BO·TT·EGA
IT minds

Program szkolenia:

Bezpieczny kod - podstawy

Informacje:

Nazwa:	**Bezpieczny kod - podstawy**
Kod:	**Arch-Sec-intro**
Kategoria:	Bezpieczeństwo
Grupa docelowa:	developerzy
Czas trwania:	3 dni
Forma:	75% wykłady / 25% warsztaty

Niezwykle często programiści od pierwszych dni swojej kariery zawodowej tworzą złożone systemy, które wdrażane są w nieprzyjaznym środowisku, jakim jest dzisiejsza Sieć. Kurs został zaprojektowany tak, aby usystematyzować i przekazać uczestnikom praktyczną wiedzę dotyczącą pisania bezpieczengo oprogramowania, które miałoby szansę sprostać powszechnym zagrożeniom takim jak przepełnienie bufora czy cross-site scripting (XSS).

Niniejsze szkolenie jest znakomitym i nastawionym na praktykę wprowadzeniem do tematyki bezpieczeństwa oprogramowania. Celem kursu jest wyposażenie uczestników w absolutne minimum wiedzy i umiejętności, które powinien posiadać każdy zawodowy programista. W trakcie szkolenia uczestnicy mogą nie tylko poznać zagrożenia i techniki obrony, ale także wyrobić sobie intuicję co do tego, jakie rozwiązania można uznać za bezpieczne i jak pisać kod odporny na ataki.

Szkolenie skierowane jest zarówno do nowicjuszy jak i doświadczonych programistów, którzy dotychczas nie mieli styczności z pisaniem bezpiecznego kodu.

Zalety szkolenia:

- Budowa solidnych fundamentów i odpowiedniej postawy wobec zagadnień bezpieczeństwa
- Praktyczna wiedza niezbędna każdemu profesjonalnemu programiście
- Demonstracja najpopularniejszych ataków i technik obrony

BO·TT·EGA
IT minds

Szczegółowy program:

1. Zasady pisania bezpiecznego kodu

- 1.1. Podstawowe pojęcia
  - 1.1.1. Uwierzytelnienie
  - 1.1.2. Autoryzacja
  - 1.1.3. Poufność
  - 1.1.4. Integralność
  - 1.1.5. Odpowiedzialność
  - 1.1.6. Dostępność
  - 1.1.7. Niezaprzeczalość
  - 1.1.8. Podstawowe pojęcia w praktyce
- 1.2. Projektowanie bezpiecznego oprogramowania
  - 1.2.1. Zagrożenia
  - 1.2.2. Projektowanie z myślą o bezpieczeństwie
  - 1.2.3. Bezpieczeństwo a użyteczność systemu
  - 1.2.4. Security by obscurity
  - 1.2.5. Kod otwarty vs kod zamknięty
- 1.3. Zasady
  - 1.3.1. Zasada minimalnego uprzywilejowania
  - 1.3.2. Defense-in-depth
  - 1.3.3. Najsłabsze ogniwo
  - 1.3.4. Obsługa sytuacji błędów i sytuacji wyjątkowych
  - 1.3.5. Bezpieczeństwo jako proces

