Slide 0 - Title

Good morning, my name is Christine Midtgaard, and today I will present a security analysis of the FitnessX mobile application for Android.

Slide 1 - Introduction

The motivation behind this project is the growing reliance on mobile apps to replace physical infrastructure such as keycards. FitnessX is a prime example, having transitioned entirely to mobile-based access. Given the app's access to sensitive data, like location, personal identifiers, and financial transactions, it's important to evaluate its security posture.
This is a review left on Google Play from a user of the app. He says: "Just let us have damn cards to get in. The app is trash is I don't let it have my location at all time. I can count on 3 hands how many times I couldn't get into the center."
And there are many more of those reviews.

Slide 2 - App Feature

The FitnessX app offers more features that just unlocking physical access to centers. It enables class bookings and cancellations, manages membership and payments, synchronizes with calendars, and stores personal data including names, photos, address, email, phone number, and date of birth. A notable feature is the 2-for-1 membership, which extends access to a 'training buddy' whose data is also processed by the app.
This leads us to the threat model.

Slide 3 - Threat Model

First of all, adversaries include individuals and organizations aiming to steal identities or access credentials. Third-party services that may mishandle data. Insider threats, meaning employees of FitnessX. The attack surface includes: the app itself, network traffic, authentication mechanisms, and third-party integrations.

Slide 4 - Security Properties

In evaluating the FitnessX app, I considered the security properties: confidentiality, integrity, non-repudiation, reliability, authenticity, privacy, and availability. These criteria formed the basis for assessing the app's resistance to attacks and its handling of sensitive user data. After going through the findings of the security analysis, I will discuss them in terms of these properties.

Slide 5 - Domains of Analysis

I evaluate the app across the four domains:

- Software security, where I go through the app's code to identify weaknesses.
- Network security, where I investigate how the app communicates with servers.
- Authentication, where I evaluate how users verify their identity.
- Privacy, where I look into how the app handles user data, and if it complies with the privacy policy.

I used publicly available tools for the analyzation. We beging with the software security.

Slide 6 - Software Security: Goals and Tools

The goals of the software security analysis is to examine the code for vulnerabilities. This is done by decompiling the APK files, search for keywords, and use scan the code with a static tool. For this I used APKPure to get the latest version of the FitnessX app. I decompiled them with JADX, and searched for keywords. I used MobSF for static analysis, which gives a report.

Slides 7 - 8 - Software Security: No Obfuscation

First of all, when decompiling the code, it showed no signs of obfuscation. This makes it much easier to analyze the app.

Slides 9 - 10 - Software Security: Hardcoded Client Secret

I found a hardcoded client secret in the BuildConfig class, used for backend authentication. This secret can be extracted by attackers to impersonate the app and perform unauthorized API requests.

Slides 11 - Software Security: Airship Secret

In addition to the client secret, I found hardcoded API credentials for the Airship service, which is likely used for push notifications. Unauthorized access to this key could allow attackers to send spoofed notifications or harvest analytics.

Slides 12 - 13 - Software Security: MobSF Results

MobSF scored the app 51 out of 100, identifying two high-severity and seventeen medium vulnerabilities. High-severity issues include accepting user-installed certificates and allowing cleartext traffic. The MobSF results also revieled some warnings including permissions. I will discuss these later. Although the app primarily uses HTTPS, as seen later, the fact that it allows cleartext traffic and accepts user-installed certificates opens the door for man-in-the-middle attacks, which leads us to testing the security of the network communication.

Slide 14 - Network Security Goals and Tools

The goals of the network security analysis is to document the TLS server configurations of the servers, the app connects to. Attemt a MITM attack to see if I can sniff traffic. For this setup i used an Andorid emulator running Android 16.0 and the newest version of the FitnessX app downloaded from Google Play. I monitored traffic with Wireshark and evaluated TLS configurations using Qualys SSL Labs. I also attempted man-in-the-middle attacks using mitmproxy.

Slide 15 - TLS Configuration Summary

Three backend servers supported outdated TLS 1.0 and 1.1. While some servers, like Reepay and Azure, achieved A+ grades, the main app backend scored only a B due to its support for weak protocols.

Slide 16 - MITM Overview

the next thing is to attempt a MITM attack. We know that the app allows user-installed root certificates and cleartext traffic, which raises the likelihood of successful MITM attacks.

Slide 17 - Initial MITM Failure

The first MITM attempt failed due to Android's default rejection of untrusted certificates.

Slide 18 - 19 - Successful MITM Attack

After installing mitmproxy's certificate on the emulator, I was able to successfully intercept traffic. The login credentials and payment data were exposed in plaintext over HTTPS, revealing a critical breach of confidentiality.

