УДК 621.397.7-182.3

АНАЛИЗ УЯЗВИМОСТЕЙ ANDROID-ПРИЛОЖЕНИЙ НА ПРИМЕРЕ ПОПУЛЯРНЫХ МОБИЛЬНЫХ БАНКОВ

Агапова В.О.

Белорусский государственный университет информатики и радиоэлектроники, г. Минск, Республика Беларусь

Научный руководитель: Лихачевский Д.В. – к. т. н., доцент, декан ФКП

Аннотация. В данной статье проведён анализ распространённых уязвимостей *Android*-приложений с фокусом на мобильные банковские приложения. Рассматриваются основные методы проведения статического и динамического анализа, а также выявленные типовые уязвимости, влияющие на безопасность пользовательских данных и транзакций. Обоснована необходимость соблюдения принципов безопасной разработки и рассмотрены рекомендации по устранению наиболее критичных проблем.

Ключевые слова: *Android*, уязвимости, безопасность приложений, мобильный банкинг, статический анализ, динамический анализ.

Введение. Мобильные банковские приложения стали неотъемлемой частью цифровой экономики. С их помощью пользователи осуществляют финансовые операции, управляют счетами и хранят конфиденциальные данные. Однако, с ростом популярности мобильного банкинга растёт и интерес со стороны злоумышленников, что делает безопасность таких приложений критически важной. *Android* как самая распространённая мобильная платформа подвержена большому количеству угроз из-за открытости экосистемы и многообразия устройств.

Цель настоящей статьи – анализ уязвимостей, наиболее характерных для мобильных банковских *Android*-приложений, и выработка рекомендаций по их предотвращению.

Основная часть. Анализ *Android*-приложений проводится двумя основными способами: статическим и динамическим. Статический анализ позволяет изучать структуру *APK*-файлов без запуска, выявляя уязвимости в коде, конфигурациях, разрешениях. Для этого применяются инструменты, такие как *JADX* (декомпиляция), *MobSF* (всесторонний анализ), *APKTool* (распаковка ресурсов). Динамический анализ требует запуска приложения в контролируемой среде. Используются эмуляторы с установленными фреймворками *Frida* или *Xposed*, позволяющими перехватывать вызовы методов. *Burp Suite* применяется для анализа сетевого взаимодействия.

Важно оценивать:

– доступ к конфиденциальной информации;

– хранение токенов и ключей;

– уязвимости в *WebView*;

– использование небезопасных библиотек и *API*;

– целостность и проверку сертификатов.

Дополнительно стоит учитывать социальную инженерию и поведенческие атаки, направленные на обман пользователя, что усиливает важность комплексного тестирования.

Практика показывает, что даже крупные банки допускают ошибки. Среди уязвимостей чаще всего встречаются:

– хранение данных без шифрования: ключи доступа, токены, логины, пароли в *SharedPreferences* или на *SD*-карте;

– SSL-подмена: отсутствие pinning приводит к возможности *MITM*-атак;

– использование *WebView* с доступом к *JavaScript*-интерфейсу без фильтрации;

– уязвимые сторонние библиотеки, особенно старые версии *SDK*;

– разрешения, не соответствующие функционалу (например, доступ к геолокации, камере, микрофону);

– открытые лог-файлы, содержащие персональные данные;

– неправильная реализация авторизации и отсутствие защиты от брутфорса.

Зачастую разработчики забывают реализовать защиту от подмены *Activity*, проверки подписи приложения, а также проверки *root*-доступа на устройстве.

В рамках работы были протестированы 5 банковских приложений, установленных из *Google Play*. Для обеспечения этических норм их названия не раскрываются. Использовалась виртуальная среда на базе *Genymotion* с *Android* 9.0 и *root*-доступом. Выявлены следующие проблемы:

– некоторые приложения отправляли чувствительные данные в логах;

– в одном из случаев токен авторизации сохранялся в незашифрованном виде;

– отсутствие *SSL pinning* позволяло успешно выполнить *MITM*-атаку с подставным сертификатом;

– приложения не проверяли наличие *root*-доступа;

– использование устаревших *SDK* от сторонних разработчиков без обновлений.

Более того, ни одно из приложений не реализовало двухфакторную аутентификацию при авторизации, что делает атаки через поддельные приложения реальными.

