COURSE TITLE	**ADVANCE MOBILE APPLICATION SECURITY**
COURSE CODE	**01CY0221**
COURSE CREDITS	**4**

Objective:

1 This course equips students with advanced skills in mobile application security, focusing on the inner workings of Android and iOS architectures and how attackers exploit their vulnerabilities. Learners will gain hands-on experience in decompiling and reverse engineering APK/IPA files, dynamically manipulating apps using tools like Frida and Objection, and uncovering flaws in authentication, APIs, and cloud integrations. The course also explores malware simulation and analysis, including C2 behaviors and MITRE ATT&CK Mobile mapping. Additionally, students will learn to extract sensitive data, identify cryptographic flaws, and bypass hardware-bound security mechanisms such as KeyStore and Secure Enclave.

Course Outcomes: After completion of this course, student will be able to:

1 Define secure mobile operating system architectures (Android/iOS) and identify mobile attack surfaces; explain threat modeling frameworks (STRIDE, DREAD).

2 Illustrate mobile reverse engineering workflows and apply decompilation, static/dynamic code analysis using tools like Frida, Ghidra, Hopper, and JADX

3 Analyze and bypass runtime protections including SSL pinning, root detection, and biometric checks using instrumentation frameworks such as Frida and Objection

4 Examine API-level and cloud misconfiguration vulnerabilities in mobile apps; simulate attacks on Firebase, S3 buckets, JWTs, and insecure CI/CD artifacts.

5 Conduct mobile malware analysis and red team simulations; map threats to MITRE ATT&CK and develop mitigation strategies.

6 Evaluate cryptographic implementation flaws and storage vulnerabilities; justify secure hardware integration methods and propose improvements.

Pre-requisite of course:1. Basic Mobile Application Development (Android/iOS) 2. Foundations of Cybersecurity and Secure Coding 3. Linux Command Line & Scripting Fundamentals

Teaching and Examination Scheme

Theory Hours	Tutorial Hours	Practical Hours	ESE	IA	CSE	Viva	Term Work
3	0	2	50	30	20	25	25

DR. KRUNAL NARUMAL VAGHELA
Digitally signed by (Name of HOD)

DR. RAJENDRASINH BAHADURSINH JADEJA
Digitally signed by (Name of Dean/ Principal)

Contents : Unit	Topics	Contact Hours
1	**Advanced Mobile Architecture, Threat Modeling& Reverse Engineering** Android and iOS secure architecture internals, Android ART/Dalvik, Zygote process, Binder IPC , iOS Secure Enclave, sandboxing, entitlements, Mobile app attack surface mapping & threat modeling (STRIDE/DREAD/PASTA), Deep dive into reverse engineering techniques , APK/IPA decompilation, Smali/ARM assembly, Analyzing obfuscated code & identifying vulnerabilities, Native code analysis and NDK-level reversing, Disassembling .so libraries using Ghidra& IDA Free	8
2	**Runtime Attacks, Dynamic Instrumentation & Bypass Techniques** Frida & Objection deep dive for runtime instrumentation, Bypassing: Root/jailbreak detection, SSL pinning using Frida custom scripts, Emulator detection & anti-debugging mechanisms, Analyzing and exploiting dynamic code loading and native exploits, Heap spraying,, Java reflection misuse, DexClassLoader abuse	8
3	**Exploiting Mobile APIs, Cloud Integration & CI/CD Security** Advanced API attacks (real-world):Broken object level authorization (BOLA), mass assignment, rate-limiting bypass, JWT cracking, insecure deserialization, shadow APIs discovery, Cloud misconfiguration exploitation (Firebase, AWS S3, Azure Blob), CI/CD pipeline exploitation in mobile apps:Exposed .env, Firebase keys, build secrets in apps,, Abuse of GitHub Actions / Mobile DevOps (Bitrise CodeMagic), Mobile DevSecOps: securing CI pipelines with SAST/DAST in GitHub/GitLab	10
4	**Deep Storage Forensics, Cryptography Failures & Secure Hardware Integration** Bypassing encrypted SharedPreferences, internal storage, SQLite DBs, Dump and decrypt app secrets from device/emulator,, Exploiting KeyStore/Keychain weaknesses, Brute-forcing PINs, ingerprint bypass using Magisk modules, Crypto implementation flaws:ECB/CBC misuse, insecure random generation, Padding oracle attacks in mobile environments, Secure hardware integration bypasses (YubiKey, fingerprint, FIDO2)	7
5	**Red Team Mobile Simulation, Advanced Malware & Exploit Case Studies** Building a red team mobile kill chain, Initial access: malicious APK via phishing,Persistence:, Android Accessibility Services, hidden receivers, Exfiltration: Covert C2 channel using Firebase or Telegram bots, Mobile malware dissection:Analyze Pegasus-like samples for behavior and network traffic, Hook malware behavior in real-time using Frida,Mobile zero- day case studies:Stagefright, iOS Pegasus, Android Binder exploits, Reporting in industry format:MITRE ATT&CK Mobile mapping + remediation techniques	7
Total Hours		**40**

