

ЗАХИСТ FLUTTER-ДОДАТКІВ ВІД ЗВОРОТНОГО ІНЖИНІРИНГУ

Вінницький національний технічний університет

Анотація

В роботі розглянуто основні методи захисту Flutter-додатків від зворотного інжинірингу. Проаналізовано основні загрози, такі як декомпіляція Dart-коду, ін'єкція шкідливого коду, витік API-ключів та обхід механізмів захисту. Розглянуто ефективні методи протидії, включаючи офускацію коду, шифрування конфіденційних даних, перевірку цілісності додатку та виявлення дебагерів.

Ключові слова: Flutter, безпека мобільних додатків, зворотний інжиніринг, офускація, шифрування, антидебаг.

Abstract

This work examines the main protection methods for Flutter applications against reverse engineering. It analyzes key threats such as Dart code decompilation, malicious code injection, API key leakage, and security bypass techniques. Effective countermeasures are reviewed, including code obfuscation, encryption of sensitive data, app integrity verification, and debugger detection.

Keywords: Flutter, mobile application security, reverse engineering, obfuscation, encryption, anti-debugging.

Вступ

Зворотний інжиніринг мобільних додатків є серйозною загрозою для безпеки програмного забезпечення, що може призвести до крадіжки інтелектуальної власності, компрометації даних та обходу ліцензійних обмежень. Суть зворотного інжинірингу полягає у відтворенні початкового коду додатку з його зібраного виконуваного файлу. Це дозволяє зловмисникам отримати доступ до вихідного коду, логіки бізнес-процесів, API-ключів та інших конфіденційних даних.

Flutter є потужним кросплатформовим фреймворком, який дозволяє розробляти додатки для Android, iOS, Web, Windows та інших платформ. Його основна перевага – можливість використання єдиного коду для різних платформ, що значно спрощує розробку та підтримку. Однак, через особливості компіляції, Flutter-додатки вразливі до реверс-інжинірингу.

Метою цієї роботи є аналіз загроз, пов'язаних із зворотним інжинірингом Flutter-додатків, та розгляд ефективних методів захисту від цих атак.

Результати дослідження

У процесі дослідження було визначено основні методи захисту Flutter-додатків від зворотного інжинірингу, а також проаналізовано їх ефективність у контексті сучасних кіберзагроз. Мобільні додатки є ключовою частиною цифрової інфраструктури, тому забезпечення їхньої безпеки є важливим аспектом розробки програмного забезпечення [1].

Основні загрози безпеки мобільних додатків

Зворотний інжиніринг є поширеною практикою серед зловмисників, що дозволяє їм аналізувати структуру додатка, модифікувати його поведінку та отримувати доступ до конфіденційних даних. Основні загрози, які виникають унаслідок зворотного інжинірингу, включають:

1. Декомпіляція коду та аналіз вихідного коду

Використовуючи спеціалізовані інструменти, такі як JADX, Hopper, Frida або Ghidra, зловмисники можуть витягти байт-код та відновити логіку додатку. Це дозволяє їм дослідити архітектуру програми, отримати доступ до бізнес-логіки та знайти потенційні вразливості [2].

2. Витік конфіденційної інформації

Недостатній рівень захисту даних може призвести до витоку API-ключів, паролів та персональних даних користувачів. Небезпечне зберігання таких даних у відкритому вигляді робить їх доступними для аналізу навіть без декомпіляції додатку.

3. Ін'єкція шкідливого коду та модифікація додатку

Зловмисники можуть модифікувати APK-файли, вбудовуючи у них власний шкідливий код або змінюючи автентифікацію, що дозволяє, наприклад, обходити платні функції чи здійснювати несанкціоновані дії від імені користувача.

4. Обхід механізмів автентифікації

Якщо додаток використовує слабкі механізми автентифікації або зберігає сесійні токени без захисту, це може дозволити зловмисникам отримати доступ до приватних даних без введення правильних облікових даних.

Методи захисту Flutter-додатків від зворотного інжинірингу

Захист мобільних додатків від зворотного інжинірингу є комплексним завданням, що включає низку заходів безпеки, кожен з яких відіграє важливу роль у запобіганні аналізу, модифікації та компрометації додатків.

1. Офускація коду

Офускація (obfuscation) – це процес перетворення вихідного коду у вигляд, який ускладнює його читання та аналіз. У Flutter підтримується офускація. Офускація дозволяє змінювати назви класів, методів та змінних, що значно ускладнює реверс-інжиніринг. Крім того, для захисту нативного коду можна використовувати ProGuard або R8, що мінімізують та оптимізують байт-код.

Основні переваги офускації:

1. Зменшення ймовірності аналізу бізнес-логіки додатку.
2. Ускладнення декомпіляції та розуміння структури коду.
3. Мінімізація ризику витоку конфіденційних даних через статичний аналіз.

