PSCG

Premium Software Consulting Group

For quote and registration: email ron@thepscg.com or call +972-54-5529466 (Course #: RASEC504)

Advanced Android Security - With Web Application Security

Length: 7 Days
Type: Hands-On
Target Audience: Mobile Developers, IT Managers, Security Personnel with Java experience.

In this comprehensive hands-on course, combining classical binary and Web Application Security Awareness modules with both Android Application Security and Android Enterprise Security modules, you will learn Android security at all possible levels, from the bootloader, through building Secure Applications, and via the end-user security and Enterprise Mobility Management. We will learn to harden both the Operating System (for device builders), and the application code itself, to protect both the organization's Intellectual Property and the user's personal data , and will also learn to take advantage of Android Provisioning services to support an IT manager perspective. The course is intended for developers, or former developers with practical Java experience. No previous Android experience is required, but it is highly recommended.

Note: The course is based on the Marshmallow and Nougat versions. Earlier versions can be targeted without additional cost, upon customer request.

Course Outline:

- Introduction to Security
  - Introduction to Security
  - Legacy and modern threats
  - Physical and Hardware Security
  - Cyber Security terminology
  - Real-time attack map demonstration. Why and who should be worried.
  - Present-time attack vectors
  - Present-time defense solutions
  - The Security Process
  - Introduction to Threat Modeling
  - Incident Response in Enterprises
- Binary Exploitation Overview
  - Motivation: Exploit Piggy-backing on Higher Level Technologies
  - Buffer Overflows and stack smashing attacks
  - Shellcode construction
  - String format errors
  - Integer overflows
  - Heap overflows and heap spraying techniques, memory corruption and double free attacks
  - Understanding dynamic library and hooking injection attacks, misusing LD_PRELOAD

    - Compiler and Operating System mitigation techniques
    - Return Oriented Programming and mitigation techniques
    - Understanding combined data leak attacks
    - Piggy-Backing revisited: Attacks on PDF, Flash, JavaScript, WebKit, Email, Images, Video Payload, Applets, JVM.
- Web Application Security
    - Web Application Architecture
    - The OWASP top 10 vulnerabilities
        - A1-Injection
        - A2-Broken Authentication and Session Management
        - A3-Cross-Site Scripting (XSS)
        - A4-Insecure Direct Object References
        - A5-Security Misconfiguration
        - A6-Sensitive Data Exposure
        - A7-Missing Function Level Access Control
        - A8-Cross-Site Request Forgery (CSRF)
        - A9-Using Components with Known Vulnerabilities
        - A10-Unvalidated Redirects and Forwards
- OWASP top 10 Labs
    - Vulnerability identification
    - Vulnerability exploitation
    - Vulnerability fix
    - Using WebGoat and Zed Attack Proxy
- Cryptographic Risks
    - The Problem With Passwords
        - Using Weak Passwords
        - Password Iteration
        - Default Passwords
        - Password Replay Attacks
        - Stop Storing Plaintext Password
        - Rainbow Tables Explained
        - Too Much Information - Invalid User or Password
    - The Problem With Random Numbers
    - PRNG, CRNG and TRNG
    - Find Code That Use Incorrect RNG
    - Determine Properly Seeded CRNG
- The Problem With Crypto Algorithms
    - Roll Your Own Algorithm
    - Using The Wrong Algorithm
    - Forgetting The Salt
    - The Difference Between Authentication, Encryption and Temper-Proofing
    - Algorithms Are Not Future-Proof
- Network Protocols Security
    - The 5/7 Layers Models
    - Network Traffic Risks

        - Eavesdropping
        - Replay
        - Spoofing
        - Tempering
        - Hijacking
        - Network Vulnerabilities
        - ARP Poisoning
        - Man In The Middle
        - (D)DoS Attacks
    - Network Authentication and Protocols
        - Kerberos / NTLM
        - SSL and HTTPS
    - Further traffic sniffing: Wireshark, Charles Proxy, Burp Suite and ZAP.
- Trusted Execution Environments
    - Motivation and definitions of Secure vs. Normal worlds
    - Terminology: TPM, TEE, SE
    - Use cases
    - Introduction to ARM TrustZone
    - Secure World OS implementations
- Android Overview - Design considerations
    - Android History
    - The android ecosystem: Partners, Entities, Design, Approach, Licensing.
- Android Overview - Bottom up discussion
    - Hardware overview: What makes an Android device.
    - Linux Kernel boot process and provided functionalities
    - Native User Space: Init services, daemons, executables and libraries
    - Enabling Java (Dalvik + ART)
    - JNI bridge layer
    - Java OS Layer (Android Frameworks)
    - Application (APK) Structure
    - System Applications
    - User Applications
    - Google Play Services
    - Android IPC terminology by example: Browser, Maps.
    - Introduction to working with the AOSP: How and where to find what.
- Android Platform Security
    - Linux driven security sandbox
    - OS and binary protection and exploitation: ASLR, PIE, DEP, RoP et. al.
    - Android hardware related permission enforcement
    - SELinux on Android
    - Data partition forensics protection via Internal and external storage encryption
    - Secure Boot
    - Android Signature model and verification:
        - Platform keys and platform app signing. Google, OEM’s and integrators.
        - Third party (and play store) application signing.

