Fakulta elektrotechniky a informatiky
Vysoká škola báňská - Technická univerzita Ostrava

Cvičení 2

POČÍTAČOVÁ OBRANA A ÚTOK - POU

SQL injection

- útok, ve kterém je vkládán nebezpečný kód do řetězců, které jsou později použity k SQL příkazům
- každá procedura, která konstruuje SQL příkaz by měla být nejdříve zkontrolována, protože SQL Server vykoná všechny validní příkazy, které se zde nachází
- některý škodlivý kód může být zanesen do tabulek k pozdějšímu použití
- často se používají dva znaky "--" pro komentář k ignorování zbytku původního příkazu

SQL injection - příklad

```
<?php
$query  = "SELECT * FROM products WHERE id = '$productId'";


$result = mysql_query($query);
?>
```

- $productId = 1’ **;drop table products --‘**
- SELECT * FROM products WHERE id=‘1’;

drop table products--’

SQL injection - přímý útok (Inband or inline)

- útoky používají stejný komunikační kanál jako vstup pro získání dat z database
- často se používají jako části zřetězených dotazů
  - **komentáře** (--, # a /*) - nejčastěji omezující podmínku WHERE
  - **skládání dotazů**
  - **konkatenace řetězců** - || (MySQL a ORACLE) a + (SQL Server)
  - **řetězce bez uvozovek** - pro obejití omezení na uvozovky
    - SELECT 0x457578
    - SELECT CONCAT(CHAR(75),CHAR(75),CHAR(75))
  - **spojení tabulek pomocí UNION**
  - **zneužití INSERT a UPDATE** - např. vytvoření nového uživatele

SQL injection – slepý útok (Inferential or Blind)

- dělí se na
  - normal blind (slepé)
    - zaměřené na aplikace, které nevypisují chybové hlášení, ale poskytují obecnou stránku s informací, že nastala chyba
  - totally blind (úplně slepé)
    - zaměřené na aplikace, které neposkytují žádný výstup

- aplikace nevrací žádné údaje, takže je potřeba použít jiný způsob kontroly odezvy – například se měří časová odezva aplikace
  - `WAITFOR DELAY 'h:m:s' - používá SQL Server`
  - `BENCHMARK(howMany, function()) - používá MySQL`
  - výstupem je například v´případě úspechu dotazu zpomalení aplikace o 10 vteřin

SQL injection – druhotný útok (Out-od-band)

- pokud aplikace přijímá vstup způsobem, který není náchylný na SQL injection a zároveň přijímá od uživatele data, z kterých nefiltruje klíčová slova SQL
- aplikace může zpracovat zadaná data a jejich výstup je pak prezentován jiným způsobem, například útočníkovi přijde email s výstupem

