Нестеров В.Ю., магістрант
Єфіменко А.А., к.т.н., доцент
Державний університет «Житомирська політехніка»

АЛГОРИТМ ТЕСТУВАННЯ БЕЗПЕКИ ANDROID-ДОДАТКІВ

У сучасних умовах швидкого розвитку мобільних технологій питання захисту Android-додатків набуває особливої актуальності. Зростання кількості кібератак, витоків даних та вразливостей у програмному забезпеченні створює ризики для конфіденційності, цілісності та доступності інформації користувачів. Відсутність належного тестування безпеки може призвести до серйозних наслідків, зокрема до викрадення персональних даних, фінансових втрат та репутаційних ризиків.

Метою цього дослідження є розробка алгоритму тестування безпеки Android-додатків, що дозволить ефективно виявляти потенційні загрози та надавати рекомендації щодо їх усунення. Алгоритм базується на сучасних підходах до мобільного пентестингу та стандартах безпеки, зокрема MASTG.

OWASP Mobile Application Security Testing Guide (MASTG) – це всеосяжний посібник з тестування безпеки мобільних додатків та reverse engineering. Він описує технічні процеси для перевірки засобів контролю, перелічених в OWASP MASVS, через слабкі місця, визначені OWASP MASWE [1].

Комплексне та результативне тестування безпеки Android-додатків передбачає виконання кількох етапів:

1. Збір та аналіз інформації про додаток: На першому етапі досліджується загальна структура Android-додатка, включаючи дозволи, компоненти та використані API. Аналізуються вихідні файли APK, визначаються можливі вразливості, пов’язані з некоректною конфігурацією безпеки та неправильним керуванням ресурсами;
2. Статичний аналіз: На другому етапі проводиться декомпіляція APK-файлу та аналіз його вихідного коду. Вивчається, чи використовуються незахищені механізми збереження даних, чи присутні слабкі алгоритми шифрування та некоректно налаштовані дозволи. Для цього застосовуються такі інструменти, як MobSF, JADX та інші засоби reverse engineering;
3. Динамічний аналіз безпеки: На третьому етапі тестується поведінка додатка під час його виконання в реальному середовищі або емуляторі. Виконується моніторинг мережевого трафіку за допомогою інструментів Burp Suite або MITMproxy, досліджується взаємодія додатка із сервером та проводиться перевірка на можливі загрози, такі як MITM-атаки чи витоки даних;
4. Аналіз безпеки зберігання даних: На четвертому етапі перевіряється, як додаток обробляє та зберігає конфіденційну інформацію. Аналізується наявність незашифрованих файлів, неправильно налаштованих баз даних SQLite, кешу та SharedPreferences. Виявляються потенційні уразливості, що можуть спричинити витік або компрометацію даних користувачів [2];
5. Перевірка автентифікації та авторизації: На п’ятому етапі оцінюється безпека механізмів входу та управління сесіями. Досліджуються методи автентифікації, зокрема використання токенів, паролів та двофакторної аутентифікації. Перевіряються можливі атаки, такі як Brute Force, Session Hijacking або Replay-атаки;
6. Експлуатація вразливостей та розробка рекомендацій: На шостому етапі здійснюється перевірка можливості експлуатації знайдених вразливостей. Проводиться тестування методів атаки та аналіз наслідків потенційного компрометування додатка. Формується звіт з детальним описом знайдених проблем та рекомендаціями щодо їх усунення.

Запропонований алгоритм тестування безпеки Android-додатків поєднує теоретичні та практичні аспекти. Теоретично він ґрунтується на системному підході до аналізу безпеки Android-додатків, що враховує всі ключові аспекти їх захисту, включаючи безпеку коду, даних, комунікацій, механізмів автентифікації тощо. Практично алгоритм реалізує покроковий процес тестування, що дає змогу виявляти й усувати вразливості додатків, використовуючи актуальні методи, інструменти та техніки, що дозволяють підвищити їхній рівень захисту та стійкості до атак.

Перспективи подальших досліджень включають вдосконалення методів виявлення вразливостей, розширення спектру тестованих аспектів безпеки та підвищення ефективності автоматизації тестування.

Список використаних джерел:

1. OWASP MASTG - OWASP Mobile Application Security. *OWASP Mobile Application Security*. URL: https://mas.owasp.org/MASTG/.
2. Gunasekera S. Android Apps Security. Berkeley, CA : Apress, 2020.