NAME: NANDIGAM KAMALI HARIPRIYA
REG NO: 21BCE2746

AI for Cyber Security with IBM Qradar (AI for Web Security)
29th August, 2023

Task 5: Explain any 10 Web Server Attacks determine them using images if available

1. **URL INTERPRETATION ATTACK**
   URL INTERPRETATION Attack, also known as URL poisoning, involves attackers manipulating URLs by altering their semantics while keeping their syntax intact. This allows for the retrieval of information beyond what is intended from the web server. This attack is common with CGI-based websites and can be seen in email applications where users can reset their passwords by answering a security question. The application then opens a page for users to set an alternative email address, where the URL carrying the request to fetch user details can be modified to fetch the details of another user, making other user information vulnerable. To prevent URL interpretation attacks, vendors typically provide fixes and in-depth checks and verification of web server configurations.

2. **SQL INJECTION**
   SQL injection attacks are used to modify or extract information from a database by feeding an SQL query with parameters from the URL. This can alter data and execute stored procedures, allowing the database to perform actions only when authorized. This attack can compromise the backend database server and be catastrophic for a company. The vulnerability exploited is when SQL queries are executed without validating input data. E-commerce websites with large databases with user information are most susceptible to this attack. Fixing SQL injection requires a thorough review of source code, following least privilege for DB applications, and deleting redundant and unnecessary users and procedures.

3. **INPUT VALIDATION ATTACK**
   Input validation attack is a cyber attack where a hacker injects code into a web server or database server, allowing the attacker to retrieve or modify information. This attack bypasses client side checking using JavaScript code and can also tamper with hidden files. Data sent to the web server can be sent through various methods, including URLs, HTTP headers, POST requests, and cookies. Neglect in code writing and trusting user data can lead to such attacks. The only preventive measure is good coding practice, which should include validation for inputs like data types, ranges, meta characters, and buffer sizes.

4. **BUFFER OVERFLOW ATTACK**
   Buffer Overflow attacks involve the deliberate overflow of buffer memory reserved for user input. When an application awaits user input, it allocates a stack with a memory location for user input data. Attackers flood this space by writing arbitrary data, causing the memory stack to be full and users to deny service. This is a way to perform denial of service attacks. Hackers can also feed an executable command in

the stack, with the execution of the command dependent on the specified return address. After the stack recovers, the command may execute and grant access to certain sections of the web server. To mitigate buffer overflow attacks, vendors should provide specific fixes, but checking application bounds and conducting buffer overflow testing and source code review are effective countermeasures.

5. IMPERSONATION ATTACK

Impersonation attacks, also known as IP spoofing, involve hackers pretending to be accessing a web server with an IP that impersonates the actual one. They use special programs to create an IP packet that appears to originate from the intranet, allowing them to access the section of the web server only accessible by authorized personnel. These attacks exploit the vulnerability of authentication protocols, allowing unauthorized access to web servers and databases. Countermeasures include locking down web configurations, using a firewall to track the source of IP, and obfuscating cookies to prevent manipulation. Strong authentication modules and traffic identification can help counter these attacks.

6. PASSWORD-BASED ATTACK

Password-based attacks are common in web servers, where the authentication system relies on a user's password to gain access. Hackers can easily access sensitive information if they can access the username and password. Older applications lack strong authentication systems, making it easier for eavesdroppers to bypass the process. Breaking a password is challenging, and hackers can use algorithms to guess passwords and gain network access. Once a hacker gains access, they can modify the configuration to restore normalcy. To prevent password-based attacks, it is recommended to keep passwords long and complex, have additional security measures to protect the database, and use cryptographic storage.

7. DENIAL OF SERVICE ATTACK

Denial of Service (DOS) attacks are a popular method where a server denies serving users due to a lack of response to their request. They can be performed through various means, such as buffer flow, and attackers can exploit the web server in various ways. DOS attacks can be categorized into volume attack, protocol attack, and Application layer attack. To prevent DOS attacks from anonymous sources, a web server firewall can be implemented to inspect entire HTTL traffic and stop malicious data packets. A network audit trail should be maintained to track changes over time, and the network should be tested both locally and on the internet. By implementing these measures, security system experts can be better prepared to protect against DOS attacks.

8. BRUTE FORCE

Brute Force is a common form of web server attack where hackers crack username and password combinations using all possible iterations. This attack is most effective when no other security measures are in place. To prevent brute force attacks, create long, complex passwords and limit the number of unsuccessful login attempts in the network. Accounts may be locked after a certain number of unsuccessful attempts, but this is not a practical solution. CAPTCHAs can also be used as an extra layer of security to prevent brute force attacks.

9. SOURCE CODE DISCLOSURE

Source code disclosure attacks allow attackers to retrieve application files without parsing, analyze the source code, and identify vulnerabilities for web server attacks. This is often due to poor design or configuration errors. Attackers can access server-side scripting languages like PHP or ASP. Net, which are not meant for public viewing. To prevent attacks, thorough checks on web server proxy configuration and careful creation of URL mappings to internal servers are essential. This can be prevented by conducting thorough checks on web server proxy configuration and ensuring proper security measures.

10. SESSION HIJACKING

Session hijacking, also known as cookie hijacking, is a common issue where hackers steal user data from web applications. This occurs when a web server determines a user's session based on a cookie, which can be intercepted through network access or saved cookies. Sniffing programs are used to automate this attack. To prevent this, server-side tracking ids can be used to match connections with time stamps and IP addresses. Cryptographically generated session ids are harder to decipher. Additionally, using server session management APIs can help prevent session hijacking attacks.

Source: https://theemon.com/blog/top-10-web-server-attacks-impact-and-prevention/