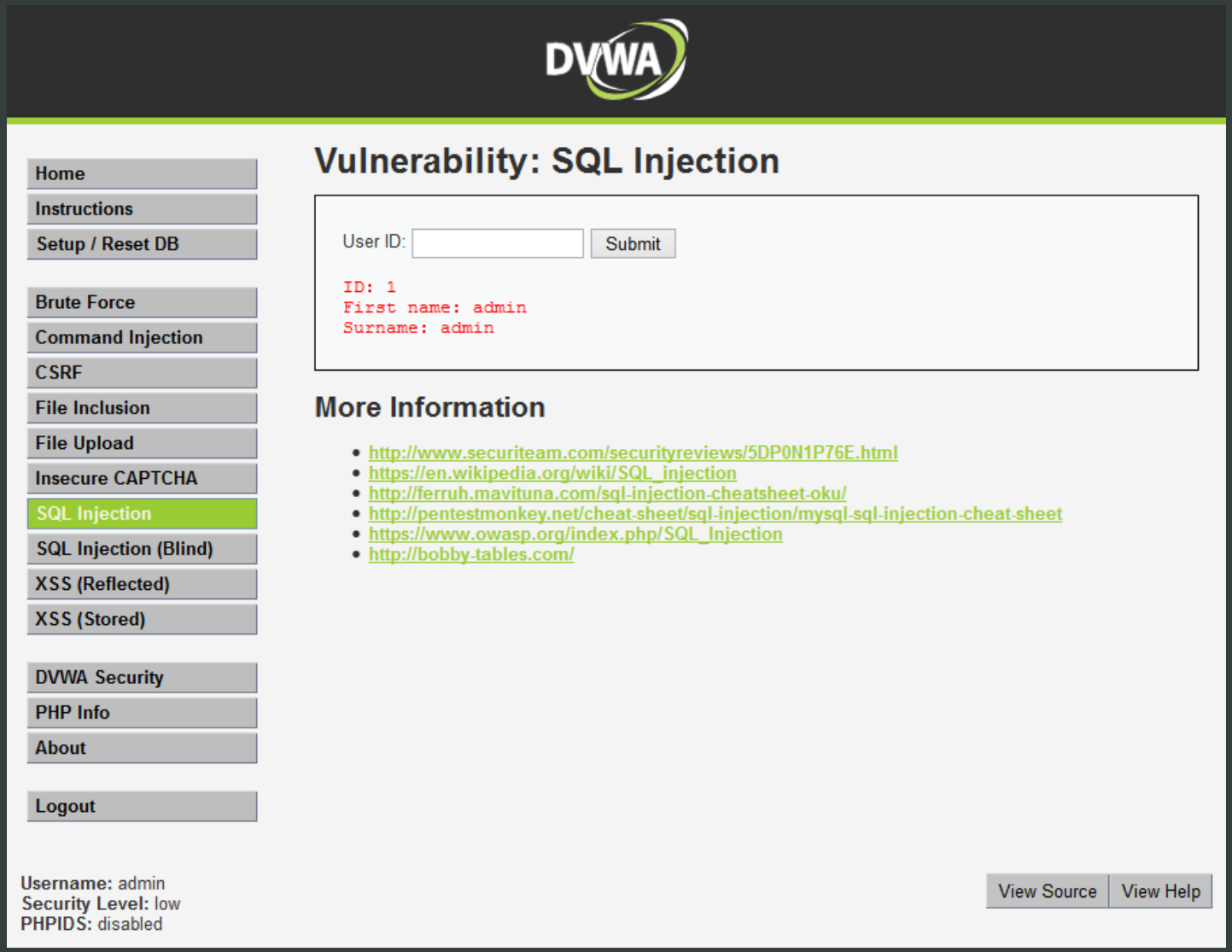
SQL injection typy

- Založené na **tautologii** (1=1 or “”=“” is always True)
  - SELECT * FROM Users WHERE UserId = 105 **or 1=1**
  - SELECT * FROM Users WHERE
    Name ="xxx" **or** ""="" AND Pass ="dunno" **or** ""=""

