SZKOLENIE ŚREDNIO ZAAWANSOWANE

# Testy bezpieczeństwa nowoczesnych aplikacji internetowych

SEC/WEB

Czas trwania: 3 dni (24h)

## Cele szkolenia

- Przekazanie wiedzy, która uczyni z uczestnika nie tylko weryfikatora bezpieczeństwa na podstawie baz wiedzy i gotowych exploitów ale da podstawy do zostania researcherem bezpieczeństwa zdolnym do pracy z nieznanymi aplikacjami i protokołami oraz wyszukiwania nowych podatności w nich

## Zalety

- Mocną stroną szkolenia są przykłady dla aplikacji .NET (stack Microsoft) i Java (Spring Boot i Docker)
- Uczestnicy wykonują zadania związane z projektem testów penetracyjnych na laboratorium w formie zwirtualizowanego środowiska symulującego typowej problemy złożonej infrastruktury
- Praktyka przed teorią - wszystkie szkolenia technologiczne prowadzone są w formie warsztatowej. Konieczna teoria jest wyjaśniana na przykładzie praktycznych zadań
- Nauka z praktykami - wszyscy trenerzy na co dzień pracują w projektach, gwarantuje to dostęp do eksperckiej wiedzy i praktycznego know-how
- Wykorzystanie narzędzi OWASP ZAP (szkolenia otwarte) i Burp Suite (dedykowane, jeśli klient jest zainteresowany i posiada licencję)

## Dla kogo?

- Szkolenie jest kierowane do testerów, programistów, administratorów aplikacji, audytorów oraz fascynatów bezpieczeństwa pragnących zdobyć całościową wiedzę z zakresu prowadzenia testów penetracyjnych oraz wykorzystania jej do weryfikacji bezpieczeństwa złożonych systemów informatycznych

## Wymagania

- Od uczestników wymagane jest doświadczenie w pracy z aplikacjami internetowymi - najlepiej jako programista lub wdrożeniowiec albo doświadczenie z dziedziny bezpieczeństwa takich rozwiązań
- Znajomość podstaw Java oraz .NET a także podstaw administracji w systemach Windows i Linux pozwoli bezboleśnie przejść przez wszystkie laboratoria

- Oprogramowanie można uruchomić jako kontenery Docker lub w maszynie wirtualnej VirtualBox

## Program

1. Rodzaje testów penetracyjnych
   a. Metodyki, typy i fazy testów penetracyjnych
   b. Checklisty w testach penetracyjnych, standardy kodowania: CIS, CERT
2. Rodzaje podatności, typy podatności według różnych klasyfikacji
   a. Klasyfikacja podatności według OWASP i CWE
   b. baza podatności CVE
   c. metody szacowania ryzyka, modelowanie zagrożeń
3. Narzędzia do rekonesansu
   a. Amass, subfinder, Foca
   b. MassDNS
   c. Google dorks
   d. Shodan
4. Narzędzia do rozponania aplikacji
   a. nmap, skrypty nmap-a
   b. Fingerprinting za pomocą komunikatów błędów
      -narzędzie BeanStack
   c. Użycie narzędzi ffuf i dirb oraz baz FuzzDB i Directory List do odkrywania ukrytych zasobów
   d. Użycie kiterunner do odkrywania ukrytych API
   e. Użycie Param Miner lub Arjun do odkrywania ukrytych parametrów
   f. Użycie JSMiner do odkrywania zasobów i danych ukrytych w plikach Javascript
5. Ataki na aplikacje webowe
   a. Wykorzystanie funkcjonalności debug
   b. Problem z uwierzytelnianiem i autoryzacją np zarządzanie sesją, ataki na JWT
      -wykorzystanie Autorepeater (Burp Suite)
      -modyfikacje nagłówków
   c. Ataki na fukcjonalności logowania, rejestracji użytkowników i odzyskiwanie hasła
      -obsługa logowania w OWASP ZAP lub Burp Suite z użyciem makr
   d. SQL/HQL Injection z wykorzystaniem SQLMap
   e. Podatności typu (XPath, XML, Command, Script, LDAP)-Injection
   f. Niebezpieczna deserializacja w Javie i .NET
      -Wykonanie kodu za pomocą XML, JSON, YAML, XStream
      -Wykorzystanie podatności Log4Shell
   g. Wykonanie kodu przez funkcjonalność uploadu plików
   h. Mass assignment
      -Wykorzystanie podatności Spring4Shell
   i. Ataki na usługi REST JAX-RS
   j. Wstrzykiwanie kodu Javascript: XSS
   k. Wykonanie akcji w uwierzytelnionej sesji użytkownika: CSRF
   l. Bezpośredni dostęp do danych i obiektów (IDOR)
      -Modyfikacje obiektów, wycieki danych osobowych
   m. Path Traversal, nieuprawnione pobieranie lokalnych i zdalnych plików

n. Ataki związane z przetwarzaniem XML: XXE
o. Spring EL Injection, ataki na Spring Framework
p. Template Injection
q. OGNL Injection
r. HTTP Request Smuggling
s. Ataki na systemy backendowe przez błędy w aplikacjach frontowych

6. Narzędzia do testów manualnych typu proxy
   a. Wykorzystanie Burp Suite, OWASP ZAP
7. Automatyczne skanery bezpieczeństwa
   a. Wykorzystanie OWASP ZAP, Burp Suite Scan, OpenVAS
   b. Rozbudowa skanera o własne reguły z pomocą Burp Bounty (Burp Suite)
   c. Narzędzia zapewniające bezpieczeństwo w trakcie developmentu: OWASP Dependency Check, Retire.js, Find-Sec-Bugs, Semgrep
8. Zbiory exploitów
   a. Wykorzystanie Metasploit
9. Skrypty i automatyzacja testów bezpieczeństwa
   a. Wykorzystanie ZAP i Mozilla ZEST
   b. Integracja OWASP ZAP z Jenkins
   c. Wykrywanie dziurawych komponentów z OWASP Dependency Check w CI
10. Testowanie WebServices
    a. XXE
    b. SOAP
    c. XSLT
    d. BPEL
11. Kryptografia
    a. Weryfikacja poprawnej konfiguracji SSL
    b. Ataki Man-in-the-middle
    c. Słabości w implementacji kryptografii
12. Ataki DoS i DDoS
    a. Ataki na logikę aplikacji: ReDOS, XML Bomb, Flood
13. Cloud
    a. Specyficzne zagadnienia dla cloud: AWS
    b. Narzędzie ScoutSuite
14. Zarządzanie informacją w trakcie testu penetracyjnego
    a. Budowanie bazy wiedzy i bazy ataków
    b. Dradis Framework, Faraday IDE, Magic Tree
15. Tworzenie raportu
    a. Co powinien zawierać dobry raport z testów penetracyjnych?
    b. Jak formułować zalecenia i obejścia?
    c. Jak bezpiecznie dostarczyć raport do klienta?
    d. Jak opisać podatność i uzyskać CVE?