Welcome to Cyber Clinic

Session 3:

OWASP Juice Box
(Packet Interception)

What is Packet Interception / Burp Suite?

- Packet interception, also known as packet sniffing, is the process of capturing and analyzing data packets as they travel across a network. It allows security professionals to inspect traffic for anomalies, performance issues, or signs of malicious activity.Use
- Cases: Network diagnostics and performance monitoring, Intrusion detection and forensic analysis, Compliance auditing (e.g., GDPR, ISO 27001)
- Tools: Wireshark, tcpdump, IDS/IPS systems

What is Packet Interception / Burp Suite? **2**

- Burp Suite is a professional-grade web application security testing platform used to identify and exploit vulnerabilities in websites. It acts as a proxy between the browser and the server, allowing users to intercept, modify, and replay HTTP/S requests.
- Features: Proxy for traffic interception, Repeater for manual testing, Intruder for automated attacks, Scanner (Pro) for vulnerability detection
- Use Cases: Penetration testing, Secure development lifecycle (SDLC) integration, Training with labs like OWASP Juice Shop

What is OWASP Juice Box?

- Definition: OWASP Juice Shop is a modern, intentionally vulnerable web application designed to teach and test web security. It includes challenges based on the entire OWASP Top Ten and mimics real-world flaws found in production systems.
- How It Works: Users interact with a fake e-commerce site and attempt to exploit vulnerabilities like SQL injection, XSS, broken authentication, and more. Progress is tracked via a hidden scoreboard.
- Use Cases: Security training and workshops, Capture the Flag (CTF) competitions, Tool testing (e.g., Burp Suite, OWASP ZAP), Governance and compliance simulations Tech Stack:Node.js, Express, Angular — fully open source and deployable via Docker, cloud, or local environments.

Installing and using OpenVPN on Kali Linux

- OpenVPN 2.6.12 should be pre-installed on your Kali Linux VM
- Download the .ovpn file from TryHackMe then drag and drop the file from the file explorer into the desktop of your Kali virtual machine

TryHackMe Setup

- https://tryhackme.com/room/openvpn
- On VMWare Machines Only
- Download OpenVPN GUI
- Import VPN File
- Follow Setup wizard as instructed for windows
- On Linux run the following: sudo apt install openvpn
- Locate full path in downloads then
- sudo openvpn /path-to-file/file-name.ovpn
- To install without VPN use the free hour session provided via the link above

OWASP Top Ten

- The OWASP Top Ten is a globally recognized list of the most critical web application security risks, maintained by the Open Worldwide Application Security Project (OWASP). It helps developers, security teams, and organizations understand and mitigate common vulnerabilities in web applications.
- It covers these 10 risks:
- Broken Access Control, Cryptographic Failures, Injection, Insecure Design, Security Logging and Monitoring Failure, Server Side Request Forgery
- Security Misconfiguration, Vulnerable and Outdates Components, Identification and Authentication Failure, Software and Data Integrity Failure

Broken Access Control

Definition: Users gain access to data or functions they are not authorized to use

How it works: Attackers manipulate tokens, URLs, roles to bypass restrictions

Example: accessing /admin or modifying userId=123 to userId=125

Risks: Data Leakage, Privilege Escalation, Regulatory Violations

Mitigation: enforce RBAC, deny by default, log and monitor attempts

Cryptographic Failures

Definition: Weak or missing encryption exposes sensitive data in transit or at rest.

How it works: Applications use outdated algorithms, store data in plaintext or misconfigure TLS.

Example: Storing passwords without hashing or using HTTP instead of HTTPS.

Risks: Credential Theft, GDPR Non Compliance, Data Tampering

Mitigation: Use strong encryption, use https site wide, hash passwords accordingly

Injection

Definition: Malicious input is interpreted as code or commands by the backend.

How it works: Threat Actors inject SQL, OS or LDAP commands via input fields.

Example: ‘ OR 1=1– which is a login bypass

Risks: Data exfiltration, full system compromise, compliance failure

Mitigation: Use parameterized queries, sanitize and validate all data, apply least privilege to backend

Insecure Design

Definition: Flawed architecture, software or logic which leads to system vulnerabilities

How it works: Security is not considered during design

Example: A password reset feature without rate limiting or token expiration

Risks: Business logic abuse, long term technical debt, exploitable design flaws

Mitigation: Integrate secure design principles, perform threat modeling, include security in SDLC

Security Misconfiguration

Definition: default settings, verbose error messages, or exposed features which lead to system vulnerabilities

How it works: Servers or frameworks are deployed insecurely

Example: Directory listing enabled or admin panels exposed to the public

Risks: Unauthorized access, information disclosure, attack surface expansion

Mitigation: Harden configurations, disable unused features, automate security baselines and reviews

Vulnerable and Outdated Components

Definition: Using libraries and platforms with known vulnerabilities

How it works: Apps rely on outdated packages without patching or version control

Example: Using a old jQuery version with known XSS flaws

Risks: Supply chain compromise, exploitation of known CVEs, inherited vulnerabilities

Mitigation: Maintain a SBOM, Monitor for CVEs, patch and update regularly

Identification and Authentication Failures

Definition: Weak login system or session handling allowing actors to impersonate users

How it works: Actors exploit predictable passwords, session fixation or missing MFA

Example: Brute forcing login or reusing stolen session tokens

Risks: Account takeover, Unauthorized access, Identity fraud

Mitigation: Enforce MFA, use secure season tokens, implement lock out and rate limiting

Software and Data Integrity Issues

Definition: Trusting unverified code, updates, or data sources

How it works: Actors tamper with pipelines, dependencies or config files

Example: Malicious code injected via a compromised NPM package

Risks: Supply chain attacks, code execution, data corruption

Mitigation: use signed packages, secure pipelines, data corruption

Security Logging and Monitoring Failures

Definition: Lack of visibility into security events or delayed incident response

How it works: Apps fail to log critical actions or alert on suspicious behavior

Example: No logs for failed logins or privilege changes

Risks: Undetected breaches, forensic gaps, compliance violations

Mitigation: Centralize and monitor logs, set up alerts for anomalies, test and rehearse incident response plans

Server Side Request Forgery

Definition: A vulnerability where the server is tricked into making unintended requests often to internal systems using the user supplied URLs

How it works: The application accepts URL input from user which may not be validated leading to it fetching internal resources

Example: Submitting https://169.254.169.254/latest/meta-data to extract AWS instance credentials

Risks: Bypasses firewalls and network segmentation, Access internal only endpoints, can lead to data exfiltration or remote code execution

Mitigation: Whitelist trusted domains, block internal IP ranges, restrict outbound traffic from the server

Risk Analysis

- Executive Summary
- Vulnerability Overview
- Risk Analysis Table
- Standards and Compliance Mapping
- Mitigation Recommendations
- Lessons Learnt

DISCLAIMER

- This session is intended for educational and ethical cybersecurity training only. All demonstrations are conducted in controlled environments using intentionally vulnerable applications. No real systems were harmed or accessed. Participants are expected to follow applicable laws, institutional policies, and responsible disclosure practices. The Clinic Staff assume no liability for misuse of the techniques discussed

- Follow everything in this session as demonstrated on the appropriate platforms as provided, any use of techniques outside of the material provided is not subject to liability of the clinic staff.