run the application. A client-server environment is one in which multiple computers share information such as entering information into a database. The 'client' is the application used to enter the information, and the 'server' is the application used to store the information.

Over the past decade or so, the web has been embraced by millions of businesses as an inexpensive channel to communicate and exchange information with prospects and transactions with customers. In particular, the web provides a way for marketers to get to know the people visiting their sites and start communicating with them. One way of doing this is asking web visitors to subscribe to newsletters, to submit an application form when requesting information on products or provide details to customize their browsing experience when next visiting a particular website.

All this data must be somehow captured, stored, processed and transmitted to be used immediately or at a later date. Web applications, in the form of submit fields, enquiry and login forms, shopping carts, and content management systems, are those website widgets that allow this to happen. Web browsers are software applications that allow users to retrieve data and interact with content located on web pages within a website.

Today's websites are a far cry from the static text and graphics showcases of the early and mid-nineties: modern web pages allow personalized dynamic content to be pulled down by users according to individual preferences and settings. Furthermore, web pages may also run client-side scripts that "change" the Internet browser into an interface for such applications as web mail and interactive mapping software (e.g., Yahoo Mail and Google Maps). Most importantly, modern web sites allow the capture, processing, storage and transmission of sensitive customer data (e.g., personal details, credit card numbers, social security information, etc.) for immediate and recurrent use. And, this is done through web applications. Such features as webmail, login pages, support and product request forms, shopping carts and content management systems,

shape modern websites and provide businesses with the means necessary to communicate with prospects and customers. These are all common examples of web applications.

Web applications are, therefore, computer programs allowing website visitors to submit and retrieve data to/from a database over the Internet using their preferred web browser. The data is then presented to the user within their browser as information is generated dynamically (in a specific format, e.g. in HTML using CSS) by the web application through a web server.

There are many programming languages used to develop a web application. Some of them are:

1) HTML – Hyper Text Markup Language
2) PHP – Hypertext Preprocessor
3) ASP.NET – Web application framework developed by Microsoft
4) Java Script

There are a lot of web browsers on which a web application runs. Some of them are:

1) Mozilla Firefox
2) Google Chrome
3) Internet Explorer
4) Aurora Web browser
5) Opera
6) Safari
7) Deepnet Explorer
8) Maxthon
9) RockMelt
10) SeaMonkey

A web application is based on client-server architecture. The client/server model is a computing model that acts as a distributed application which partitions tasks or workloads between the providers of a resource or service, called servers, and service requesters, called clients. A server machine is a host that is running one or more server programs which share their resources with clients. A client does not share any of its resources, but requests a server's content or service function. Clients therefore initiate communication sessions with servers which await incoming requests. Functions such as email exchange, web access and database access are built on the client/server model. Specific types of servers include web servers, ftp servers, application servers, database servers, name servers, mail servers, file servers, print servers, and terminal servers.

A database is a collection of information that is organized so that it can easily be accessed, managed, and updated. The most prevalent approach is the relational database, a tabular database in which data is defined so that it can be reorganized and accessed in a number of different ways. The database data collection with Database Management System is called a database system. A database management system (DBMS) is a software package with computer programs that controls the creation, maintenance, and use of a database. It allows organizations to conveniently develop databases for various applications. A database is an integrated collection of data records,

files, and other objects. A DBMS allows different user application programs to concurrently access the same database. Well known DBMSs include Oracle, IBM DB2, Microsoft SQL Server, Microsoft Access, PostgreSQL, MySQL, and SQLite. A database is not generally portable across different DBMS, but different DBMSs can inter-operate to some degree by using standards like SQL and ODBC together to support a single application.

Figure: Clients-Server

Authentication Bypass:

This form of SQL injection occurs when user input is not filtered for escape characters and is then passed into an SQL statement. This result in the potential manipulation of the statements performed on the database by the end-user of the application. A SQL attack can take a long time because it involves trail n error method.

For example the target website uses this vulnerable, unsecured authorization script:

```
<?php
    $sql = " SELECT * FROM users WHERE username=' " . $_POST['username'] . " ' AND
    password=' " . $POST_['password'] . " ' "
    response = mysql_query($sql);
?>
```

As you can see, the user's input is not getting checked or filtered.
This is how the MySQL Query looks now:

```
SELECT * FROM users WHERE user=" AND password=";
```

To bypass the authentication we use a simple string and that is:

0'or'0'='0

Now the MySQL query becomes:

SELECT * FROM users WHERE user='0’or’0’=’0' AND password='0’or’0’=’0';

When a user tries to authenticate with the some credentials he will be authenticated only if the username and password matches which are stored in the database. In the above string we have two conditions, i.e., ‘=’ and ‘0’=’0’. When we insert the above string in place username and password first ‘=’ will be processed for the authentication and if it is not correct it goes to next condition that is ‘0’=’0’ and that is a universal truth and database accepts it and authenticate a user.