Chapitre 1 : Contexte et état de l’art

1.1. Panorama de la sécurité mobile

La sécurité mobile est devenue un enjeu majeur en raison de la forte adoption des smartphones et de l’utilisation croissante des applications mobiles. Ces appareils contiennent une grande quantité de données personnelles et professionnelles, ce qui les rend attractifs pour les cyberattaquants. Les attaques ciblant les applications mobiles peuvent avoir des conséquences graves telles que le vol de données sensibles, l’espionnage, ou l’intrusion dans des systèmes d’information critiques. Il est donc crucial d’intégrer des mécanismes de sécurité dans le développement et l’utilisation des applications mobiles.

- **les Enjeux de la sécurité mobile**

Les enjeux de la sécurité mobile sont multiples et concernent plusieurs domaines :

- Confidentialité des données : Les applications doivent garantir que les données des utilisateurs (mot de passe, informations bancaires, etc.) sont correctement protégées.

- Intégrité des données : Les attaques visant l'intégrité des données peuvent altérer les informations stockées ou transmises, ce qui peut compromettre le bon fonctionnement de l'application.

- Disponibilité des services : Les attaques par déni de service peuvent rendre l'application ou l'appareil inutilisables.

- Authentification et autorisation : Une mauvaise gestion de l’authentification et des permissions peut entraîner des accès non autorisés à des données ou à des fonctionnalités sensibles

- **Spécificités du système Android**

Android, étant le système d’exploitation mobile le plus utilisé, présente une surface d’attaque étendue. Il repose sur le modèle d’application basé sur les APK (Android Package), qui contiennent tout le nécessaire pour l'exécution d'une application mobile. Certaines spécificités du système Android, telles que l'architecture en couches (comprenant le noyau Linux, la machine virtuelle Dalvik/ART, et les applications), peuvent rendre la détection et la gestion des vulnérabilités complexes. De plus, Android permet une grande flexibilité en termes de permissions et d’accès aux ressources, ce qui peut entraîner des failles de sécurité si mal gérées.

- **Risques liés aux applications mobiles**

Les applications Android sont souvent exposées à divers risques, parmi lesquels on trouve :

- Fuite de données sensibles : Des informations comme des mots de passe, des coordonnées bancaires, ou des données personnelles peuvent être exposées en raison de mauvaises pratiques de stockage ou de transmission.

- Espionnage : Certaines applications malveillantes ou piratées peuvent accéder à la caméra, au microphone, ou aux contacts de l’utilisateur sans son consentement explicite.

- Exécution de code malveillant : Des vulnérabilités dans le code des applications peuvent permettre à des attaquants d’exécuter des commandes ou d'injecter du code malveillant dans l’application, compromettant ainsi la sécurité de l’appareil.

1.2. Vulnérabilités courantes dans les applications Android

Les applications Android peuvent présenter diverses vulnérabilités, dont les plus courantes sont :

- **Permissions excessives**

  Les permissions sont une caractéristique fondamentale des applications Android. Toutefois, certaines applications demandent des permissions excessives qui ne sont pas nécessaires pour leur bon fonctionnement, ce qui expose l'utilisateur à des risques de fuite de données ou d'abus. Par exemple, une application de gestion de tâches demandant l'accès aux contacts ou à la localisation peut compromettre la confidentialité de l'utilisateur

- **Injections de code (JavaScript, SQL)**

  Les injections de code sont l’une des attaques les plus fréquentes. Elles se produisent lorsqu'un attaquant insère du code malveillant (SQL, JavaScript, etc.) dans une application via un champ de saisie ou une communication avec une base de données ou un serveur. Ces injections peuvent compromettre l'intégrité des données ou permettre un contrôle total de l'application

- **Mauvaise gestion des données sensibles**

  Le stockage et la transmission non sécurisés des données sensibles (comme les mots de passe, les informations personnelles, etc.) sont des vulnérabilités courantes. Par exemple, stocker des données sensibles en texte clair dans une base de données locale ou les envoyer via des protocoles non sécurisés (HTTP au lieu de HTTPS) expose l'application à des attaques

- Usage de protocoles non sécurisés

Certaines applications Android ne chiffrent pas correctement leurs communications réseau, utilisant des protocoles non sécurisés comme HTTP au lieu de HTTPS. Cela permet aux attaquants de réaliser des attaques de type "man-in-the-middle", où ils interceptent et modifient les données en transit entre l'application et le serveur.

- **Stockage non sécurisé**

  De nombreuses applications mobiles utilisent des moyens de stockage non sécurisés pour stocker des informations critiques. Cela peut inclure des fichiers sur le stockage interne ou externe de l'appareil qui ne sont pas cryptés, rendant ces informations vulnérables à une lecture non autorisée en cas de vol ou de compromission de l’appareil

1.3. Approches existantes pour la détection des vulnérabilités

- **Analyse statique : principes et limites**

  L’analyse statique consiste à examiner le code source ou les APK sans les exécuter. Elle permet d'identifier des vulnérabilités telles que des permissions excessives, des pratiques de codage risquées, ou des mauvaises configurations. Les outils d'analyse statique comme MobSF, Androguard, et JADX analysent le code pour en extraire les informations pertinentes. Cependant, cette approche présente des limites, notamment en ce qui concerne l'analyse des comportements dynamiques de l'application, tels que les interactions réseau ou les manipulations d'API

- **Analyse dynamique : avantages et inconvénients**

  L’analyse dynamique permet d’observer le comportement d’une application pendant son exécution, ce qui permet de détecter des vulnérabilités telles que les fuites de données via des canaux réseau ou le stockage non sécurisé. Des outils comme Frida, Burp Suite et Wireshark sont utilisés pour intercepter et analyser les appels réseau, les accès aux bases de données, ou d’autres comportements d'exécution. Cette approche est plus réaliste mais nécessite des ressources pour configurer un environnement d'exécution sécurisé et simuler les interactions de l'utilisateur

  - **Outils existants**

    l existe plusieurs outils permettant d'effectuer des analyses statiques et dynamiques des applications Android. Parmi eux

- **MobSF** : Un framework de sécurité pour les applications Android et iOS permettant d'effectuer des analyses statiques et dynamiques
- **AndroBugs** : Un outil d’analyse statique des APK pour détecter des vulnérabilités courantes
- **QARK** : Un outil d'analyse statique permettant de détecter les vulnérabilités dans les applications Android.
- **Frida** : Un framework dynamique pour l'instrumentation en temps réel des applications et la manipulation de leur comportement.
- **Burp Suite et Wireshark** : Des outils utilisés pour l'analyse dynamique des communications réseau des applications.