    - Android application sandbox: Single and multi physical user.
    - Android Permissions:
        - Pre-Marshmallow (API Level < 23)
        - Post-Marshmallow: User policies, user responsibilities, application developer responsibilities, dynamic permission checking and revocation.
        - Defining custom permissions, restricting Application components (Activity, Service, Content Provider, Broadcast Receiver)
    - Android Security Patches
    - Android Nougat additions
    - Android Nougat native linker changes
    - Android Oreo additions
    - Android Oreo and Project Treble HAL and Kernel restrictions
    - Android Linux Kernel Hardening Features
    - Secure Boot cont. - Android Verified Boot v2
    - Android P expected additions
- Security terminology and real-life attacks, “breaking Android”:
    - Glossary attack vectors, attack surfaces, vulnerabilities and exploits.
    - Privilege escalation attacks - theory and practice
    - Dynamic code loading attacks and mitigation
        - Native code
        - Java code via DexLoader
        - Live (on device) code scanning techniques using the PackageManager
    - Binary exploitation and device rooting
    - Remote exploitation and DoS attacks
    - Signature based attacks
    - SE Linux discussion
    - On device Anti-Virus and Anti-Malware building techniques
- Penetration Testing and Dynamic Analysis
    - Android “debugging”: Introducing am, pm, wm, service, procfs, sysfs and friends.
    - Android Penetration testing tools
    - Finding exposed application components
    - Android fuzzing tools by example: fuzzing the Stagefright framework
    - Penetration testing and exploitation with drozer/metasploit
    - Project Treble HAL/Kernel interface fuzzing
- Reverse-Engineering Applications and Static Analysis
    - Android application installation process, paths, optimized bytecodes, ELF types
    - Dalvik bytecode structure and ART binary format
    - Decompiling/disassembling ART and Dalvik based files.
    - Rejoining and decompiling /disassembling optimized byte code.
    - Unpacking APK resources, repacking, resigning.
    - Disassembling vs. Decompiling: Tools and strategies: where to spend your time?
    - Survey of open source and commercial tools and analyzers.
    - Off device Anti-Virus and Anti-Malware building techniques
- Android Application Secure Coding I: Code and app behavior

    - Code protection techniques: Obfuscation, stripping, encryption, anti-tampering techniques. Native code techniques with NDK, gcc, and clang.
    - SQL Injection and protection from it.
    - Manifest level component access control
    - SELinux and Middleware MAC
    - IPC level runtime component access control
    - Webview and Javascript protection/restriction best practices for hybrid apps
    - Protecting from other applications, protecting from user judgement
    - Dynamic loading attack prevention (DEX, .so and .js)
    - Dynamic permission control best practices
    - Introduction to Android cryptography: BouncyCastle, BoringSSL
    - Protecting WebView code
    - Security Provider live-patching using ProviderInstaller
    - Static Analysis Checklist - using the Android lint tool (and other commercial tools)
- Android Application Secure Coding II: Securing User and Application data.
    - Android Storage layout - what's open and what's not.
    - SQLite inspection and protection with CQLCipher
    - Introduction to applied cryptography
        - Cryptography goals: Authentication, Integrity, Encryption.
        - Symmetric and Asymmetric cipher suites
        - Key generation techniques and trade-offs
        - Software vs. Hardware based techniques.
    - Android Applied cryptography
        - Protection models (Encryption vs. Authentication)
        - Software based protection via software based cryptography
        - Hardware based protection via the keystore
        - Hardware based authentication via Fingerprint API
        - Timed authentication via gatekeeper
        - Data encryption - protection and optimization.
- Android Application Secure Coding III: Secure Network Communications
    - Network privacy dangers: Packet sniffers and interceptors. MITM attacks.
    - Certificate Authority (CA) Chain of trust: A solution and the introduced problems
    - Secure communication with TLS/SSL
    - Encrypted network privacy dangers: Sniffers and interceptors. MITM attacks.
    - CA management in Android: Platform and application management
    - Custom TrustManager's and Certificate pinning
    - IP layer security teaser, VPN (more in the Android For Work section)
    - Clear-Text opting out and TLS enforcing
    - Network Security Configuration (Nougat, Oreo, +)
- Enterprise Mobility Management: Android Enterprise (formerly: Android for Work)
    - Enterprise Mobility Management (EMM) - definition and market survey
    - EMM: The IT manager vs. the private user
    - Device administration APIs - an IT manager biased arsenal
    - Work profiles - the compromise between the IT and the user.
    - Application restrictions

- Device provisioning: Apps, networks, etc.
- Per platform and Per app Virtual Private Networks (VPNs)