DR. KRUNAL NARUMAL VAGHELA
Digitally signed by (Name of HOD)

DR. RAJENDRASINH BAHADURSINH JADEJA
Digitally signed by (Name of Dean/ Principal)

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
1	**Practical 1** Reverse Engineering a Banking APK Decompile a real-world banking APK to identify hardcoded secrets, sensitive logic, and obfuscated code patterns. Tools: APKTool, MobSF, JADX, Bytecode Viewer	2
2	**Practical 2** Bypassing SSL Pinning with Frida Intercept and decrypt HTTPS traffic from an SSL-pinned app using Frida hooks to bypass certificate validation at runtime. Tools: Frida, Objection, Burp Suite Community Edition	2
3	**Practical 3** Runtime Hooking to Bypass Biometric Authentication Hook and modify biometric authentication results in real-time to simulate unauthorized access. Tools: Frida, Objection, ADB	2
4	**Practical 4** Firebase Misconfiguration Exploitation Identify misconfigured Firebase backends in APKs and exploit insecure read/write permissions to extract user data. Tools: MobSF, Firebase Scanner, APKLeaks	2
5	**Practical 5** Extracting Secrets from iOS IPA Files Analyze an iOS app's IPA to retrieve embedded keys, insecure configurations, and logic flaws. Tools: Hopper Disassembler, Frida, Class-dump, Cycript	2
6	**Practical 6** Analyzing and Modifying Native Libraries (.so Files) Disassemble and analyze native .so binaries used in Android apps to uncover hidden logic and cryptographic functions. Tools: Ghidra, IDA Free, Radare2	2
7	**Practical 7** Insecure Local Storage & Database Decryption Access and decrypt sensitive data stored in SharedPreferences or SQLite with weak encryption. Tools: ADB, DB Browser for SQLite, Rooted Emulator, KeyStore Extractor	2
8	**Practical 8** API Security Testing and BOLA Exploitation Use Burp Suite to intercept mobile API requests and exploit Broken Object Level Authorization vulnerabilities. Tools: Burp Suite, Postman, JWT Tool, MobSF API Scanner	2
9	**Practical 9** CI/CD Secrets Extraction from APK Artifacts Identify exposed build secrets such as .env keys or Firebase configs in the APK that were leaked via CI/CD misconfigurations. Tools: APKLeaks, MobSF, GitLeaks, TruffleHog	2

Suggested List of Experiments:

Contents : Unit	Topics	Contact Hours
10	**Practical 10** Simulating Android Malware for Red Teaming Create a custom malicious APK that exfiltrates sensitive data using covert channels like Telegram or Firebase. Tools: APKTool, Termux, Metasploit, Python-Telegram-Bot	2
Total Hours		**20**

Textbook :

1 Android Hacker's Handbook, Joshua J. Drake, Zach Lanier, Collin Mulliner, et al., John Wiley & Sons, 2014

References:

1 Dominic Chell, Tyrone Erasmus, Shaun Colley, Ollie Whitehouse, Dominic Chell, Tyrone Erasmus, Shaun Colley, Ollie Whitehouse, Dominic Chell, Tyrone Erasmus, Shaun Colley, Ollie Whitehouse, John Wiley & Sons, 2015

2 Practical Mobile Forensics, Practical Mobile Forensics, Rohit Tamma, Heather Mahalik, Satish Bommisetty, Packt Publishing, 2016

Suggested Theory Distribution:

The suggested theory distribution as per Bloom's taxonomy is as follows. This distribution serves as guidelines for teachers and students to achieve effective teaching-learning process

Distribution of Theory for course delivery					
Remember / Knowledge	**Understand**	**Apply**	**Analyze**	**Evaluate**	**Higher order Thinking / Creative**
0.00	0.00	30.00	30.00	25.00	15.00

Instructional Method:

1 The course delivery method will depend upon the requirement of content and need of students. The teacher in addition to conventional teaching method by black board, may also use any of tools such as demonstration, role play, Quiz, brainstorming, MOOCs etc.

2 The internal evaluation will be done on the basis of continuous evaluation of students in the laboratory and class-room.

3 Practical examination will be conducted at the end of semester for evaluation of performance of students in laboratory.

4 Students will use supplementary resources such as online videos, NPTEL videos, e-courses, Virtual Laboratory.

Supplementary Resources:

1 OWASP Mobile Security Testing Guide (MSTG) https://owasp.org/www-project-mobile-security-testing-guide/

DR. KRUNAL NARUMAL VAGHELA
Digitally signed by (Name of HOD)

DR. RAJENDRASINH BAHADURSINH JADEJA
Digitally signed by (Name of Dean/ Principal)

Supplementary Resources:

2 Frida: Dynamic Instrumentation Toolkit https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/

DR. KRUNAL NARUMAL VAGHELA
Digitally signed by (Name of HOD)

DR. RAJENDRASINH BAHADURSINH JADEJA
Digitally signed by (Name of Dean/ Principal)