Analisi delle vulnerabilità di un’applicazione per il trasporto pubblico

CANDIDATO
Domenico Femia

RELATORE
Prof. Gianluigi Folino

ANNO ACCADEMICO
2025/2026

Struttura della Presentazione

1. **Contesto e Motivazioni**
   Obiettivi, metodologia e sistema analizzato
2. **Tecnologie e Architettura**
   Stack tecnologico frontend/backend e infrastruttura
3. **Metodologia di Analisi**
   Black box, analisi statica e dinamica, OWASP MASVS
4. **Vulnerabilità Lato Server**
   API REST, autenticazione, configurazione insicura
5. **Vulnerabilità Lato Client**
   Storage locale, sessioni, certificate pinning
6. **Compromissione dei Segreti**
   Path traversal, JWT, Firebase, SMTP
7. **Aspetti Organizzativi**
   Vulnerability Disclosure, ISO/IEC 29147 e 30111
8. **Conclusioni**
   Riepilogo risultati e raccomandazioni

Contesto e Motivazioni

> *Everything should be believed insecure until demonstrated otherwise.*
>
> — Bruce Schneier

Oggetto

App mobile per servizio di trasporto pubblico a chiamata (DRT)

Approccio

Analisi black box: nessun supporto interno. Simulazione di un attaccante esterno

Obiettivo

Valutare la sicurezza dell'intera infrastruttura: client mobile e backend REST API

Tecnologie e Architettura

FRONTEND

React Native
Framework cross-platform (JavaScript/TypeScript)

Hermes Engine
Bytecode AOT, offuscamento codice nativo

BACKEND

Django + DRF
Framework Python, API RESTful

JWT Auth
HS256, access 15min, refresh 31gg (Simple JWT)

Django Channels
WebSocket per posizione real-time (Daphne)

INFRASTRUTTURA

PostgreSQL + SQLite
Database relazionale principale e failover

Apache HTTP + Plesk
Reverse proxy e pannello di controllo

Firebase FCM + Valhalla
Notifiche push e routing geografico OSM

Metodologia di Analisi

1

Analisi Statica

Decompilazione dell'APK tramite APKTool e Hermes dec, con analisi statica per l'estrazione di endpoint, stringhe e dipendenze.

2

Analisi Dinamica

Intercettazione e analisi del traffico di rete dell'applicazione tramite mitmproxy, con supporto di Android Debug Bridge e tecniche di dynamic instrumentation mediante Frida per il bypass del certificate pinning su dispositivo Android.

3

Test API

Analisi e manipolazione delle richieste API tramite Postman e script custom, con tecniche di fuzzing e alterazione dei parametri mediante ffuf.

4

OWASP MASVS

Riferimento allo standard Mobile Application Security Verification Standard (MASVS) e alla tassonomia delle vulnerabilità CVE.

Vulnerabilità Lato Server — Controlli & Logica

Bypass Registrazione Email

A04:2021 Insecure Design

Nessuna validazione del dominio lato server. Chiunque può creare un account con qualsiasi email.

IDOR Prenotazioni

IDOR Read/Write

L'endpoint /trip/[id] non verifica che l'id appartenga all'utente. Incremento sequenziale espone tutti i dati.

Mancato Controllo Transizioni di Stato

Broken Access Control

Qualsiasi utente può modificare lo stato di qualsiasi prenotazione (init→completed) senza essere autista.

Settings Modificabili da Tutti

Privilege Escalation

L'endpoint PUT /settings non verifica il ruolo. Chiunque può modificare geo-fence, orari e max passeggeri.

Mancata Rotazione Refresh Token

OWASP Session Mgmt

Il refresh token (31gg) non viene invalidato al rinnovo. Accesso fisico al device → sessione 31 giorni.

WebSocket Senza Autenticazione

Missing Auth

Canale /ws/driver/[id] accessibile senza token. Chiunque può inviare posizioni false a tutti i client.

