API Security and Pentesting

Module 1: Introduction to API Security

1. Introduction to Application Programming Interface
    1. What is an API?
    2. Need for an API
    3. Why should you secure your APIs?
    4. APIs vs Web Applications
2. Understanding API Architecture
    1. Overview of HTTP Protocol
    2. Stateless and Stateful Requests
    3. Overview of API Architecture
        1. API Protocols
        2. API Data formats
        3. Different types of APIs
    4. Simple Architecture
        1. How are APIs typically deployed?
    5. Complex Architecture
3. Threat modelling of APIs
    1. Traditional VAPT vs API VAPT
4. API Defenses
    1. Input validation
    2. Authentication
    3. Authorization
    4. Data encryption
    5. Transport Security
    6. Error Handling and Logging
    7. Supply chain security

5. **Hands-on Exercises:**
    1. Understanding an API language (endpoints, verbs, and state)
    2. Understand cURL command, perform CRUD operations using API

Module 2: API Security Tools of the trade

1. Moving parts in an API
    1. API Gateway
    2. Load Balancer/Reverse Proxy
    3. Message Queues
2. Critical toolchain for API Development
    1. Source code management
    2. CI/CD tools
    3. Artefact management
    4. Cloud Platform
    5. Infrastructure as code
    6. Monitoring and logging tools
    7. Collaboration tools
3. Containerization
4. Ability to talk to API
    1. cURL(curl)
    2. Postman
    3. OpenAPI (Swagger)
    4. Python
    5. A MITM Proxy

5. **Hands-on Exercises:**
    1. Setup the burp suite for API security testing
    2. Understand APIs using OpenAPI specifications

    3. Performing reconnaissance on API
    4. Enumerate user accounts from an API
    5. Hunt for vulnerable APIs

Module 3: Authentication Attacks and Defenses

1. Overview of API authentication
2. Types of Authentication
    1. No Authentication(Public APIs)
    2. HTTP Basic Authentication
    3. API Token Authentication
    4. OAuth/OIDC Authentication
    5. JSON Web Tokens(JWTs)
    6. Mutual TLS
3. Authentication Attacks
    1. Brute force
    2. Weak password storage
    3. Password reset workflows
    4. Account lockouts
    5. Insecure OAuth configuration
    6. Insecure JWTs validation
4. Authentication Defenses
    1. Secure Authentication workflows
    2. Strong password and key validation
    3. Multi-factor authentication
    4. Securely storing the tokens
        1. Cookies and Local Storage
        2. Token storage and XSS
    5. Rate limiting
    6. CAPTCHA

5. **Hands-on Exercises:**
    1. Talk to an API using Basic, API Token and OAuth and JWTs
    2. Broken Authentication using API Token, Oauth and JWTs
    3. Exploiting weak passwords
    4. Bruteforcing the passwords
    5. Exploit misconfigurations in scope
    6. Token Forging
    7. JSON Web Token Abuse

Module 4: Authorization Attacks and Defenses

1. Overview of API Authorization
2. Types of Authorization
    1. No Authorization
    2. Role-Based Access Control (RBAC)
    3. Discretionary access control
    4. Attribute-Based Access Control (ABAC)
    5. Relationship-Based Access Control (ReBAC)
3. Authorization Attacks
    1. Horizontal Privilege Escalation
    2. Vertical Privilege Escalation
    3. Broken Object Level Authorization
    4. Broken   Level Authorization
    5. Misconfigured permissions
4. Authorization Defenses
    1. Attribute-Based Access Control (ABAC)
    2. Relationship-Based Access Control (ReBAC)
    3. Implementing Authorization using Open Policy Agent (OPA)

5. OAuth 2.0 and OAuth 2.1
    1. OAuth Specification
    2. Different Authorization workflows
    3. Different types of Tokens
        1. Access Token
        2. Refresh Token
        3. ID Token
6. **Hands-on Exercises:**
    1. Bypassing the access control
    2. Exploiting Broken Object Level Authorization
    3. Exploiting Broken Function Level Authorization
    4. Exploit weak/default permissions
    5. Finding another cell phone user’s location

Module 5: Input validation Threats and Defenses

1. What is Input validation
    1. Implementing input validation
    2. Client-side vs. server-side validation
    3. Whitelisting & Blacklisting
    4. Regular Expressions
2. Injection vulnerabilities
    1. OWASP API Top 10
    2. Cross-Site Scripting (XSS)
    3. SQL Injection
    4. ORM Injection
    5. NoSQL injection
    6. Server-Side Request Forgery
    7. Deserialization Issues
    8. Mass Assignment Issues

