

# **Live Android Bug Bounty**

## **Training for InfoSec Startups**

**Online Training** 24x7 Availability



**email:** info@thehacktivists.in

Contact Anytime Call | WhatsApp : +91 96809 81337

**Advanced Android Bug Bounty**

**For InfoSec Startup | Security Researchers**

**#live #training**

**Training Fee: 6000 inr | 100 usd**

# The Hacktivists

Information Security Training Providing Company

**Call us : 96809 81337**



# Syllabus: Android Bug Bounty Training for InfoSec Startups

Requisite: Android Application Penetration Testing | Mobile OWASP TOP 10

Level: Intermediate -> Advanced

**Fee : 6000 INR | 100 USD**

**Training Level: Live Android Applications Bug Bounty**

## Why you Join us :

01. 100% #bugs #Practical on Android Applications | Challenges
02. Covered Bugs & Vulnerabilities with two-time practicals
03. Real-time Challenges with every Training Modules
04. Customized Training also provide for InfoSec Employer
05. Cover each Bugs according to Bug Bounty Platforms like HackerOne and Bugcrowd

## Introduction Android Bug Bounty Approach

- Android Penetration Testing Methodologies - Detailed Explanation
- Android Bug Bounty Methodology according to Bug Hunting Platforms
- Differences between Android Pentesting and Android Bug Bounty Approach
- Traditional Android Penetration Testing Report - Test Cases
- Traditional Android Penetration Testing Approach and Guidelines
- Android Application Attack Surface - Client Side Vulnerabilities
- Android Application Attack Surface - Server Side vulnerabilities
- Android Application Attack Surface - Logical Security Threats

## Setup AndroidApp Security Testing lab Environment

- Install Android Ptest Operating System
- Genymotion Android Emulator Installation
- Installing Android App components (GSuite)
- Installing Android App components ARM Translator
- An Overview of the Android Architecture
- An Overview of the Application Framework
- An Overview of the Android Permissions Model

## Primary Stage to Security Analysis of an Android Applications

- Setup Android Debug Bridge Utility (adb)
- Android Debug Bridge (adb) Pentester Utilities
- Vulnerable Android Application Source Code Analysis
- Understanding Android Application Source Code Compilation Process
- Structure of an Android Application Package (APK)
- Unzipping and Unpacking Android Applications
- Reversing an Android application using dex2jar
- Reversing an Android application using apktools

## Code Quality and Build Settings of Android Apps

- Android Application Manifest Overview
- Security Review of Manifest Elements
- Security Analysis of Manifest Elements

## Tampering Android Application and Security Analysis

- Signing an Android Applications Manually
- Android Code Obfuscation & Code Protection
- Adding Malicious Code to Android Apps
- Debugging Detection
- Root Detection
- VM Detection

## Security Analysis of Android Source Code

- Steps for Static Source Code Analysis
- Searching Vulnerable Functions in Source Code
- Steps for Dynamic Security Analysis of Application
- Dynamic Security Analysis using Drozer Security Testing Framework

## Security Testing - Insufficient Transport Layer Protection

- Dynamic Security Analysis using BurpSuite
- An Introduction and Installation of Xposed Framework
- Android SSL Verification and Certificate Pinning
- Bypass SSL Pinning to Perform Active Man-in-the-Middle

## Insecure Connection and Untrusted Connection Security Issues

- Use of Insecure Network Protocols
- Authentication over Insecure Protocols
- Data Transmission over Insecure Protocols

## Insecure Logging Security Issues

- Insecure Logging - Verbose Error Logging
- Insecure Logging - Authentication Token Leakage
- Insecure Logging - Sensitive Information Disclosure
- Insecure Logging - Personally identifiable information (PII)

## Insecure Sensitive Hardcoding Issues

- Insecure Hardcoding - API Keys Leakage
- Insecure Hardcoding - Authentication Token
- Insecure Hardcoding - Internal IP Disclosure
- Insecure Hardcoding - Git Repository Disclosure
- Insecure Hardcoding - Embedded Third-Party Secrets
- Insecure Hardcoding - Sensitive Information Disclosure