Compromissione dei Segreti Applicativi

CATENA DI ATTACCO

Debug Mode

DRF in debug
Stack trace pubblica

→

Path Traversal

Double-dot (..) non filtrato
Esfiltrazione sorgenti

→

JWT Secret

Chiave HS256 da settings.py
Token per qualsiasi user

→

Admin Access

Token user id=1 → privilegi
Lista utenti completa

IMPATTO

JWT Secret
Impersonificazione di qualsiasi utente/autista

Firebase Token
Spam di notifiche push a tutti i dispositivi

Credenziali DB
Accesso al database (limitato per IP)

Credenziali SMTP
Invio email con dominio legittimo (phishing)

Vulnerabilità Lato Client

⚠ PROBLEMATICHE

Insecure Local Storage
MASWE-0006

Token JWT, nome, email salvati in chiaro nel file mmkv.default senza cifratura (Android Keystore non utilizzato)

Logout Solo Locale

Il logout cancella il file locale ma NON invalida il token sul backend. Il token rimane valido fino a scadenza.

✓ BUONE PRATICHE

HTTPS Ovunque

OkHttp forza HTTPS per tutte le comunicazioni

Certificate Pinning

Blocca attacchi MitM: solo cert. specifici accettati

Offuscamento Hermes

Bytecode AOT rende difficile il reverse del codice

No Backup

Flag android:allowBackup="false" impostata

Vulnerability Disclosure & Aspetti Organizzativi

ISO/IEC 29147 — Vulnerability Disclosure

- Canale dedicato per le segnalazioni
- Chiave PGP per comunicazioni cifrate
- File security.txt (RFC 9116)
- Policy di divulgazione pubblica
- Programma bug bounty

ISO/IEC 30111 — Vulnerability Handling

- Triage: gravità e validità
- Investigazione tecnica approfondita
- Sviluppo della correzione (patch)
- Distribuzione e comunicazione
- Tracciabilità e documentazione

Gap Rilevato nell'Organizzazione

Nessun canale dedicato (solo info@dominio.srl generico) · Nessuna sezione security sul sito · Assenza di security.txt · Dinamiche di disclosure non formalizzate preventivamente a causa della mancanza di procedure interne.

Riepilogo delle Vulnerabilità

4	5	3	12
CRITICA	ALTA	MEDIA	TOTALE

Vulnerabilità	Categoria OWASP/CWE	Severity
Bypass registrazione email	A04:2021 Insecure Design	Alta
IDOR Prenotazioni R/W	A01:2021 Broken Access Control	Critica
Path Traversal Backend	A05:2021 Security Misconfiguration	Critica
JWT Secret Hardcoded	A02:2021 Crypto Failure	Critica
Django Debug in Produzione	A05:2021 Security Misconfiguration	Critica
Insecure Local Storage	MASWE-0006	Alta
WebSocket Senza Auth	A07:2021 Auth Failure	Alta
Mancata Rotazione Token	OWASP Session Mgmt	Alta
Logout Solo Locale	MASWE	Media

Conclusioni e Raccomandazioni

1

Security by Design

Adottare validazione server-side per tutti i vincoli (dominio email, orari, geo-fence, ruoli)

2

Correzione Configurazioni

Disattivare Django DEBUG in produzione, proteggere Swagger/Redoc, aggiungere rate limiting

3

Gestione Segreti

Rimuovere credenziali hardcoded, usare variabili d'ambiente o secret manager dedicato

4

Cifratura Locale

Implementare Android Keystore / iOS Keychain per token e dati sensibili nel client

5

Invalidazione Token

Implementare rotazione refresh token e invalidazione server-side al logout

6

Vulnerability Disclosure

Istituire canale dedicato, pubblicare security.txt, adottare ISO/IEC 29147 e 30111

Grazie dell’attenzione!

CANDIDATO
Domenico Femia

RELATORE
Prof. Gianluigi Folino

ANNO ACCADEMICO
2025/2026