

УДК 004.056.5

Дьячук Т.С.¹, Антонюк Д.С.²

¹ асист. НУ "Запорізька політехніка"

² студ. гр. КНТ-516 НУ "Запорізька політехніка"

ЗАХИСТ ANDROID-ЗАСТОСУНКУ ВІД ЗЛОМУ

При розробці мобільного застосунку слід враховувати, що дані, якими оперує цей застосунок, можуть представляти певний інтерес для третіх осіб. Міра цінності цих даних варіюється в широких межах, проте, навіть найбільш проста приватна інформація, наприклад, пароль входу у застосунок, вимагає опрацювання її захисту. Особливо це важливо у світі поширення мобільних застосунків на усі сфери електронних послуг, включаючи фінансові, банківські операції, зберігання і передачу особистих даних тощо. Задача, яка стоїть перед розробником застосунку – створити надійне програмне забезпечення, яке буде виконувати свої безпосередні функції, швидше та краще за інших і при цьому мати обов'язково захист від злому, крадіжки, копіювання [1].

Гаджет - невеликий пристрій, призначений для полегшення і удосконалення життя людини. Сьогоднішній ринок розумних гаджетів пропонує близько 80% смартфонів під керівництвом операційної системи (ОС) Android, що приводить до високого попиту на застосунки цієї ОС. Рівень піратства в екосистемі Android досить високий. Застосунок не тільки легко вкрати - його легко зламати, відв'язати від сервісів перевірки, відключити рекламу (економічна вигода розробника) або навіть впровадити в нього «бекдор» (метод обходу стандартних процедур аутентифікації).

Існує достатньо уразливостей застосунків, наприклад: використання незахищених локальних сховищ; зберігання критично важливих даних в коді; застосування алгоритмів зі зберіганням приватного ключа; використання асиметричного алгоритму з приватним ключем, відомим серверу; використання самописних алгоритмів шифрування і захисту; передача критично важливих даних в зовнішнє середовище у відкритому виді; реверсивна інженерія алгоритмів, що представляють інтелектуальну цінність.

Розглянемо основні методи захисту застосунків: системи ліцензування, вбудовані в магазини застосунків, платформи залежні бібліотеки для перевірки ліцензії; обчислення на стороні сервера (Server-side computation);

запобігання декомпіляції (написання частин коду на «нативній» мові, обфускація коду [2], мінімізація коду); перенесення частини програми на сервер; шифрування передачі даних клієнт-сервер; зберігання даних у обробленому та/або шифрованому вигляді; мінімізація запитів до серверу; використання надійних хеш- та шифро- стандартів; заборона збереження даних на зовнішньому накопичувачі.

Не рекомендується вигадувати нові принципи шифрування або створення хеш функції, адже велика ймовірність прогледіти вразливість власного алгоритму і швидкість його підбору та злому. Тим не менш, користування декількома методами шифрування одночасно над одними і тими самими даними – не доцільне. Додаткові методи рішення та поради:

- найбільш вдалим варіантом призначеного для користувача коду є графічний код, цифровий (6 цифр або більше), - як додатковий варіант;

- HTTP-клієнт має бути налаштований на примусове використання захищеного каналу зв'язку (HTTPS);

- у релізній збірці застосунка мають бути відключені усі налагоджувальні функції, наприклад, опція `debuggable`, щоб уникнути можливості підключення до програми зовнішнім налагоджувальним застосуванням;

- на екранах застосунка, де розміщується приватна інформація користувача, буде не зайвою примусова заборона на програмне створення скриншотів вікна застосунка, а також відключення показу скриншотів в диспетчерові завдань;

- будь-яка приватна інформація може додатково перевірятися запитом особистого ключа користувача, заданого їм для входу в застосунок (якщо такий є);

- усередині коду застосунка рекомендується користуватися відносними шляхами (`getCanonicalPath`) замість абсолютних шляхів (`getAbsolutePath`);

- використовувати в застосунку відкриті компоненти (наприклад, `Service` або `Content Provider`) мають бути обов'язково закриті за допомогою прапора `exported = false` в маніфесті застосунка (`Android Manifest`). Це дозволить заборонити доступ до цих компонент з іншого застосунка.

Таким чином, в умовах, коли користувач або зломисник може отримати повний доступ до застосунка, існує лише одне оптимальне рішення: слід зберігати як можна менше секретних даних на стороні клієнта, завантажуючи важливу інформацію з боку сервера, при необхідності. Використовувати захищений протокол. Забезпечити повний захист від злому неможливо, і з цим доводиться миритися. Однак система вважається надійною, якщо вартість даних виявляється нижче, ніж вартість їх отримання. Комбінація описаних вище способів захисту допоможе створити надійний та безпечний Android - застосунок.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Георгий Малюков. Безопасность данных в разработке мобильных приложений [Электронный ресурс] / George Malyukov // Habr. – Режим доступа: <https://habr.com/ru/post/327760> – Назва з екрана.

2. Денис Матвеев. Защита Андроид приложения от отладки [Электронный ресурс]/ Denis Matveev // Cryptoworld. – Режим доступа: <https://cryptoworld.su/zashhita-android-prilozheniya-ot-otladki> – Назва з екрана.