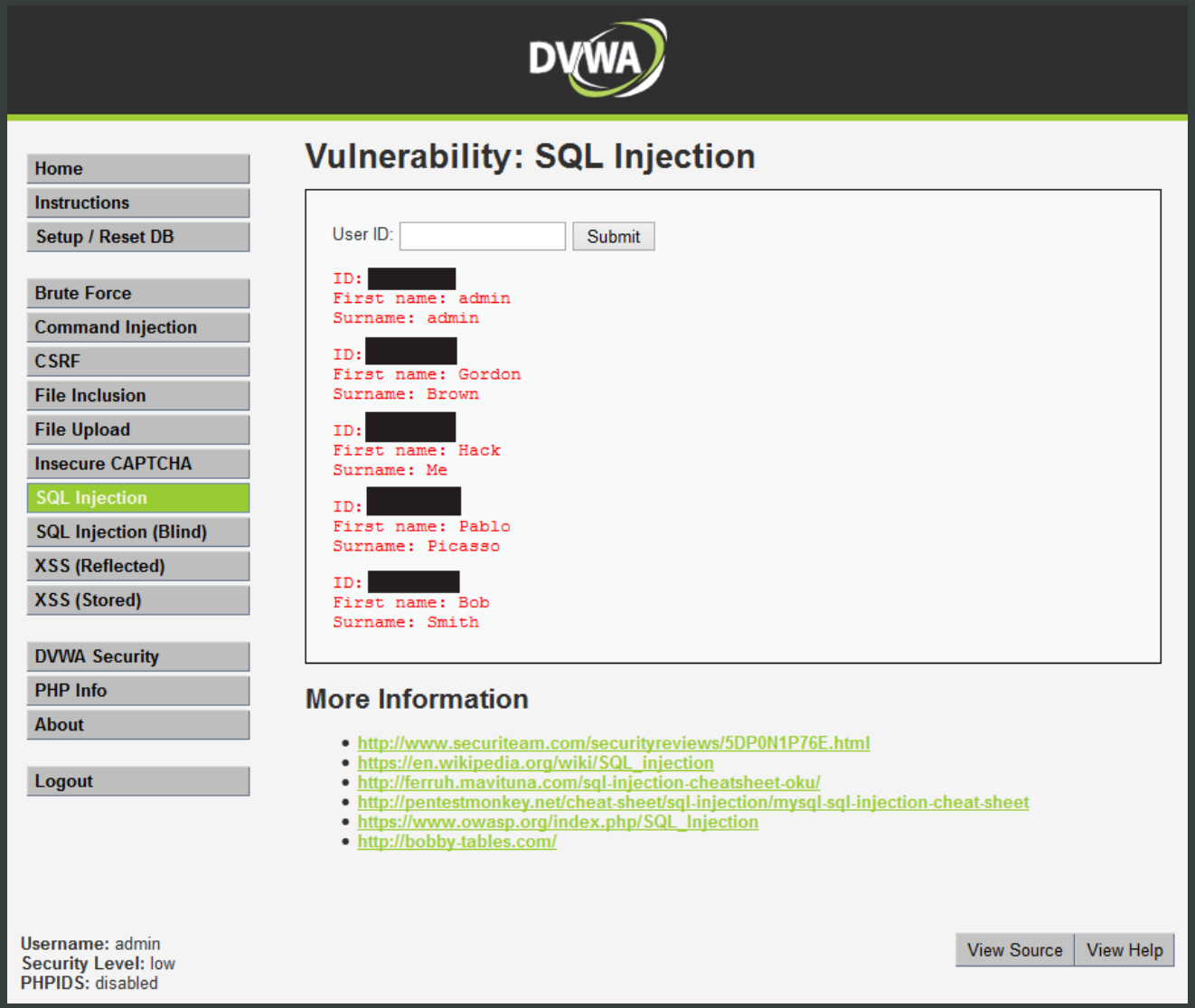
- Založené na **dávkovém zpracování**
  - SELECT * FROM Users WHERE UserId = 105; DROP TABLE Users

- Založené na **unionu**
  - SELECT name,price FROM Products WHERE id=1
    UNION SELECT userName, password FROM Users

