

Ataki na aplikacje internetowe

Michał Kołodziejski



Dlaczego bezpieczeństwo systemów jest ważne?

22:26
18/11/2014

Włamanie na serwery Państwowej Komisji Wyborczej (wykradziono hashe haseł i klucze urzędników)

Autor: redakcja | Tagi: Hacked!, hasła, PKW, Polska, wybory

Za tym, że dane wyciekły z serwerów PKW.gov.pl przemawia także fakt, że na forum **Devil Team** opublikowano informacje o 2 podatnościach, XSS, oraz SQL injection (bazy danych zazwyczaj wykrada się właśnie przy pomocy dziury tego typu).

Dlaczego bezpieczeństwo systemów jest ważne?

Z punktu widzenia:

- użytkowników systemu
- zamawiającego system (klienta)
- wykonawcy systemu (firmy IT)

Niby wszystkim powinno zależeć na jak największym poziomie bezpieczeństwa, ale...

Garść statystyk

WhiteHat Website Security Statistics Report, 2013:

- „**86%** of all websites tested by WhiteHat Sentinel had at least one serious vulnerability”
- „The average number of serious vulnerabilities identified per website was **56**”

Tak wygląda źle zabezpieczony system...



Dlaczego tak wiele systemów jest „dziurawych”?

- „Niewidzialność” (nie)bezpieczeństwa...
... aż do pierwszej „wpadki”
- Brak zrozumienia wagi problemu przez klienta
- **Brak wiedzy u programistów i architektów**
- Pośpiech (harmonogram...)
- Brak audytów bezpieczeństwa

Zaatakujmy przykładową aplikację...

...wycieczka po aplikacji...

FakeBook - specyfikacja

Architektura:

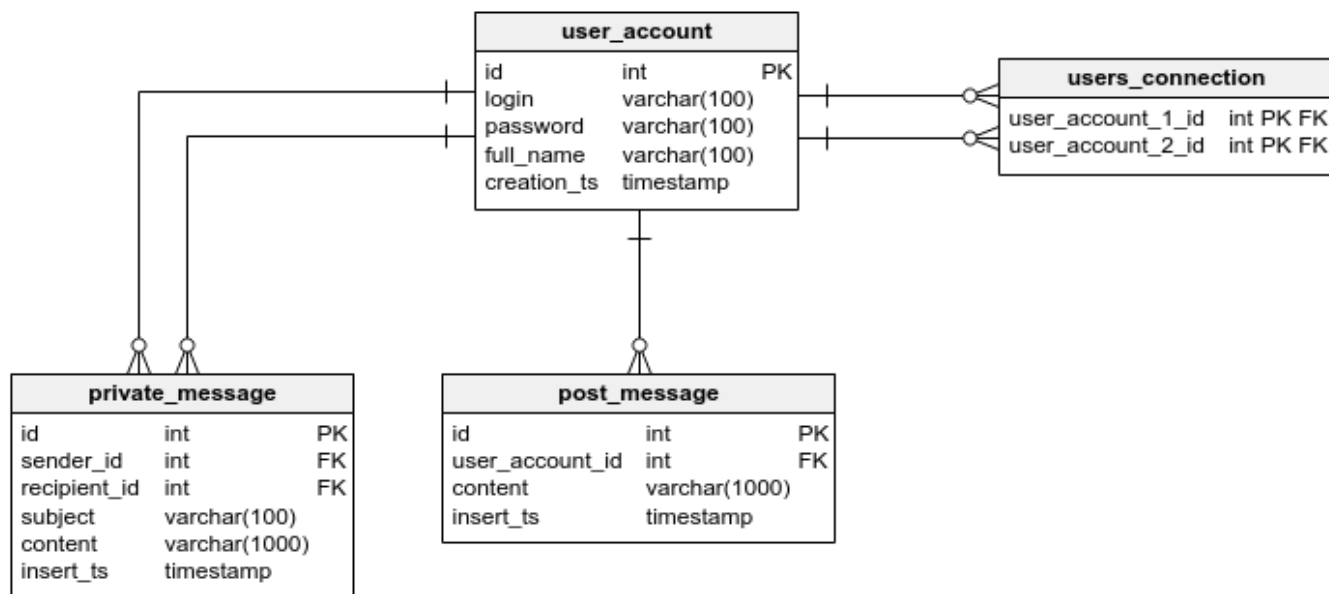
- JEE 6, brak frameworka
- „czyste” JDBC – brak ORM
- zewnętrzne biblioteki: Guice, Guava, Freemarker
- PostgreSQL 9.1
- JBoss AS 7.1

