Week 10

Day 46

Web Application Attacks

Web Application Attacks target vulnerabilities in web-based applications that arise due to insecure coding practices, poor configuration, or lack of proper security controls. Since web applications handle sensitive user data, they are common targets for attackers. Understanding these attacks is essential for securing modern web systems.

Tools Used

- **Burp Suite:** A web vulnerability testing tool used to intercept, modify, and analyze HTTP/HTTPS requests and responses.
- **OWASP ZAP:** An open-source web application security scanner used to identify vulnerabilities automatically and manually.

Common Web Application Attacks Studied

i. **SQL Injection (SQLi):** Manipulating database queries through malicious input.

ii. **Cross-Site Scripting (XSS):** Injecting malicious scripts into web pages viewed by users.

iii. **Cross-Site Request Forgery (CSRF):** Forcing users to perform unwanted actions without their knowledge.

iv. **Broken Authentication & Session Management**

v. **Insecure Direct Object References (IDOR)**

vi. **Security Misconfigurations**

Practical Work Performed

- Used Burp Suite to:
    - Intercept and analyze HTTP/HTTPS traffic
    - Modify requests to test input validation
    - Identify authentication and session-related flaws
- Used OWASP ZAP to:
    - Perform automated vulnerability scanning
    - Identify common OWASP Top 10 vulnerabilities
- Practiced testing in a safe, authorized web application environment.
- Studied the impact of vulnerabilities and recommended mitigation techniques.

Work Description

The topic of Web Application Attacks was studied in detail as per the cybersecurity syllabus. All practical tasks were performed in a controlled lab environment on authorized applications, ensuring ethical and legal compliance. This provided hands-on experience in identifying, analyzing, and understanding web application vulnerabilities.

Conclusion

Web application security is critical due to the widespread use of web-based services. Through hands-on practice with Burp Suite and OWASP ZAP, practical understanding of common web attacks and their detection was gained. This study highlighted the importance of secure coding practices, regular vulnerability testing, and proactive security measures to protect web applications from cyber threats.

Day 47

SQL Injection (SQLi)

SQL Injection is a critical web application vulnerability where an attacker manipulates input fields to execute malicious SQL queries on a backend database. This occurs when user input is not properly validated or sanitized. SQL injection can allow attackers to view, modify, or delete database data and, in severe cases, gain administrative access to the system.

Tools Used

- **Burp Suite:** Used to intercept, analyze, and modify HTTP/HTTPS requests to identify injection points.
- **SQLmap:** An automated penetration testing tool used to detect and exploit SQL injection vulnerabilities.

Types of SQL Injection Studied

i. **In-band SQL Injection** (Error-based, Union-based)

ii. **Blind SQL Injection** (Boolean-based, Time-based)

iii. **Out-of-band SQL Injection**

Practical Work Performed

- Used Burp Suite to:
    - Intercept HTTP requests and identify vulnerable input parameters
    - Modify request payloads to test for SQL injection
- Used SQLmap to:
    - Automatically detect SQL injection vulnerabilities

    - Enumerate databases, tables, and columns in a controlled environment

- Studied how SQL injection impacts database confidentiality and integrity.
- Analyzed secure coding practices to prevent SQL injection attacks.

Work Description

The topic of SQL Injection was studied in detail as per the cybersecurity syllabus. All practical exercises were conducted on authorized web applications in a controlled lab environment, ensuring ethical and legal compliance. These exercises provided hands-on experience in identifying and understanding SQL injection vulnerabilities.

Conclusion

SQL Injection remains one of the most dangerous web application vulnerabilities. Through practical exercises using Burp Suite and SQLmap, hands-on knowledge was gained in detecting and analyzing SQL injection flaws. This study emphasized the importance of proper input validation, parameterized queries, and secure database handling to prevent SQL injection attacks.

Day 48

XSS and CSRF

Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF) are common and dangerous web application vulnerabilities that exploit trust between users and web applications. Both attacks target user interactions and can lead to data theft, unauthorized actions, and session compromise.

Cross-Site Scripting (XSS)

XSS occurs when a web application fails to properly validate or sanitize user input, allowing attackers to inject malicious scripts that execute in a victim's browser.

Types of XSS Studied

i. **Stored XSS:** Malicious script is permanently stored on the server.

ii. **Reflected XSS:** Script is reflected through user input in real time.

iii. **DOM-based XSS:** Script is executed within the browser via insecure client-side code.

Impact of XSS

- Session hijacking
- Cookie theft
- Defacement of web pages
- Redirection to malicious websites

Cross-Site Request Forgery (CSRF)

CSRF is an attack where a victim is tricked into performing unauthorized actions on a trusted website while logged in. The attacker exploits the trust that a web application has in the user's browser.

Impact of CSRF

- Unauthorized fund transfers
- Password changes
- Account manipulation

Tools Used

- **Burp Suite:** Used to intercept and analyze HTTP requests, identify vulnerable parameters, and test input validation.