3. Fuzzing
    1. What is fuzzing
    2. Fuzzing APIs using open-source and commercial tools
    3. Tools to fuzz
        1. Burp Suite Intruder
        2. Wfuzz
        3. FFUF
4. Injection Defenses
    1. Input validation
    2. Output Encoding
        1. HTML Encoding
        2. Character Encoding
        3. Output Encoding
    3. Prepared Statements
    4. Content Security Policy
    5. Trusted Types
5. **Hands-on Exercises:**
    1. Input validation using industry best practices
    2. Find a way to get free coupons without knowing the coupon code
    3. Vulnerability assessment approaches effectively
    4. Fuzzing APIs using ffuf
    5. Fuzz with postman for improper asset management
    6. Exploiting Mass Assignment Vulnerabilities

Module 6: Other API Security Threats

1. Improper Inventory and asset management
2. Excessive Data Exposure
3. Lack of Rate Limiting
4. Security Misconfigurations
5. Insufficient Logging & Monitoring
6. Attacking caching layers(Memcache, proxies, etc.,)
7. Abusing Microservices
8. Attacking GraphQL APIs
9. Attacking SOAP APIs
10. Attacking SPA backed by APIs
11. Post-Exploitation in the API world
12. **Hands-on Exercises:**
    1. Bypass the rate-limiting
    2. Extract sensitive data by abusing default API behaviour
    3. Find and mitigate the IDOR vulnerability
    4. Exploit the CORS misconfiguration
    5. Exploit the undisclosed API calls
    6. Sensitive information in the server logs

Module 7: Other API Security Defenses

1. GraphQL API Security Best Practices
2. SOAP API API Security Best Practices
3. Protecting SPA backed by APIs

4. Data Security
    a. What are Encoding and Decoding
    b. Escaping
    c. hashing
    d. Encryption and Decryption
    e. Encoding vs. Encryption
    f. Securing Data at Rest using Encryption
        - Password storage and its considerations
        - Picking a secure algorithm
        - Storing credentials for service-to-service communication
        - Secure file storage and access management
    g. Securing Data in Transit using TLS
5. Rate Limiting Best Practices at different stages
    a. Reverse Proxy
    b. Load Balancer
    c. API Gateways and WAFs
6. Security headers
    a. Cache-Control
    b. Content Security Policy
        - Implementing CSP at Scale
        - Common Misconfigurations while using CSP
        - Defending against common security issues using CSP
            1. XSS
            2. CSRF
    c. X-Frame-Options
    d. X-XSS-Protection
    e. HTTP Strict Transport Security (HSTS)
    f. Cross-Origin Resource Sharing (CORS)
        - Cookie Based Implementations
        - Token Based Implementations

7. Implement Sufficient Logging & Monitoring
    a. Logging using Syslog format
    b. Using ELK to capture the log data
8. **Hands-on Exercises:**
    a. Bypassing CSP header
    b. Configure HSTS to prevent MITM attacks
    c. Find the missing security headers and fix them
    d. Implement Rate limiting using HA Proxy and Nginx

Module 8: Implementing API Security Mechanisms

1. API Security Design Best Practices
2. Authentication Implementation (MFA)
3. Authorization Implementation
4. Rate-limiting Implementation
5. Securely store secrets using Hashicorp Vault
6. Secure Logging Implementation
7. Data Security Implementation
8. Using Transport Layer Security (TLS)
9. **Hands-on Exercises:**
    1. Bypassing WAFs and Security Products
    2. How to configure TLSv1.2 and beyond securely to achieve A+ on SSLlabs scans
    3. Adding CSP header to the API
    4. Second-order sensitive information leakage

Module 9: API Security, the DevSecOps Way

1. OWASP ASVS Framework
    1. What is ASVS, and how is it useful
    2. How to create checklists
    3. How to use ASVS framework to secure application
2. Automated vulnerability discovery
3. Find Insecure Dependencies using Software Component Analysis
4. Find vulnerabilities in code using Static Application Security Testing
5. Automat API attacks using Dynamic Application Security Testing
6. Fixing API Security issues at scale
7. **Hands-on Exercises:**
    1. Create a simple CI/CD pipeline
    2. Deploy a microservice/docker container to production
    3. Exploit a microservice using docker misconfiguration
    4. Exploit a microservice using API vulnerabilities
    5. Find and Fix API Security issues using SCA, SAST, and DAST in CI/CD pipelines