... **ale to nie ma znaczenia** – mógłby być np. PHP + MySQL

FakeBook - specyfikacja

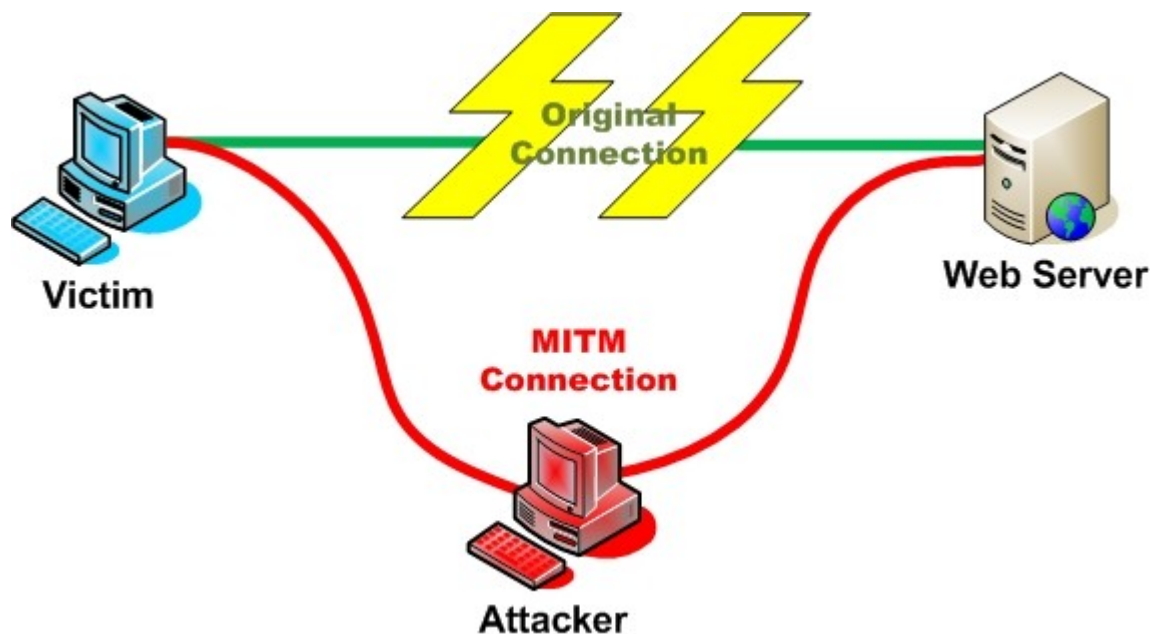
Baza danych:

- 4 tabele, 3 widoki, 3 sekwencje



Zaatakujmy przykładową aplikację...

1. HTTP vs. HTTPS



(źródło: owasp.org)

Inni też mieli z tym problemy



20:00
23/2/2010

Błędy w WP.pl i O2.pl pozwalają na podsłuchanie hasła do poczty

Autor: Piotr Konieczny | Tagi: atak, cookies, e-mail, GMail, hasła, O2.pl, Onet, poczta, web, WP.pl

(źródło: niebezpiecznik.pl)

DI > Wiadomości > Bezpieczeństwo

rozmiar tekstu: **A A A**  

Facebook z domyślnie włączonym HTTPS! Zobacz, czemu to ważne

Adrian Nowak, 20-11-2012, 18:56

(źródło: di.com.pl)

