Niraianbu porchezhian
21BCE0965

Task-4

In the intricate world of web servers, vulnerabilities can be exploited by various attacks, each with its unique modus operandi and potentially catastrophic consequences. Let's navigate through ten such assailants that cunningly prey on the cracks in web server defenses.

1.SQL Injection (SQLi):

Picture this: a seemingly innocent search bar on a website. Unbeknownst to users, hackers deftly slip malicious SQL queries into it. When inadequately validated by the server, these queries can compromise databases, granting unauthorized access to sensitive information.

2. Cross-Site Scripting (XSS):

Imagine a cyber-illusionist conjuring a malicious script within a comment box. As unsuspecting users read the comments, this script springs to life within their browsers, stealing credentials or even redirecting them to a fake website.

3. Cross-Site Request Forgery (CSRF):

Visualize a situation where an authenticated user, logged into a shopping site, unknowingly clicks a link that triggers a purchase on their behalf. Crafty attackers manipulate this active session to perform actions without the user's knowledge.

4. Distributed Denial of Service (DDoS):

Envision an army of virtual zombies relentlessly bombarding a web server with an overwhelming torrent of requests. This digital onslaught incapacitates the server, rendering it inaccessible to genuine users.

5. Server-Side Request Forgery (SSRF):

Picture an attacker exploiting vulnerable server-side code, tricking the server into making unauthorized requests to other systems. This can lead to breaches of sensitive data or even launching attacks on internal networks.

6. XML External Entity (XXE) Attack:

Imagine a scenario where a misconfigured XML parser unknowingly gobbles up malicious input. The attacker cleverly injects external entities, opening pathways for information leakage or crippling the server.

7. Remote File Inclusion (RFI) and Local File Inclusion (LFI):

Visualize hackers manipulating file paths in a web application. By forcing the server to include external files, they can gain unauthorized access, execute arbitrary code, or even compromise the entire system.

8. Directory Traversal:

Imagine an attacker exploiting lax input validation to navigate directories beyond intended boundaries. By ascending paths, they can gain unauthorized access to files, unearthing sensitive information.

9. HTTP Response Splitting:

Envision a stealthy attacker injecting malicious input containing newlines. When the server responds, these newlines divide responses into two parts. This manipulation can lead to browser-based attacks or cache poisoning.

10. Brute Force Attacks: Picture an assailant relentlessly trying various username and password combinations, like a persistent locksmith attempting every key. This method preys on weak credentials or defaults, ultimately breaching server defenses.

To fortify against these crafty adversaries:

- Regularly update web server software and applications to patch vulnerabilities.
- Implement rigorous input validation and output encoding to thwart injection attacks.
- Enforce robust authentication mechanisms and stringent password policies.
- Configure vigilant firewalls and intrusion detection systems to detect and repel attacks.
- Safeguard data in transit with SSL/TLS encryption.
- Employ the guardianship of web application firewalls (WAFs) to filter and scrutinize incoming traffic.
- Foster a culture of secure coding practices to weave a formidable web of defense.

Remember, the digital realm is an ever-evolving landscape, demanding unwavering vigilance. Regular assessments, continuous learning, and swift adaptation are the sentinels of a resilient web server fortress.