Slide 20 - Authentication: Goals and Tools

for the authentication analysis the goals are to document the authentication mechanisms and evaluate their security. This was done using the same setup as before with the app installed on the emulator and mitmproxy. I looked for hardcoded credentials, token handling, and brute-force protections.

Slide 21 - Authentication Findings

The app uses single-factor login with minimal password complexity. Passwords are exposed in plaintext during login and reset, as we saw earlier in the successful MITM attack. Additionally, both the code and the intercepted traffic revealed that the hardcoded client secret was used in the authentication flow. Reset emails deliver new passwords in plaintext without expiration or forced change, making accounts vulnerable to takeover. However, logging out revealed that token revocation works.

Slide 22 - Token Revocation

Token revocation works correctly upon logout, returning 401 Unauthorized on reuse. This is a positive finding that mitigates session hijacking risks.

Slide 23 - Privacy: Goals and Tools

The goals of the privacy analysis are to examine how the application collects and handles data, and ensure its compliance with the data protection principles. For this, I reviewed the app's AndroidManifest, used Exodus Privacy to identify trackers, and compared permissions against the stated privacy policy.

Slide 24 - Permissions

The MobSF revealed that the app requests sensitive permissions such as location, calendar access, camera and Bluetooth. These are not mentioned in the privacy policy, indicating poor transparency. Testing the functionality of the app with Bluetooth turned off revelaed that the app was still fully functional, indicating that it doesn't use it. This is an unnessecary increase of attack surface.

Slide 25 - Detected Trackers

Four trackers were found: Google Firebase Analytics, Crashlytics, Airship, and Alt-Beacon. Firebase analytics and Crashlytics collects usage data. Airship enables push notifications. Again, the AltBeacon library enables proximity detection through BLE beacons, which may be a remnant from older versions of the app.

Slide 26 - Airship Credentials

As mentioned earlier, hardcoded Airship credentials could allow attackers to impersonate the app or compromise user privacy via unauthorized notifications.

Slide 27 - Privacy Policy Gaps

The privacy policy has not been updated since 2021, and fails to mention the Android version of the app, released in 2023. There are mismatches between stated and actual data handling practices in the privacy policy. Both App Store and Google Play mention tata collection, and what is attached to the user.

Slide 28 - Summary of Vulnerabilities

So, to summarize: the software security analysis revealed hardcoded secrets, lack of encryption, which I haven't talked about a lot here, and the app allows user-installed certificates and cleartext. The network security analysis yielded a successful MITM attack, where credentials were visible in plaintext. The authentication analysis revealed insecure password handling and reset policies. The privacy analysis revealed undisclosed trackers, and an outdated and incomplete privacy policy.

Slides 29 - 30 - Discussion: Security Properties

In terms of security properties:
Confidentiality: The app transmits sensitive user information—including login credentials and payment details—in plaintext during a successful man-in-the-middle (MITM) attack. Lack of certificate or public key pinning, combined with accepting user-installed root certificates and cleartext traffic, allows adversaries to intercept this information.
Integrity: Hardcoded credentials and poor network security open the door for adversaries to impersonate the app and possibly alter backend requests or responses. This compromises the integrity of data such as visit logs, bookings, and membership changes
Authenticity: The absence of certificate pinning and the ability to impersonate the backend API through hardcoded secrets mean that unauthorized actors could access or submit data as if they were the user. The authentication process lacks robustness to prevent this.
Non-repudiation: The app's weak authentication setup (single-factor login with lenient password rules and no login attempt throttling) means users could plausibly deny having performed actions. There is no cryptographic evidence binding actions to the user.
Privacy: The app collects personal, payment, and behavioral data but fails to transparently disclose all uses of this data in the privacy policy. Undocumented trackers (e.g. Firebase, Airship, AltBeacon) and excessive permissions (Bluetooth, calendar access) further exacerbate privacy concerns.
Reliability: The app demonstrates weak handling of sensitive actions, such as sending a new password in plaintext email with no forced password change. This undermines user trust and data handling reliability. Furthermore, misuse of push notifications (Airship with hardcoded keys) could lead to spoofed notifications or app misuse.
Availability: While no specific incidents of downtime were reported, reliance on a vulnerable app for physical access (without fallback options) and the risk of account lockouts or hijacking due to weak session handling could impair availability during critical moments

Slide 31 - Conclusion and Recommendation

In conclusion: users are at risk of unauhtorized access, crential theft, and personal data leaks. In conclusion, the FitnessX app cannot be considered secure for handling sensitive user data. Making the app more secure could involve implementing public key pinning, removing hardcoded credentials, updating TLS settings, enforcing strong password policies, and revising the privacy policy for transparency and compliance.