22:00
28/8/2014

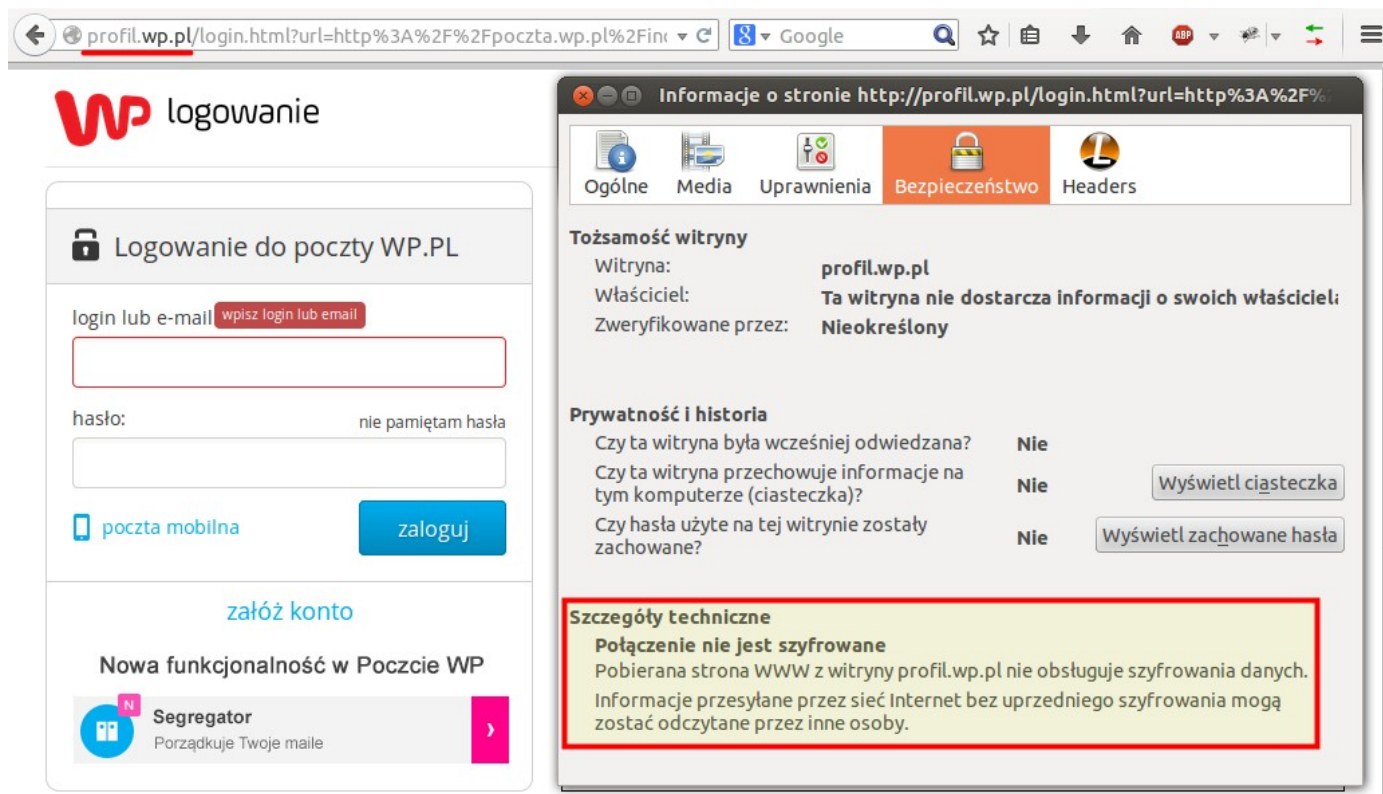
Poczta o2.pl nie zawsze wspiera HTTPS, czyli tzw. bezpieczne połączenie, co ułatwia podsłuchanie twoich e-maili

Autor: redakcja | Tagi: e-mail, fail, hasła, HTTPS, JS, o2, O2.pl, prywatność, wyciek

(źródło: niebezpiecznik.pl)

Konkurs

Strona logowania po HTTP, formularz wysyłany po HTTPS



Czy to w pełni bezpieczne i dlaczego nie?

Zaatakujmy przykładową aplikację...

2. SQL Injection

Zaatakujmy przykładową aplikację...

2. SQL Injection – jak się bronić:

- **Prepared Statements**

- Java

```
PreparedStatement stmt = connection.prepareStatement("select id, login, full_name from user_account" +  
    " where login = ?" +  
    " and password = ?");  
stmt.setString(1, login);  
stmt.setString(2, password);
```

- PHP

```
$stmt = $dbh->prepare("INSERT INTO REGISTRY (name, value) VALUES (:name, :value)");  
$stmt->bindParam(':name', $name);  
$stmt->bindParam(':value', $value);
```

- Python

```
params = ('sister', 'yellow')  
c.execute('SELECT * FROM users WHERE username=? AND room=?', params)
```

Zaatakujmy przykładową aplikację...

2. SQL Injection



(źródło: gizmodo.com)

Zaatakujmy przykładową aplikację...

