AND201 - DEFENDING ANDROID

Course Learning Objectives

The OWASP Top 10 provides a list of common vulnerabilities in software application, and apps developed in Android with Java or Kotlin are no exception. This course details baseline guidance for developers to address vulnerabilities in Android apps by delving into the causes of common vulnerabilities and the defenses to mitigate them. Developers will explore secure coding practices that defend against weaknesses such as poor authentication, sensitive data leakage, weak cryptography, and injection attacks.

Description

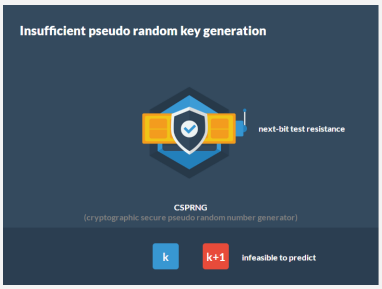
Explore defenses against common vulnerabilities in Android applications developed with Java and Kotlin. This course covers industry best practices in secure coding as it relates to authentication and authorization, secure data transfers, secure data storage, cryptography, and secure data ingestion

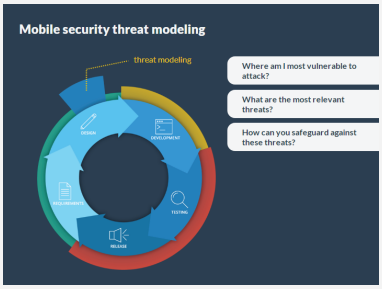
Audience

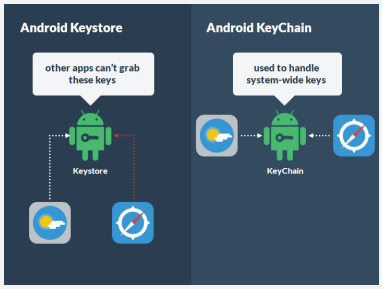
Android mobile application developers
Android application architects
Security professionals

Time Required

Tailored learning - 65 minutes total







AND201 - DEFENDING ANDROID

Course Outline

1. Authentication and authorization

- Authentication vs authorization
- Untrusted incoming requests
- Client-side authentication bypass
- No account lockout or throttle
- Insufficient password policy requirements
- Insufficient authorization requirements
- Integration with password managers
- Password management applications
- Suggest strong password at account creation
- Token management on the server side
- How OAuth works
- About 'appsecret_proof'

2. Secure data transfer

- Unencrypted communications
- Improper certificate validation
- Code: Mismatched certificate bypass
- Secure Network Connections
- Certificate pinning
- Code: Implementing certificate pinning

3. Secure data storage

- Sensitive data stored in plaintext
- Sensitive data stored on a device
- Background apps
- Automatic snapshots
- Shared clipboards
- Screen recording and broadcasting
- Store sensitive data
- Code: Encrypt application data storage area
- Code: Encrypting files on the SD card
- Code: Storing the master key
- Code: Store sensitive data in a Shared Preference Object
- Android Keystore and KeyChain
- Code: Store credentials in Android Keystore
- Clear data for background apps
- Sanitize the snapshot screen
- Code: Disable clipboard copy and paste
- Code: Clear last entry on clipboard
- Mask sensitive inputs

4. Cryptography

- Insufficient pseudo random key generation
- Symmetric key cryptography with hardcoded keys
- Code: Random key generation
- Password-based key derivation function (PBKDF2)
- PBKDF2 in an application
- Code: Key generation
- Code: Encryption with PBKDF2
- Implementing PBKDF2
- Code: Decryption with PBKDF2

5. Secure data ingestion

- About secure data ingestion
- Client-side SQL Injection
- WebView input 1 of 3
- WebView input 2 of 3
- WebView input 3 of 3
- Keyboard data caching
- Third-party keyboards
- Parameterizing SQL commands
- Safer SQL solution
- Data cleansing against XSS
- Disable local file access
- Disable UITextField caching
- Detect third-party keyboards

SecurityCompass