Project S1 2025

Task 1 – Web Application

Develop a web application of a theme of your choice using a Content Management System (CMS) (for example, Drupal, Joomla ….). The application will incorporate multiple security methods to safeguard user data, protect against common threats, and ensure reliable user experience.

Project Requirements:

1. **User Authentication:** Implement a robust user authentication system that ensures secure user login and registration.
2. **Access Control:** Develop a comprehensive access control system to restrict unauthorized access to sensitive areas or data. User roles and permissions should be properly defined, and access to certain resources should be logged and monitored.
3. **Data Encryption:** Implement data encryption methods such as SSL/TLS to secure data transmission between clients and the server. Use encryption algorithms to protect sensitive data stored in the database.
4. **SQL Injection Prevention:** Employ prepared statements or use Object Relational Mapping (ORM) libraries to mitigate SQL injection risks.
5. **Session Management:** Manage user sessions securely, including proper session timeout settings and session token security.
6. **File Upload Security:** your application allows file uploads, ensure that uploaded files are scanned for malware, and that only authorized users can access or download them.
7. **Error Handling and Logging:** Implement robust error handling and logging mechanisms to capture, log, and report security-related events or exceptions.
8. **Penetration Testing:** Perform penetration testing to identify vulnerabilities and weaknesses in your application. Document your findings and apply fixes as needed.

Task 2 – Network Traffic Analysis and Decryption Challenge

You are tasked with capturing, analyzing, and decrypting network traffic to investigate potential security threats.

1. Capture Encrypted Network Traffic:

- Use Wireshark or any tool/program to capture HTTPS (TLS-encrypted) network traffic on your own machine.
- Visit secure websites (e.g., https://example.com) to generate encrypted traffic.
- Save the captured packets in a PCAP file.

2. Simulate a Network Attack:

- Generate suspicious network activity:
  - Conduct a port scan using Nmap.

- Attempt multiple failed logins on an SSH server.
- Send malformed packets using Scapy (Python).
- Use a simple MITM proxy to capture unencrypted credentials.

- Capture this attack traffic in a separate PCAP file.

3. Extract and Analyze Network Data:

- Identify encrypted HTTPS sessions and analyze handshake information.
- Locate and describe any attack patterns present in the traffic.
- If applicable, use a TLS session key (exported from a browser or captured using SSLKEYLOGFILE) to decrypt the encrypted traffic in Wireshark.

4. Implement Real-Time Network Monitoring:

- Use a tool (**Snort**, **Suricata**, …) to monitor live network traffic.
- Configure rules to detect at least **one type of attack**.
- Capture logs or alerts generated by your monitoring system and analyze them.

5. Report and Secure Your Findings:

- Document each step taken, with screenshots where necessary.
- Explain the security risks associated with the attack techniques observed.
- Propose mitigation strategies to prevent such attacks in real-world networks.

INSTRUCTIONS AND INFORMATION

- The project can be worked on individually or in groups of two (maximum).
- **Each Group must choose one task to work on.**
- Pay attention! In the case of binomials, the grade is the same for binomials; during the last session, a question and/or task missed by one of the two students will affect the overall grade of the pair.
- Plagiarism will not be tolerated. You will risk failing the material in case of plagiarism!
- Ensure ethical usage: Do not attack unauthorized systems.
- Use existing tools (Wireshark, tshark, tcpdump, OpenSSL, Nmap, Scapy, etc.).
- No need to develop custom decryption algorithms—focus on proper analysis and documentation.