Testy Penetracyjne Lab 3

Julia Sadecka, Cyberbezpieczeństwo

Zad 1

OWASP Web Security Testing Guide jest to kompleksowy przewodnik po testowaniu bezpieczeństwa aplikacji i usług internetowych. Zawiera on zbiór zaleceń i technik pozwalających w uschematyzowany sposób przeprowadzić testy.

WSTG składa się z:

- *Wprowadzenia*
- *The OWASP Testing Framework* (przeglądu frameworka)
- *Web Application Security Testing* (sekcja dotycząca przeprowadzania różnych rodzajów testów tj. testy autoryzacji czy testów walidacji danych

OWASP TOP 10 to lista dziesięciu najbardziej krytycznych zagrożeń bezpieczeństwa aplikacji webowych. Lista jest aktualizowana co jakiś czas (2017/2021 rok) To narzędzie pomaga organizacją zidentyfikować i zrozumieć kluczowe ryzyka związane z bezpieczeństwem aplikacji

Dokument OWASP TOP 10 składa się z:

- *Wprowadzenia*
- *OWASP Top Ten Risks* (lista dziesięciu najważniejszych zagrożeń)
- *Mapping to Standards*
- *Data Verification Requirements*

Oba te narzędzia mogą być wykorzystywane jako odnośniki do przeprowadzania testów bezpieczeństwa. TOP 10 jako punkt odniesienia do identyfikacji kluczowych zagrożeń, a WSTG jako narzędzie do przeprowadzania tych testów.
Dodatkowo WSTG może uwzględniać zalecenia z TOP 10 podczas opisywania konkretnych testów lub praktyk.

Zad 2 - Plan Testu Penetracyjnego

1. **Rozpoznanie serwera i infrastruktury:**
    a) Zidentyfikuj wersję systemu operacyjnego i serwera HTTP/HTTPS.
    b) Analizuj konfigurację serwera, w tym obsługę błędów, nagłówki (np. HSTS).
    c) Sprawdź popularne ścieżki (robots.txt, security.txt) oraz dostęp do konfiguracji TLS.
2. **Analiza aplikacji i endpointów:**
    a) Zbierz informacje o wykorzystywanych endpointach.
    b) Zidentyfikuj frameworki i technologie używane w aplikacji.
    c) Analizuj ciasteczka, ich flagi oraz inne nagłówki bezpieczeństwa (np. CSP).

d) Sprawdź, jakie metody zapytań są akceptowane i czy występują zapytania do zewnętrznych usług.

3. **Testowanie bezpieczeństwa środowiska:**
   a) Poszukaj znanych podatności w aplikacji, frameworkach i serwerze.
   b) Sprawdź konfigurację serwera, np. czy jest włączone wyświetlanie listy plików w folderze.
   c) Sprawdź, czy istnieją stare endpointy, które mogą być potencjalnie podatne.
   d) Oceń dostępność zasobów, do których nie powinno być dostępu.
4. **Bezpieczeństwo identyfikacji użytkowników:**
   a) Analizuj sposób generowania nazw użytkowników.
   b) Sprawdź, czy istnieje możliwość enumeracji użytkowników.
   c) Oceń nadawanie uprawnień, zwłaszcza administratorów.
   d) Zbadaj zdefiniowane role i ich dostęp do wrażliwych zasobów.
5. **Bezpieczeństwo uwierzytelniania użytkowników:**
   a) Sprawdź, czy system akceptuje domyślne/typowe nazwy użytkowników i hasła.
   b) Przeanalizuj zabezpieczenia przed atakami bruteforce.
   c) Oceń procedurę resetowania hasła i wymagania dotyczące hasła.
   d) Sprawdź, czy hasła są przesyłane w formie zaszyfrowanej.
6. **Bezpieczeństwo autoryzacji użytkowników:**
   a) Poszukaj możliwości eskalacji uprawnień.
   b) Sprawdź, czy istnieją podatności typu Path Traversal.
   c) Przeanalizuj, czy istnieją słabości protokołu OAuth.
7. **Zarządzanie sesjami:**
   a) Przeanalizuj generowanie sesji, ich flagi i czas ważności.
   b) Testuj funkcję wylogowywania użytkowników.
   c) Sprawdź podatność na ataki CSRF.
8. **Obsługa danych od użytkownika:**
   a) Przeprowadź ataki XSS, SQL Injection itp.
   b) Zbadaj funkcje pozwalające na wgrywanie plików przez użytkowników.
9. **Testy bezpieczeństwa API:**
   a) Sprawdź autoryzację i uwierzytelnianie w API.
   b) Przeanalizuj dostępność i zabezpieczenia endpointów API.
   c) Przetestuj, czy istnieje możliwość ataków na API, takich jak SQL Injection.
10. **Testowanie Słabej Kryptografii:**
    a) Oceń używane algorytmy kryptograficzne w aplikacji.
    b) Sprawdź, czy klucze kryptograficzne są bezpiecznie przechowywane.
    c) Przeanalizuj protokoły komunikacyjne pod kątem zastosowania silnych szyfrów.
    d) Przetestuj odporność na ataki typu man-in-the-middle, zwłaszcza w kontekście kryptografii.
11. **Testowanie Logiki Biznesowej:**
    a) Zidentyfikuj i zweryfikuj, czy logika biznesowa aplikacji jest odpowiednio zabezpieczona przed manipulacją.

b) Przetestuj scenariusze działania aplikacji, które mogą prowadzić do niepożądanych rezultatów lub dostępu do niedozwolonych zasobów.
c) Sprawdź, czy mechanizmy kontroli dostępu są skonfigurowane zgodnie z zasadami logiki biznesowej.

12. Testowanie Zabezpieczeń Mobilnych:

a) Sprawdź, czy aplikacja mobilna jest zabezpieczona przed atakami, takimi jak reverse engineering.
b) Przeanalizuj zastosowane mechanizmy uwierzytelniania i autoryzacji w kontekście mobilnych platform.
c) Przetestuj, czy dane przechowywane na urządzeniach mobilnych są odpowiednio zaszyfrowane.
d) Przeprowadź analizę podatności związanych z korzystaniem z funkcji urządzeń mobilnych, takich jak aparat, GPS itp.