Stuti Maitra Sarkar
21BCI0126

10 common web server attacks:

1. SQL Injection (SQLi):

SQL Injection is a prevalent and dangerous web application attack that occurs when an attacker injects malicious SQL queries into application inputs. If the application fails to validate and sanitize user input, these queries can manipulate the underlying SQL database. Successful SQLi attacks can lead to unauthorized access to sensitive data, data modification, or even complete control of the database. Preventing SQL Injection requires proper input validation and the use of parameterized queries or prepared statements.

2. Cross-Site Scripting (XSS):

Cross-Site Scripting is an attack where malicious scripts are injected into web pages, which are then executed by other users' browsers. Attackers exploit this vulnerability to steal user data, such as cookies, or perform actions on behalf of the victim user. XSS attacks come in various forms, including stored, reflected, and DOM-based. Proper input validation and output encoding are essential for preventing XSS vulnerabilities.

3. Cross-Site Request Forgery (CSRF):

CSRF attacks trick users into executing unintended actions without their consent. Attackers forge malicious requests on behalf of authenticated users, potentially leading to unintended actions, data modifications, and session hijacking. To prevent CSRF, web applications should use anti-CSRF tokens and ensure that actions are only executed upon user intent.

4. Denial of Service (DoS) and Distributed Denial of Service (DDoS):

DoS attacks flood a web server with a massive volume of requests to overload its resources and make it unresponsive. DDoS attacks involve multiple systems coordinating these requests, amplifying the impact. These attacks can disrupt service availability, resulting in downtime. Implementing DDoS mitigation measures, such as traffic filtering and content delivery networks, is crucial for defense.

5. Directory Traversal (Path Traversal):

Directory Traversal attacks exploit weaknesses in input validation to access files and directories outside the intended directory. Attackers manipulate input to navigate to sensitive areas of the server, potentially exposing confidential data, including configuration files or source code. Ensuring proper input validation and server configuration can mitigate these attacks.

Stuti Maitra Sarkar
21BCI0126

6. Server-Side Request Forgery (SSRF):

SSRF attacks manipulate a web application's requests to initiate requests to internal or external resources. Attackers can access sensitive data, disrupt services, and potentially launch further attacks within a network. Properly validating and sanitizing user input and using access controls can mitigate SSRF risks.

7. File Inclusion Vulnerabilities:

File Inclusion vulnerabilities allow attackers to include and execute malicious files. Attackers can manipulate input to trick the server into executing unauthorized code or accessing restricted files, potentially leading to data breaches or server compromise. Protecting against file inclusion vulnerabilities requires strong input validation and secure file access practices.

8. XML External Entity (XXE) Attacks:

XXE attacks exploit web applications that process XML input from untrusted sources. Attackers use malicious XML entities to read internal files, launch denial-of-service attacks, or even execute remote code. Preventing XXE attacks involves disabling external entity references and carefully validating XML input.

9. Server-Side Template Injection (SSTI):

SSTI vulnerabilities occur when user input is directly embedded in server-side templates. Attackers can inject malicious code into templates, leading to arbitrary code execution, data theft, or even server compromise. Protection against SSTI attacks requires robust input validation, secure template engines, and safe rendering practices.

10. Brute Force and Credential Stuffing Attacks:

Brute force attacks involve systematically trying various password combinations until the correct one is found. Credential stuffing attacks use stolen credentials from one service to gain unauthorized access to others, exploiting password reuse. These attacks can lead to account compromises, unauthorized access, and data breaches. To defend against these attacks, organizations should implement account lockout mechanisms, rate limiting, and educate users about password security.