Module 6

Q.1: What is Session Hijacking Explain with Techniques?

Ans: Session Hijacking also known as session stealing or cookie hijacking.

It is a type of attack where an attacker gains control of a user's session with a web application.

This allow the attacker to access sensitive information, perform unauthorized actions or impersonate the victim.

There are several techniques that can be used for session hijacking:

1. Packet sniffing: An attacker can intercept network traffic and capture packets that contain session information.
2. Session fixation: This involves an attacker forcing a user to use a specific session ID, the attacker can set session ID themselves or trick the user into using a session ID that they have provided.
3. Cross-site scripting: An attacker can inject malicious code into a website, which can then steal the victim's session ID.
4. Session sidejacking: This involves and attacker intercepting session cookies transmitted over an unencrypted connection such as HTTP

Q.2 Find DoS/DDoS Attack Tools.

Ans: The following is the list of best DoS/DDoS Attack tools:

1. GoldenEye
2. Slowloris
3. LOIC (Low Orbit Ion Cannon)
4. HOIC (High Orbit Ion Cannon

5. THC-SSL-DoS
6. HULK (HTTP Unbreakable Load King)
7. Pyloris
8. TOR's Hammer
9. XOIC
10. RUDY (R U Dead Yet?)
11. DAVOSET
12. OWASP HTTP POST

Q.3: Explain SYN Flooding Attack with example.

Ans: **SYN** Flooding is a type of cyber attack that exploits the way In which a server establishes a connection with a client through the TCP.

In a SYN flood attack, the attacker sends a large number of SYN packets to the server, but does not respond to the server's SYN-ACK packets.

This causes the server to keep waiting for the ACK packet,while tyoing up resources such as memory and processing power.

As a result, legitimate clients may be unable to establish a connection with the server. Leading to denial of service.

For example: Imagine a website that is hosted on a server. An attacker sends a large number of SYN packets to the server, pretending to be legitimate clients. The server sends SYN-ACK packets back to the attacker, but since the attacker does not respond with the ACK packets, the server keeps waiting for them. As a result, the server's resources become exhausted, and legitimate clients are unable to access the website. This can cause a significant disruption to the website's operations and can result in financial losses.

Q.4: List of Web App Hacking Methodology

Ans: Web app Hacking is a process of finding vulnerablitlies or weaknesses in web applications and exploiting them to gain unauthorized access to sensitive data or perform malicious activities.

The methodology that can be used for we app hacking is:

1. Reconnaissance: This involves gathering information about the target application, such as its architecture, technologies used, and potential vulnerabilities.
2. Scanning: This involves using automated tools to scan the target application for vulnerabilities, such as SQL injection, cross-site scripting (XSS), and others.
3. Enumeration: This involves discovering and mapping out the application's resources and functions, such as user accounts, files, and directories.
4. Exploitation: This involves using the discovered vulnerabilities to gain unauthorized access to the application or its data, such as executing code or extracting sensitive information.
5. Post-exploitation: This involves maintaining access to the compromised system or application, such as creating backdoors or installing persistent malware.
6. Covering tracks: This involves covering up the evidence of the attack, such as deleting logs or modifying system files, to avoid detection.

Q.5: SQL Injection Methodology

Ans: SQL injection is a web app vulnerability that allows attackers to manipulate SQL queries in a way that enables them to execute unauthorized actions on a database.

Following is a vulnerability that can be used to exploit SQL injection vulnerabilities:

1. Reconnaissance: The attacker first gathers information about the target application, including the type of database management system (DBMS) used, the type of SQL queries used, and the structure of the database.
2. Identifying injection points: The attacker identifies where input parameters are used in SQL queries, such as in search boxes, login forms, or product categories.
3. Fingerprinting: The attacker determines the type of DBMS being used and the version by sending specific SQL queries that trigger error messages or return different results.
4. Exploiting the vulnerability: The attacker injects SQL code into the input parameters to manipulate the query and execute unauthorized actions on the database, such as extracting sensitive data, modifying or deleting data, or adding new data.
5. Escalating privileges: If the attacker gains access to a low-privileged account, they can use SQL injection to escalate their privileges to gain administrative access to the database.
6. Covering tracks: The attacker tries to cover up the evidence of the attack by deleting or modifying logs, changing passwords, or dropping tables from the database.

Q.6: Explain sql injection with any tool

Ans: SQL injection is a web app vulnerability that allows attackers to manipulate SQL queries in a way that enables them to execute unauthorized actions on a database.

One popular tool used for SQL injection attacker is calld SQLMap.

SQLMap is a command line tool thaat aiutomates the process of detecting and exploiting SQL injection vulnerabilities in web application.

Here is an example of how SQLMap can be used to exploit a SQL injection vulnerability:

1. Reconnaissance: First, the attacker would use SQLMap to scan the target web application for vulnerabilities by running the following command: sqlmap –u http://example/com/search.php?q=test This command instructs SQLMap to test the "search.php" page for a SQL injection vulnerability using the "test" parameter.
2. Identifying injection points: If SQLMap finds a vulnerability, it will attempt to identify the injection point by sending various SQL queries with specially crafted payloads to the application.
3. Fingerprinting: SQLMap will then try to identify the type and version of the database being used, by sending specific queries to the application to extract information about the database.
4. Exploiting the vulnerability: Once SQLMap has identified the injection point and the type of database being used, it can then execute a variety of attacks to extract, modify, or delete data from the database. For example, the attacker could use SQLMap to extract all the usernames and passwords from the database, as follows: sqlmap –u http://example/com/search.php?q=test --dump

This command tells SQLMap to dump the contents of the database by extracting all the usernames and passwords.

5. Escalating privileges: If the attacker gains access to a low-privileged account, they can use SQL injection to escalate their privileges to gain administrative access to the database.
6. Covering tracks: Finally, the attacker would try to cover up their tracks by deleting or modifying logs, changing passwords, or dropping tables from the database.