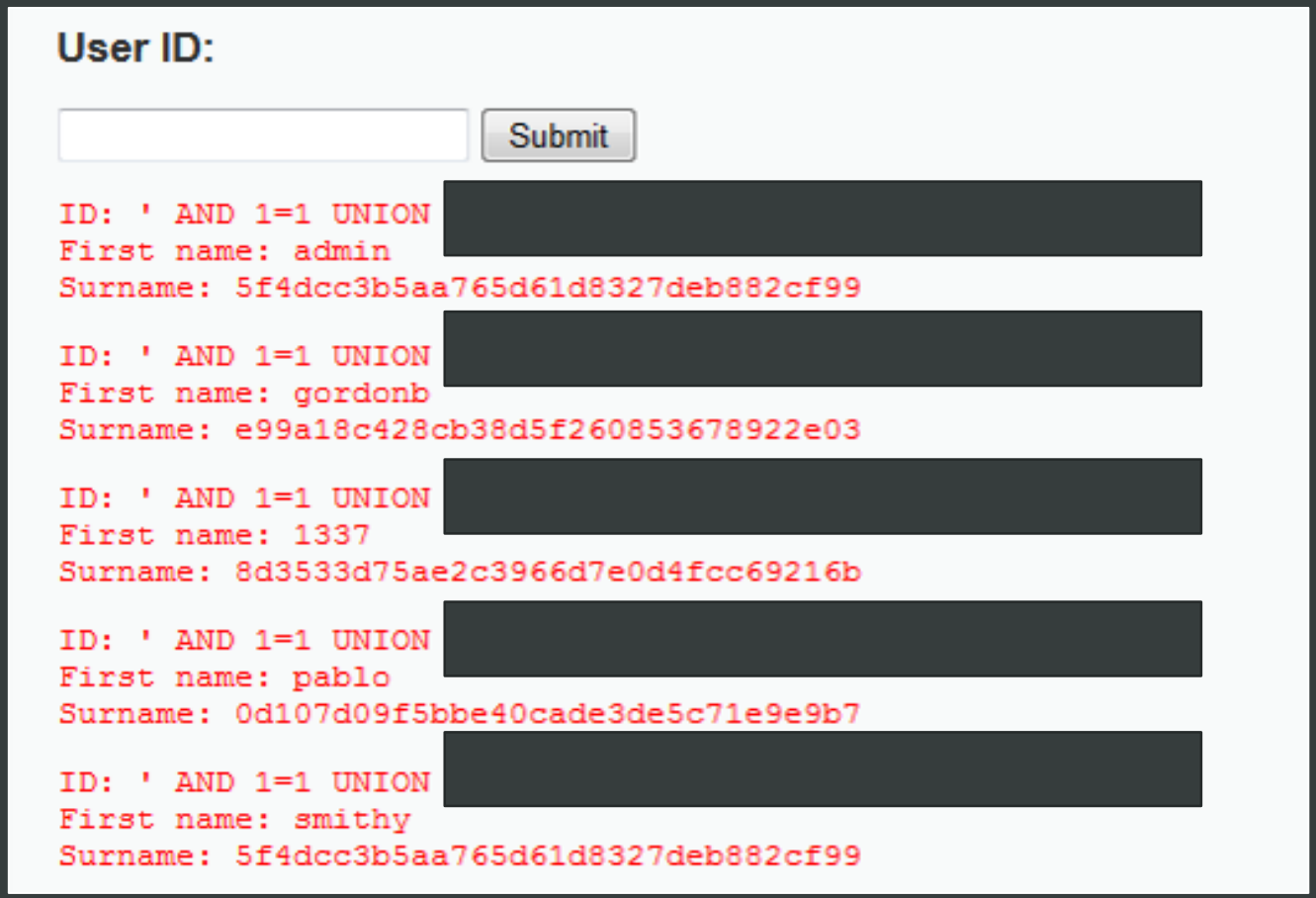
SQL injection – legitimní vstup



SQL injection – tautology condition

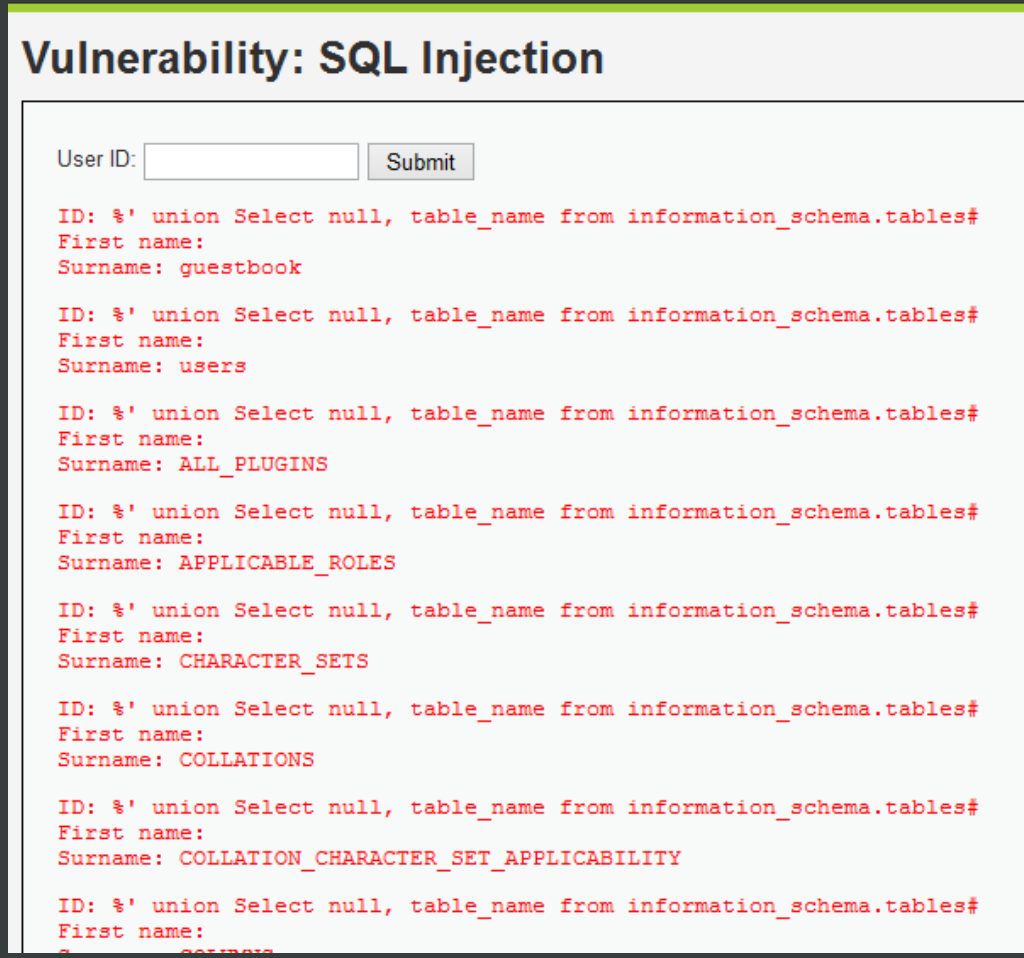


SQL injection – union condition



SQL injection – information_schema

- definovaná v SQL standardu
- nástroj k popisu metadat databáze
- množina pohledů, která vrací informace o tabulkách, uložených procedurách a pohledech
- information_schema.tables – informace o tabulkách v db
- information_schema.columns – informace o sloupcích tabulek v db
- information_schema.pg_user – informace o uživatelích db



SQL injection – information_schema.tables

- zobrazení tabulek db
- příkaz

```
%' union select null, table_name from
information_schema.tables#
```

- odpovídá SQL příkazu

```
SELECT first_name, last_name FROM users WHERE
user_id = '%' union select null, table_name from
information_schema.tables#'
```

Ochrana proti SQL injection

- Ochrana pomocí ošetření vstupů
- Parametrizace SQL dotazů
- Pravidlo least privilege
- zákaz čtení information_schema
- zákaz modifikace dat v tabulkách, kde to nevyžaduje aplikace

SQL injection - varování