3. Brak sprawdzenia uprawnień do zasobu

- OWASP: „**A4** - Insecure Direct Object References”
- **bardzo częsty** błąd, szczególnie wśród młodszych programistów

Zaatakujmy przykładową aplikację...

3. Brak sprawdzenia uprawnień do zasobu – jak się bronić:

- przeglądy kodu
- wyrobienie nawyku w programistach
- audyty bezpieczeństwa

Zaatakujmy przykładową aplikację...

4. HTML Injection

Zaatakujmy przykładową aplikację...

5. Cross-site scripting (XSS)

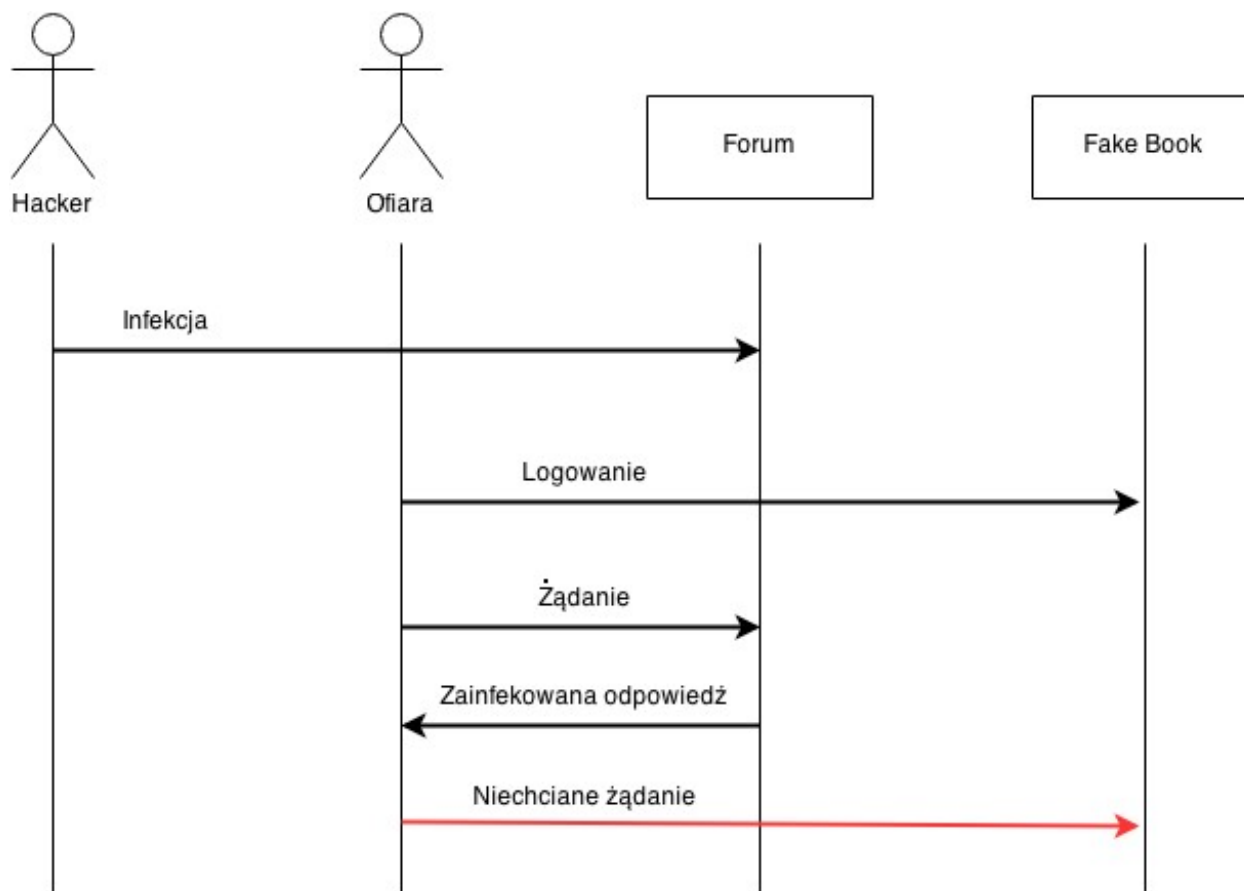
Zaatakujmy przykładową aplikację...

5. Cross-site scripting (XSS) – jak się bronić:

- zabezpieczenie cookie sesyjnego: HttpOnly, Secure
- uniemożliwienie używania tagów HTML przez użytkowników (formatowanie: BBCode, Markdown, ...)
- „escape”-owanie treści pochodzącej spoza serwisu
- Content Security Policy (CSP)

Zaatakujmy przykładową aplikację...