2. Techniki pisania bezpiecznego kodu

BO·TT·EGA IT minds

- 2.1. Przepełnienie bufora, czyli wróg publiczny numer 1
  - 2.1.1. Anatomia ataku typu "stack smashing"
  - 2.1.2. Pomocne biblioteki
  - 2.1.3. Analiza statyczna
  - 2.1.4. A co ze stertą?
  - 2.1.5. Ataki polegające na "przekręcaniu liczników"
- 2.2. Manipulacja stanem klienta HTTP, czyli dlaczego nie możemy ufać przeglądarkom?
  - 2.2.1. Atak na naiwną aplikację Webową
  - 2.2.2. Rozwiązania
    - 2.2.2.1. Przechowywanie stanu aplikacji na serwerze
    - 2.2.2.2. Zabezpieczenie integralności stanu po stronie klienta
  - 2.2.3. Ciasteczka (cookies)
    - 2.2.3.1. Gdzie kryje się niebezpieczeństwo?
    - 2.2.3.2. Zasady bezpiecznego obchodzenia się z ciasteczkami
  - 2.2.4. Kod JavaScript
    - 2.2.4.1. Czy można zaufać programom działającym w przeglądarce?
    - 2.2.4.2. Kod JavaScript jako wektor ataku
- 2.3. Wstrzykiwanie SQL, czyli najpopularniejsze zagrożenie dla aplikacji Webowych
  - 2.3.1. Scenariusz ataku
  - 2.3.2. Rozwiązania
    - 2.3.2.1. Blacklisting
    - 2.3.2.2. Whitelisting
    - 2.3.2.3. Escaping
    - 2.3.2.4. Jak może pomóc Twoja biblioteka dostępu do bazy danych?
    - 2.3.2.5. Procedury składowane

BOTTEGA IT minds

2.3.3. Zasada minimalnego uprzywilejowania (znowu!)

2.4. Jak bezpieczenie zarządzać kontami użytkowników?

2.4.1. Ataki słownikowe na hasła

2.4.2. Szkic bezpiecznego rozwiązania

2.4.2.1. Rejestracja

2.4.2.2. Logowanie

2.4.2.3. Odzyskiwanie i resetowanie haseł

2.4.2.4. Wylogowanie

2.5. Bezpieczeństwo aplikacji działających w przeglądarce

2.5.1. Cross-Site Request Forgery (CSRF)

2.5.1.1. Przykładowy atak

2.5.1.2. Techniki obrony

2.5.2. Cross-Site Script Inclusion (CSSI)

2.5.2.1. Przykładowy atak

2.5.2.2. Techniki obrony

2.5.3. Cross-Site Scripting (XSS)

2.5.3.1. Przykładowy atak

2.5.3.2. Techniki obrony

3. Ochrona danych poufnych, czyli kryptografia dla początkujących

3.1. Szyfrowanie jako metoda ochrony przed nieuprawnionym dostępem

3.1.1. Algorytmy szyfrujące

3.1.2. Jak poprawnie wybrać tryb działania szyfru blokowego?

3.1.3. Bezpieczny wybór: algorytm AES

3.1.4. Poznajemy bibliotekę i narzędzia OpenSSL

3.2. Kryptografia asymetryczna jako rozwiązanie problemu wymiany kluczy

BOTTEGA IT minds

3.2.1. Problem kryptografii symetrycznej: jak bezpiecznie wymieniać się kluczami?

3.2.2. Algorytmy stosowane w praktyce:

3.2.2.1. Teraźniejszość: algorytm RSA

3.2.2.2. Przyszłość: algorytm ECC

3.2.2.3. Czy przyszłość już nadeszła?

3.2.3. Infrastruktura klucza publicznego (PKI), czyli powiązanie klucza z jego właścicielem

3.3. Klucze, czyli dlaczego diabeł tkwi w szczegółach

3.3.1. Jak wygenerować dobry klucz?

3.3.2. Jak bezpieczenie przechowywać klucze?

3.4. Podpisy i uwierzytelnianie, czyli chronimy przed nieuprawnioną modyfikacją danych

3.4.1. Wykrywanie niepożądanych modyfikacji danych za pomocą kodów uwierzytelniających MAC

3.4.2. Weryfikacja tożsamości nadawcy i integralności wiadomości za pomocą podpisów cyfrowych

3.5. SSL, czyli zabezpieczamy połączenia sieciowe

3.5.1. Jak działa protokół SSL?

3.5.2. Certyfikaty SSL: o czym należy wiedzieć wybierając wystawcę certyfikatu

4. Podsumowanie

4.1. Bezpieczeństwo to proces

4.2. Co dalej?