Practical Work Performed

- Used Burp Suite to:
    - Intercept and modify requests to identify XSS vulnerabilities
    - Test form inputs for reflected and stored XSS
    - Analyze request tokens and headers related to CSRF protection
- Studied how improper validation leads to XSS and CSRF attacks.
- Reviewed secure coding techniques to prevent these vulnerabilities.

Work Description

The topic of XSS and CSRF was studied in detail as per the cybersecurity syllabus. All practical tasks were performed on authorized web applications in a controlled lab environment, ensuring

ethical and legal compliance. These exercises helped in understanding the exploitation and prevention of client-side web vulnerabilities.

Conclusion

XSS and CSRF are significant threats to web application security. Through hands-on practice using Burp Suite, practical understanding of how these vulnerabilities occur and how they can be detected was gained. This study emphasized the importance of input validation, output encoding, CSRF tokens, and secure session handling to protect web applications from client-side attacks.

Day 48

Sniffing & Spoofing

Sniffing and Spoofing are network-based attack techniques used to intercept, manipulate, or impersonate network traffic. These attacks exploit weaknesses in network protocols and insecure communication, making them important topics in cybersecurity and network defense.

Sniffing

Sniffing is the process of capturing and analyzing data packets as they travel across a network. Attackers can use sniffing to gather sensitive information such as usernames, passwords, session cookies, and confidential data if traffic is not encrypted.

Types of Sniffing

- **Passive Sniffing:** Monitoring traffic without altering it.
- **Active Sniffing:** Involves injecting packets to capture traffic on switched networks.

Spoofing

Spoofing involves impersonating a trusted device or user by falsifying network identity information such as IP addresses, MAC addresses, or ARP responses. It is commonly used to redirect traffic or perform man-in-the-middle attacks.

Common Types of Spoofing

- **ARP Spoofing**
- **IP Spoofing**
- **DNS Spoofing**
- **Tools Used**

- **Wireshark:** Used to capture and analyze network packets in real time.
- **Ettercap:** A network attack tool used for performing ARP spoofing and man-in-the-middle attacks in controlled environments.

Practical Work Performed

- Used Wireshark to:
    - Capture live network traffic
    - Analyze protocols such as TCP, UDP, HTTP, and DNS
    - Observe unencrypted data transmission
- Used Ettercap to:
    - Perform ARP spoofing in a controlled lab setup
    - Simulate man-in-the-middle attacks
- Studied the impact of sniffing and spoofing on network security.
- Analyzed defense mechanisms such as encryption, secure protocols, and network monitoring.

Work Description

The topic of Sniffing & Spoofing was studied in detail as per the cybersecurity syllabus. All practical tasks were conducted in a controlled lab environment on authorized systems, ensuring ethical and legal compliance. The exercises provided hands-on experience in understanding network vulnerabilities and attack techniques.

Conclusion

Sniffing and spoofing attacks pose serious threats to network security if proper safeguards are not implemented. Through practical exercises using Wireshark and Ettercap, hands-on understanding of packet interception and identity impersonation techniques was gained. This study highlighted the importance of encryption, secure network configurations, and continuous monitoring to protect networks from such attacks.

Day 50

Session Hijacking

Session Hijacking is a cyber-attack in which an attacker takes control of a user's active session by stealing or manipulating session identifiers such as session IDs or cookies. Once the session is hijacked, the attacker can impersonate the victim and gain unauthorized access to sensitive information or services.

Tools Used

- **Burp Suite:** Used to intercept and analyze HTTP/HTTPS traffic, cookies, and session tokens.
- **Ettercap:** Used to perform man-in-the-middle (MITM) attacks to capture session data in a controlled network environment.

Types of Session Hijacking Studied

- **Active Session Hijacking**
- **Passive Session Hijacking**
- **Session Fixation**
- **Man-in-the-Middle Based Hijacking**

Practical Work Performed

- Used **Burp Suite** to:
    - Intercept requests and responses
    - Analyze session cookies and authentication tokens

    - Test session management weaknesses

- Used **Ettercap** to:
    - Perform ARP spoofing for MITM simulation
    - Capture session-related network traffic
- Observed how unencrypted traffic can expose session information.
- Studied preventive measures such as HTTPS, secure cookies, and session expiration.

Work Description

The topic of Session Hijacking was studied in detail as per the cybersecurity syllabus. All practical exercises were conducted in a controlled lab environment on authorized systems, ensuring ethical and legal compliance. These exercises provided hands-on experience in understanding how session vulnerabilities can be exploited.

Conclusion

Session hijacking is a serious threat to web application security. Through practical exercises using Burp Suite and Ettercap, hands-on understanding of session handling flaws and attack techniques was gained. This study emphasized the importance of secure session management practices such as encryption, secure cookies, and proper authentication controls to protect user sessions.