6. Cross-site request forgery (CSRF)



Zaatakujmy przykładową aplikację...

6. Cross-site request forgery (CSRF) – jak się bronić:

- sesyjny token CSRF
- weryfikacja tokenu przy żądaniach modyfikujących stan aplikacji
- dobra praktyka:
 - GET – pobranie zasobu
 - POST – modyfikacja zasobu (tu sprawdzamy token)

To nie wszystkie rodzaje ataków

Istnieje wiele innych ataków:

- brute force
- **Session Fixation (!)**
- Path Traversal (Directory Traversal)
- Remote File Inclusion
- XXE (XML External Entity Processing)
- ReDoS (Regex Denial of Service)
- Phishing
- ...

Ataki socjotechniczne

*„Najśłabszym ogniwem systemu zabezpieczeń jest **człowiek**”*

Kto jest **szczególnie** narażony na taki atak?

- pracownicy nieświadomi niebezpieczeństwa
- ludzie posiadający dostęp do kluczowych informacji lub elementów systemu (kodu, infrastruktury, etc.)
- ludzie posiadający dostęp do ludzi j.w.

Ataki socjotechniczne

Jak wygląda atak socjotechniczny?

- dogłębny wywiad środowiskowy
- powołanie się na osobę decyzyjną
- podszycie się pod osobę decyzyjną (e-mail, telefon)
- stosowanie perswazji
- stosowanie presji (np. presji czasu)
- stosowanie groźby (np. groźba zwolnienia)

Ataki socjotechniczne

Jak się bronić?

- przeszkolenie pracowników – w tym sekretarki!
- opracowanie procedur reakcji na takie zdarzenia
- przeprowadzenie audytu

Audyty bezpieczeństwa

Typy audytów:

- blackbox
- whitebox

Audyt może obejmować sprawdzenie bezpieczeństwa:

- kodu aplikacji
- architektury systemu (w tym np. integracji z aplikacjami mobilnymi)
- infrastruktury sieciowej
- konfiguracji (np. systemów operacyjnych, bazy danych)

Aspekty prawne

Art. 267 KK

§ 1. **Kto bez uprawnienia uzyskuje dostęp do informacji dla niego nieprzeznaczonej**, otwierając zamknięte pismo, podłączając się do sieci telekomunikacyjnej lub **przełamując albo omijając** elektroniczne, magnetyczne, **informatyczne** lub inne szczególne jej **zabezpieczenie**, **podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 2.**

§ 2. **Tej samej karze podlega, kto bez uprawnienia uzyskuje dostęp do całości lub części systemu informatycznego.**

§ 3. **Tej samej karze podlega, kto w celu uzyskania informacji, do której nie jest uprawniony, zakłada lub posługuje się urządzeniem podsłuchowym, wizualnym albo innym urządzeniem lub oprogramowaniem.**

Aspekty prawne

Art. 268 KK

§ 1. Kto, nie będąc do tego uprawnionym, niszczy, uszkadza, usuwa lub zmienia zapis istotnej informacji albo w inny sposób udaremnia lub znacznie utrudnia osobie uprawnionej zapoznanie się z nią, podlega grzywnie, karze ograniczenia wolności albo pozbawienia wolności do lat 2.

§ 2. Jeżeli czyn określony w § 1 dotyczy zapisu na informatycznym nośniku danych, sprawca podlega karze pozbawienia wolności do lat 3.

§ 3. Kto, dopuszczając się czynu określonego w § 1 lub 2, wyrządza znaczną szkodę majątkową, podlega karze pozbawienia wolności od 3 miesięcy do lat 5.

Dziękuję

Michał Kołodziejski
michal.kolodziejski@e-point.pl



Akademia e-point

11.12	AGILE w praktyce menadżera projektów IT Grzegorz Ścisło
Luty	Generatory kodu Bartłomiej Jańczak
Marzec	Blaski i cienie rekrutacji do firm informatycznych Marek Berkan
Marzec	Biznesowe aspekty projektowania aplikacji Wawrzyniec Hyska
Kwiecień	JavaScript na poważnie Tomasz Traczyk
Maj	Komunikacja w projekcie – kod, ludzie, procesy Dariusz Chojnacki



<https://github.com/mkolodziejski/webapps-security>