## Confidential Information Exposure By Design (Side Channel Data Leakage)

- Confidential Information Leakage - Insecure Backup Storage
- Confidential Information Leakage - Screen Capture on Personal Data
- Confidential Information Leakage - Application Level Denial-of-Service (DoS)
- Confidential Information Leakage - Leaks of Personal Data using Virtual Keyboard
- Confidential Information Leakage - Exposure of Sensitive Data Copied to Clipboard
- Confidential Information Leakage - Sensitive Data Disclosure Through UserInterface
- Confidential Information Leakage - Cleartext Storage of Sensitive data in Memory

## Security Issues in OAuth Implementations

- Leaking OAuth Tokens - Android logcat
- Leaking OAuth Tokens - Shared Preferences
- Leaking OAuth Tokens - OAuth HardCoded Secret Tokens
- Leaking OAuth Tokens - Inadequate transmission protection

## Insecure Cryptographic Storage

- Insecure Cryptographic Storage - SQLite Databases
- Insecure Cryptographic Storage - Internal Storage
- Insecure Cryptographic Storage - External Storage
- Insecure Cryptographic Storage - Shared Preferences

## Unprotected Application Components

- Unprotected Application Components - Unprotected Services
- Unprotected Application Components - Unprotected Activities
- Unprotected Application Components - Leaking Content Providers
- Unprotected Application Components - Typos in Custom Permissions
- Unprotected Application Components - Implicit Broadcasts (Sending)
- Unprotected Application Components - Implicit Broadcasts (Receiving)
- Unprotected Application Components - Android Fragment Injection
- Unprotected Application Components - Allowing Manipulation Unprotected Activities

## Private File Access Security Issues

- Private File Access - Local File Inclusion
- Private File Access - Remote Command Execution Vulnerability
- Private File Access - Private Data Overwrite due to Path Traversal
- Private File Access - Private Data Overwrite due to ZIP File Traversal

## Testing Code Quality and Injection Flaws

- Injection Flaws - SQL Injection
- Injection Flaws - HTML Injection
- Injection Flaws - Cross Site Scripting
- Injection Flaws - Improper Markup Sanitisation
- Injection Flaws - Crash App & DoS using other app
- Injection Flaws - Insecure Deeplink leads to Sensitive Information Disclosure

## Security Analysis of API Endpoints with Telerik Fiddler

- Composing Application API Calls - Functional API
- Capturing Application API Calls - Functional API
- Filtering Application Request Traffic

- Analyzing the Authentication Endpoints
- Analyzing an Additional API Call
- Analyzing Sensitive Data Disclosure in API Endpoint

## Insufficient Anti Automation

- Insufficient Anti Automation - Registration
- Insufficient Anti Automation - Login (static)
- Insufficient Anti Automation - Password Reset Function

## Insecure Authentication and Authorization

- Bypass One Time Verification Codes
- OTP SMS or Voice Code Leaked in Response
- Bypass Second Factor Authentication (2FA)

## Improper Access Control

- Improper Access Control
- Insufficient Entropy For Random Values
- Personally Identifiable Information (PII) Disclosure

## Server Side Vulnerabilities

- Improper Session Handling
- Leakage of API Auth Tokens
- Improper Restriction of Misconfigured API
- Improper Restriction of Unprotected APIs Endpoint
- Transporting API Auth tokens as Cleartext Allowed

## Beware of Recent Android Vulnerabilities

- Tapjacking Vulnerability
- Remote Wipe Vulnerability
- AAPT Time Zone Disclosure Bug
- Android Master Key vulnerability
- Address Bar Spoofing Vulnerability

**\*Contact us :**

---

- Need technical assistance? Speak with a support representative by calling **+91-9680-981-337**

“

---