Для анализа были использованы следующие инструменты:

– *JADX* – декомпиляция исходного кода;

– *MobSF* – анализ конфигураций, разрешений, кода, библиотек;

– *Frida* – инструмент динамического перехвата вызовов;

– *Burp* Suite – прокси-сервер для перехвата *HTTPS*-трафика;

– *Wireshark* – анализ сетевых пакетов;

– *Genymotion* – эмулятор с *root*-доступом.

Такая комбинация позволяет получить всестороннее представление об архитектуре и поведении приложения. Также важно проводить тестирование в условиях максимально приближенных к реальным: использовать настоящие банковские функции, симулировать атаки, анализировать логи и трафик.

На основе анализа можно предложить следующие меры:

– реализация *SSL pinning* и защита от *MITM*;

– шифрование всех хранимых данных (*AES*-256);

– использование *KeyStore* для хранения ключей;

– проверка подписи приложения при запуске;

– проверка *root*-доступа и *emulator detection*;

– минимизация прав доступа;

– внедрение двухфакторной аутентификации;

– регулярный аудит кода и зависимостей;

– использование библиотек, поддерживающих защиту от реверс-инжиниринга.

Кроме того, важно просвещать пользователей относительно основных рисков, не допускать публикации приложений без сквозного тестирования и регулярно обновлять компоненты *SDK*.

Заключение. Безопасность мобильных *Android*-приложений требует комплексного подхода, особенно в сфере банковских сервисов, где ошибки могут привести к серьёзным финансовым потерям. Даже крупные организации могут допускать базовые ошибки в реализации, что подтверждается результатами анализа. Применение методик безопасной разработки, регулярный аудит, соблюдение рекомендаций *OWASP* и внедрение многоуровневой защиты – всё это необходимо для надёжного функционирования банковских *Android*-приложений. Создание безопасного мобильного банкинга должно стать приоритетом каждого разработчика и организации, работающей с персональными данными клиентов.

Список литературы

1. *Android Security Best Practices – Android Developers … [Electronic resource]. – Date of access: March 19, 2025. – https://developer.android.com/topic/security/best-practices.*
2. *OWASP Mobile Security Testing Guide. – OWASP Foundation… [Electronic resource]. – Date of access: March 19, 2025. – https://owasp.org/www-project-mobile-security-testing-guide/.*
3. *Mobile Security Framework (MobSF) … [Electronic resource]. – Date of access: March 19, 2025. –: https://mobsf.github.io*
4. *NIST SP 800-163 Rev. 1 – Vetting the Security of Mobile Applications / Michael Ogata, Vincent Sritapan. – 2022. – Pp. 48.*
5. *A Study of Android Application Security / Enck, W., Octeau, D., McDaniel, P., Chaudhuri, S. // USENIX. – 2011. – Pp. 16.*
6. *Frida – Dynamic Instrumentation Toolkit … [Electronic resource]. – Date of access: March 19, 2025. – https://frida.re.*
7. *Baloch, R. Ethical Hacking and Penetration Testing Guide. – CRC Press … [Electronic resource]. – Date of access: March 19, 2025. – https://www.booksfree.org/wp-content/uploads/2022/04/Ethical-Hacking-and-Penetration-Testing-Guide-by-Rafay-Baloch-booksfree.org_.pdf.*
8. *OWASP MASVS – Mobile Application Security Verification Standard… [Electronic resource]. – Date of access: March 19, 2025. –: https://owasp.org/www-project-mobile-app-security/.*
9. *Burp Suite Documentation – PortSwigger … [Electronic resource]. – Date of access: March 19, 2025. – https://portswigger.net/burp/documentation.*
10. *Android Keystore System – Android Developers … [Electronic resource]. – Date of access: March 19, 2025. – https://developer.android.com/training/articles/keystore.*

UDC 621.397.7-182.3

ANALYSIS OF ANDROID APPLICATION VULNERABILITIES USING POPULAR MOBILE BANKS

Agapova V.O.

Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus

Likhachevsky D.V.. – Cand. of Sci., associate professor, dean of FCAD

Annotation. This article analyzes common vulnerabilities in Android applications with a focus on mobile banking applications. It covers the main methods of static and dynamic analysis, as well as the identified typical vulnerabilities that affect the security of user data and transactions. It substantiates the need to adhere to the principles of secure development and provides recommendations for eliminating the most critical problems.

Keywords: Android, vulnerabilities, application security, mobile banking, static analysis, dynamic analysis.