Bezpieczeństwo aplikacji mobilnych

Laboratorium 6

Testowanie bezpieczeństwa aplikacji mobilnych

Cel zajęć:

- Zdobycie praktycznych umiejętności w zakresie testowania bezpieczeństwa aplikacji mobilnych.
- Przeprowadzenie testów bezpieczeństwa na stworzonych aplikacjach mobilnych
- Poznanie narzędzi i technik służących do identyfikacji i analizy podatności w aplikacjach mobilnych.

Tematyka:

- Podstawy testowania bezpieczeństwa aplikacji mobilnych.
- Wykorzystanie narzędzi do analizy bezpieczeństwa, takich jak OWASP ZAP i MobSF.
- Identyfikacja i analiza podatności aplikacji na różne typy ataków.
- Dokumentowanie wyników testów i proponowanie rozwiązań.

Wymagane narzędzia:

- Node.js oraz npm lub yarn.
- Expo CLI.
- Emulator Android/iOS lub fizyczne urządzenie z aplikacją Expo Go.
- Narzędzia do testowania bezpieczeństwa:
    - **OWASP ZAP** (Zed Attack Proxy)
    - **MobSF** (Mobile Security Framework)
- Aplikacja mobilna do testowania

Zadania do wykonania:

1. **Przygotowanie środowiska:**
    - Zainstaluj narzędzia do testowania bezpieczeństwa, takie jak OWASP ZAP i MobSF.
    - Upewnij się, że masz dostępną aplikację Expo React Native do testowania. Możesz użyć aplikacji stworzonej w poprzednich laboratoriach.
2. **Analiza statyczna aplikacji za pomocą MobSF:**
    - Uruchom MobSF na swoim komputerze.
    - Wyeksportuj plik APK (dla Androida) lub odpowiedni plik dla iOS z aplikacji Expo.
    - Zaimportuj plik aplikacji do MobSF i przeprowadź statyczną analizę.
    - Przeanalizuj raport wygenerowany przez MobSF, zwracając uwagę na:
        - Uprawnienia wymagane przez aplikację.
        - Obecność wrażliwych informacji w kodzie (np. klucze API).
        - Potencjalne podatności i zalecenia dotyczące bezpieczeństwa.
3. **Konfiguracja OWASP ZAP do analizy dynamicznej:**
    - Skonfiguruj OWASP ZAP jako proxy, przez które będzie przechodzić ruch sieciowy z aplikacji mobilnej.

- Na urządzeniu lub emulatorze ustaw odpowiednie proxy, aby ruch sieciowy był przechwytywany przez OWASP ZAP.

4. **Analiza dynamiczna aplikacji za pomocą OWASP ZAP:**
   - Uruchom aplikację mobilną i wykonaj typowe operacje, np. logowanie, pobieranie danych, wysyłanie informacji.
   - W OWASP ZAP obserwuj przechwycony ruch sieciowy.
   - Analizuj żądania i odpowiedzi pod kątem:
     - Czy dane są przesyłane w sposób zaszyfrowany (HTTPS)?
     - Czy wrażliwe informacje nie są przesyłane w postaci niezaszyfrowanej?
     - Czy w nagłówkach lub treści żądań nie ma ujawnionych wrażliwych danych?
5. **Identyfikacja podatności na ataki:**
   - Sprawdź, czy aplikacja jest podatna na ataki typu:
     - **SQL Injection:** Wprowadzaj w polach wejściowych znaki specjalne i obserwuj reakcję aplikacji.
     - **Cross-Site Scripting (XSS):** Próbuj wprowadzić skrypty w polach tekstowych.
     - **Brak walidacji certyfikatów SSL/TLS:** Sprawdź, czy aplikacja akceptuje nieprawidłowe certyfikaty.
   - Dokumentuj wszelkie nieprawidłowości i potencjalne luki.
6. **Testowanie zabezpieczeń aplikacji:**
   - Wykorzystaj narzędzia i techniki do symulacji ataków na aplikację.
   - Sprawdź, jak aplikacja reaguje na nieoczekiwane dane wejściowe lub manipulacje w ruchu sieciowym.
   - Upewnij się, że mechanizmy zabezpieczające, takie jak walidacja danych i obsługa błędów, działają poprawnie.
7. **Raportowanie wyników:**
   - Sporządź raport zawierający:
     - Opis przeprowadzonych testów i użytych narzędzi.
     - Wykryte podatności i ich potencjalny wpływ na bezpieczeństwo aplikacji.
     - Zrzuty ekranu lub logi potwierdzające istnienie podatności.
     - Zalecenia dotyczące naprawy wykrytych problemów.
8. **Propozycje poprawek i wdrożenie:**
   - Na podstawie wyników testów zaproponuj konkretne poprawki w kodzie aplikacji.
   - Wprowadź niezbędne zmiany w aplikacji, aby usunąć wykryte podatności.
   - Przykładowe działania:
     - Dodanie walidacji danych wejściowych.
     - Poprawa konfiguracji sieciowej, aby wymusić użycie HTTPS.
     - Usunięcie wrażliwych informacji z kodu źródłowego.
9. **Weryfikacja skuteczności poprawek:**
   - Ponownie przeprowadź testy, aby sprawdzić, czy wprowadzone zmiany skutecznie usunęły podatności.
   - Upewnij się, że aplikacja nadal działa poprawnie po wprowadzeniu poprawek.