Проте слід зазначити, що офускація не є ідеальним методом захисту, оскільки досвідчені аналітики можуть використовувати динамічний аналіз, налагоджувачі та інші техніки для обходу цього механізму.

2. Шифрування конфіденційних даних

Шифрування є важливим методом захисту конфіденційної інформації у Flutter-додатках. Воно забезпечує, що навіть у разі отримання доступу до сховища або трафіку додатка зловмисник не зможе прочитати зашифровані дані.

Основні підходи до шифрування даних:

Шифрування API-ключів та конфіденційних облікових даних замість їх зберігання у відкритому вигляді в коді. Найкращою практикою є використання змінних середовища або збереження ключів на сервері.

Використання FlutterSecureStorage [3] для безпечного зберігання паролів, токенів автентифікації та інших чутливих даних.

Реалізація наскрізного шифрування (E2EE) для захисту даних користувачів під час передачі.

Застосування AES (Advanced Encryption Standard) з надійним управлінням ключами для локального збереження даних.

Попри високий рівень захисту, шифрування має використовуватися у поєднанні з іншими заходами безпеки для запобігання витоку ключів або модифікації даних.

3. Перевірка цілісності додатка

Перевірка цілісності додатка дозволяє виявити несанкціоновані модифікації. Якщо зловмисник спробує змінити код додатка або втрутитися у його роботу, такі перевірки можуть ідентифікувати ці зміни та заблокувати виконання програми.

Методи забезпечення цілісності додатка:

Використання криптографічних підписів для перевірки автентичності APK або IPA-файлу.

Реалізація перевірки контрольних сум під час виконання, що дозволяє виявити зміни у файлах додатка.

Застосування Google Play Integrity API (для Android) [4] та App Attest від Apple (для iOS) для перевірки оригінальності додатка перед виконанням.

Серверна перевірка цілісності додатка, де бекенд перевіряє, чи відповідає запущена версія очікуваній.

Ці методи ускладнюють зміну додатка та унеможливають його використання у модифікованому вигляді.

4. Виявлення дебагерів та емуляторів

Зловмисники часто використовують інструменти налагодження та емулятори для аналізу та зміни Flutter-додатків. Виявлення таких інструментів дозволяє запобігти несанкціонованому доступу та модифікаціям.

Методи виявлення дебагерів та емуляторів:

Перевірка наявності інструментів налагодження (Frida, Xposed, GDB) шляхом аналізу процесів і використання API для виявлення дебагера.

Виявлення емуляторів через перевірку характеристик пристрою (серійний номер, параметри апаратного забезпечення, мережеві налаштування).

Моніторинг системних викликів, які можуть свідчити про втручання у роботу програми.

Застосування анти-дебаг технік, таких як динамічне виявлення брейкпойнтів та припинення роботи додатка у разі виявлення дебагера.

Інтеграція цих заходів значно ускладнює аналіз та модифікацію коду додатка.

Висновки

Flutter-додатки, як і будь-яке програмне забезпечення, можуть стати об'єктом атак, пов'язаних із зворотним інжинірингом та компрометацією безпеки. Однак застосування офускації коду, шифрування даних, перевірки цілісності та виявлення дебагерів дозволяє значно зменшити ризики несанкціонованого доступу та маніпуляцій. Безпека мобільних додатків є безперервним процесом, що потребує постійного вдосконалення та адаптації до нових кіберзагроз.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Flutter Documentation – Security Best Practices [Електронний ресурс] – Режим доступу: <https://docs.flutter.dev/security> (дата звернення: 18.03.2025).

2. Sajjad S. Protecting Your Flutter App: A Comprehensive Guide to Avoid Reverse Engineering [Електронний ресурс] – Режим доступу: <https://medium.com/@samra.sajjad0001/protecting-your-flutter-app-a-comprehensive-guide-to-avoid-reverse-engineering-66fbdc590958> (дата звернення: 18.03.2025).

3. Flutter Secure Storage [Електронний ресурс] – Режим доступу: https://pub.dev/packages/flutter_secure_storage (дата звернення: 18.03.2025).

4. Google Play Integrity API [Електронний ресурс] – Режим доступу: <https://developer.android.com/google/play/integrity> (дата звернення: 19.03.2025).

Осипенко Ірина Віталіївна – студентка групи ІІСТ-24м, кафедра автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: student.osypenko@gmail.com.

Богач Ілона Віталіївна – к.т.н., доцент кафедри автоматизації та інтелектуальних інформаційних технологій, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м.Вінниця, e-mail: ilona.bogach@gmail.com.

Osypenko Iryna Vitaliivna – student of IIIST-24m group, Department of Automation and Intelligent Information Technologies, Faculty of Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: schlkr7@gmail.com.

Bogach Ilona Vitaliivna – Associate Professor of Automation and Intelligent Information Technologies Department, Faculty of Intelligent Information Technology and Automation, Vinnytsia National Technical University, Vinnytsia, e-mail: ilona.bogach@gmail.com.