EH ASSIGNMENT 6

Q.1 What is Session Hijacking Explain with Techniques?

ANS. Session hijacking is a type of cyberattack in which an attacker gains unauthorized access to a user's session on a website or application. Once the attacker has gained access to the session, they can impersonate the user and perform actions on their behalf, such as making unauthorized purchases or stealing sensitive information.

There are several techniques that attackers use to carry out session hijacking attacks:

1. Session fixation: This technique involves an attacker forcing a user's session ID to a known value, which allows the attacker to predict the user's session ID and hijack their session.
2. Session sniffing: This technique involves an attacker intercepting network traffic between the user and the server to capture the user's session ID.
3. Cross-site scripting (XSS): This technique involves an attacker injecting malicious code into a website or application that causes the user's browser to send their session ID to the attacker.
4. Cross-site request forgery (CSRF): This technique involves an attacker tricking a user into performing an action on a website or application without their

knowledge or consent, which allows the attacker to use the user's session to perform actions on their behalf.

5. Man-in-the-middle (MITM) attack: This technique involves an attacker intercepting and modifying network traffic between the user and the server to capture the user's session ID.

To prevent session hijacking attacks, website and application developers can implement various security measures such as using secure communication protocols (HTTPS), using strong session IDs that are not predictable or easily guessable, implementing two-factor authentication, and regularly monitoring network traffic for suspicious activity.

Q.2Find DoS/DDoS Attack Tools.

ANS. Low Orbit Ion Cannon (LOIC)

High Orbit Ion Cannon (HOIC)

R.U.D.Y (R-U-Dead-Yet)

Protocol and transport layer (L3/L4) attack tools

Application layer (L7) attack tools

Low and slow attack tools

Q.3 Explain SYN Flooding Attack with example

ANS. SYN Flooding is a type of denial-of-service (DoS) attack where an attacker sends a flood of TCP/SYN packets to a target server with the intention of overwhelming its ability to

respond to legitimate requests. This attack exploits a weakness in the TCP protocol's three-way handshake process, which establishes a connection between two computers.

Here's an example of how a SYN Flooding attack might work:

1.The attacker sends a flood of TCP/SYN packets to the target server, each with a different fake source IP address.

2.The server receives the SYN packets and sends a SYN/ACK response to each one, expecting a final ACK packet from the client to complete the three-way handshake process.

3.However, because the source IP addresses are fake, the server never receives the final ACK packets and the connections remain open.

4.As a result, the server's resources become consumed with these half-open connections, which can prevent it from responding to legitimate requests.

5.Eventually, the server's resources become exhausted and it becomes unable to handle any new connections, effectively shutting down the service.

Q.4List of Web App Hacking Methodology

ANS. Web app hacking methodology can vary depending on the type of web application and the specific vulnerabilities that exist. However, here is a general list of common web app hacking methodology:

1.Information gathering: Gathering information about the target web application, its architecture, and its underlying technologies.

2.Mapping: Mapping the web application and identifying all its components, such as URLs, parameters, inputs, and outputs.

3.Fuzzing: Sending a large number of inputs to the web application to identify any unexpected behavior or vulnerabilities.

4.Injection attacks: Attempting SQL injection or other injection attacks to manipulate the backend database or system.

5.Cross-Site Scripting (XSS): Injecting malicious code into a web page that is executed by a victim's browser.

6.Cross-Site Request Forgery (CSRF): Forcing a victim to perform an action on a web application without their knowledge or consent.

7.Session hijacking: Stealing or impersonating a user's session to gain unauthorized access to the web application.

8.Brute forcing: Attempting to crack passwords or other sensitive data by trying different combinations.

9.Social engineering: Using social engineering tactics to trick users into revealing sensitive information or performing actions on the web application.

10.Client-side attacks: Exploiting vulnerabilities in the client-side code of the web application, such as JavaScript or HTML.

11.Misconfiguration: Exploiting misconfigured settings or options in the web application.

12.File inclusion attacks: Attempting to access sensitive files or directories by exploiting file inclusion vulnerabilities.

Q.5 SQL Injection Methodology

ANS. SQL Injection is a type of attack that targets web applications that use SQL databases. The goal of SQL Injection is to manipulate the SQL queries that the web application uses to interact with the database. This allows an attacker to gain unauthorized access to sensitive information, modify or delete data, or perform other malicious actions.

Here are the steps involved in performing a SQL Injection attack:

1.Reconnaissance: The first step in performing a SQL Injection attack is to gather information about the web application and the database it uses. This can be done using various tools like web crawlers and network scanners.

2.Identify injection points: Once you have gathered information about the web application, the next step is to identify the injection points where SQL queries are generated dynamically. This can be done by looking for parameters in the URL, form data, cookies, and other input fields.

3.Determine the type of database: The next step is to determine the type of database that the web application is using. This can be done by examining the error messages that

are returned by the web application when you submit invalid input.

4.Test for vulnerabilities: Once you have identified the injection points and the type of database, the next step is to test for vulnerabilities. This can be done by submitting malicious input to the web application and observing the response. If the application returns an error message or displays unexpected behavior, it may be vulnerable to SQL Injection.

5.Exploit the vulnerability: Once you have identified a vulnerable injection point, the next step is to exploit the vulnerability by injecting malicious SQL code into the query. This can be done by modifying the input parameters in the URL or form data.

6.Extract data: Once you have successfully injected SQL code into the query, you can use it to extract data from the database. This can be done by using SQL statements like SELECT and UNION to retrieve data from tables.

7.Modify or delete data: If you have sufficient privileges, you can also use SQL Injection to modify or delete data in the database. This can be done by using SQL statements like UPDATE and DELETE to modify or delete records in tables.

8.Cover tracks: Finally, it is important to cover your tracks by deleting any traces of your activity. This can be done by modifying log files and other records that may reveal your presence.

Q.6 Explain sql injection with any tool

ANS. SQL injection is a type of security exploit that targets the vulnerabilities in the SQL database. It is a technique where an attacker inserts malicious SQL code into a web application's input fields, in order to manipulate the database and access sensitive information.

To explain SQL injection with a tool, let's consider the following example:

Suppose there is a login page for a website, which takes username and password as input and checks them against the database. The SQL query used by the website might look something like this:

SQL

```
SELECT * FROM users WHERE username = '<username>' AND password = '<password>'
```

The values of <username> and <password> are taken from the input fields of the login page. Now, if an attacker enters a

special string in the username field, it can manipulate the SQL query to reveal sensitive information or even gain unauthorized access to the website.

For example, the attacker can enter the following string as the username:

vbnet

```
' OR 1=1 –
```

This string is called an SQL injection payload. When the SQL query is executed with this payload, it becomes:

sql

```
SELECT * FROM users WHERE username = '' OR 1=1 --' AND password = '<password>'
```

Notice that the payload closes the opening quote of the username field and adds OR 1=1 which is always true, resulting in all users being returned. The double dash -- indicates the start of a comment which tells the database to ignore everything after it.

Now, the SQL query will return all users, regardless of their username or password, effectively bypassing the login system.

There are many tools available for performing SQL injection attacks, such as SQLMap, Havij, and many others. These tools automate the process of finding vulnerable websites and injecting payloads to exploit them. It is important to note that these tools should only be used for ethical and legal purposes, and not for any malicious intent.