- Hacking je nelegální, neboť se tímto způsobem snažíme dostat do systému, kterého nejsme vlastníkem
- Pokud chceme praktikovat legální hacking, pak leda na vlastním systému

SQL injection - protokol

Počítačová obrana a útok
Protokol z předmětu

Tématická oblast: SQL Injection

Přednášející: prof. Ing. Ivan Zelinka, Ph.D., Ing. Jan Plucar
Cvičící: Ing. Jiří Znoj

Jméno a číslo studenta:
Datum vypracování:

Zadání:

1) Seznamte se s teorií útoku typu SQL Injection
2) Proveďte praktický test útoku SQL Injection ve webové aplikaci DVWA (low protection).

Postup řešení:

1) **Otestujte legitimní vstup a možnost zadání škodlivého kódu**
2) **Získejte seznam všech uživatelů (např. tautology (always true) condition)**
3) **Získejte název a verzi databáze**
4) **Získejte seznam tabulek databáze (schématu) dvwa**
5) **Získejte seznam tabulek a jejich sloupců týkajících se uživatele (user)**
6) **Získejte seznam [jméno, příjmení, heslo, číslo kreditní karty, pin] všech uživatelů**
7) **Z disku vypište obsah soubor "c:/windows/system32/drivers/etc/hosts"**

Závěr:
Uveďte, krom výsledků, také skripty, které jste pro útoky použili a popište způsoby obrany