Bezpečnosť v databáze

- Základné pojmy
- Základy SQL dopytov
- Správa a zabezpečenie databázy
  - Pripojenie do DB
  - Používateľské účty a autentifikácia
  - Ukladanie prihlasovacích údajov
  - Oprávnenia
  - Konfigurácia
- Základy SQL Injection

Relačné databázové systémy

- Dáta v štrukturovanej forme v relačných tabuľkách
- Fixná schéma tabuliek
- Structured Query Language (SQL)
- Rôzne technológie systémov riadenia bázy dát (SRBD)
  - MySQL/MariaDB, PostgreSQL, MSSQL, Oracle, …
- Nasadenie
  - Rovnaký server, ako webová aplikácia (malé aplikácie, test)
  - Samostatný DB server, prípadne viacero serverov (škálovateľnosť, vysoká dostupnosť)
- Nerelačné databázy: NoSQL (neštrukturované dáta, Big Data)

Štruktúra SQL databázy

- Databáza
- Tabuľka (relácia)
- Stĺpce
- Špeciálne databázy
  - Informácie o databázach, tabuľkách, stĺpcoch, používateľských oprávneniach, etc.
  - Napr. INFORMATION_SCHEMA v MySQL

test.users

id	username	password
1	admin	admin123
2	alice	Pa55w0rd
3	bob	superman

SQL dopyty – CREATE

- Vytvorenie databázy

```
CREATE [OR REPLACE] {DATABASE | SCHEMA} [IF NOT EXISTS] db_name

CREATE DATABASE test;

USE test; # pripojenie k DB
```

- Vytvorenie tabuľky

```
CREATE [OR REPLACE] [TEMPORARY] TABLE [IF NOT EXISTS] tbl_name

    (create_definition,...)
```

  - create_definition – definícia stĺpcov

```
CREATE TABLE IF NOT EXISTS users (id bigint auto_increment primary key, username varchar(32), password
varchar(32));
```

- Odstránenie databázy tabuľky

```
DROP {DATABASE | TABLE} {db_name | tbl_name}

DROP DATABASE test;
```

SQL dopyty – SELECT

id	username	password	email
1	admin	admin123	admin@localhost
2	alice	Pa55w0rd	alice@mail.com
3	bob	superman	bob@mail.eu

```
SELECT * FROM users;

SELECT * FROM users WHERE username = 'admin';

SELECT username,password FROM users WHERE email = 'admin@localhost';

SELECT email FROM users WHERE username LIKE 'a%';
```

Správa a zabezpečenie databázy

- Zabezpečenie pripojenia
- Autentifikácia
- Ukladanie poverení
- Oprávnenia
- Bezpečná konfigurácia
- Poznámka: zameranie na MariaDB, princípy platia aj pre ostatné SRDB.

Pripojenie do databázy

- Lokálne
  - `mysql -u user -password test # nebezpečné`
  - heslo sa nachádza v histórii a taktiež v zozname bežiacich procesov!
  - `mysql -u root -p # výzva na zadanie hesla`
- Vzdialene
  - `mysql -h 192.168.1.20 -u user -p myapp`

Pripojenie do databázy - zabezpečenie

- Izolácia DB servera, čo najviac je to možné
  - Zákaz prístupu cez sieť (TCP)
  - Počúvanie iba na localhost
  - Obmedzenie prístupu cez sieť pomocou firewall-u
  - Samostatný sieťový segment/DMZ pre DB server, oddelený od aplikačného servera
- Obmedzenie prístupu k administračnému rozhraniu (napr. PHPMyAdmin)
- Šifrovanie TCP spojenia
  - TLS, silné šifry, ...

Používateľské účty a autentifikácia

- Vytvorenie používateľského účtu

```
CREATE USER account_name [authentication_option];
```

- account name
  - ‘user_name’@’host_name’ # povolenie prihlásenia zo stroja s hostname ‘host_name’
  - ‘user_name’@’192.168.1.0/24’ # povolenie prihlásenia zo sieťového rozsahu
  - ‘user_name’ # hodnota host_name bude ’%’ (wildcard – povolenie všetkých zdrojov)
- authentication_option
  - IDENTIFIED BY ‘password’ # v plaintexte, v DB bude uložené zahashované funkciou PASSWORD()
  - IDENTIFIED BY PASSWORD ‘password_hash’
  - IDENTIFIED VIA authentication_plugin
- Príklady:

```
CREATE USER ‘user’@’localhost’ IDENTIFIED BY ‘password’;

CREATE USER ‘foo’@’test’ IDENTIFIED VIA pam;
```

Používateľské účty a autentifikácia - zabezpečenie

- Vyžadovanie autentifikácie pre všetky prístupy (vrátane lokálneho)
- Používanie silných hesiel
- Samostatný používateľský účet pre každú aplikáciu
- Pravidelná revízia účtov
  - Odstránenie nepotrebných účtov (napr. zrušená aplikácia, odchod zamestnanca
  - Oprávnenia

Oprávnenia

- Pridelenie oprávnení

```
GRANT priv_type ON priv_level TO username;
```

- priv_type
  - ALL PRIVILEGES
  - globálne oprávnenia (administrácia DB, používateľov, etc.)
    - napr. `SHOW DATABASES, SHUTDOWN, GRANT OPTION, CREATE USER,...`
  - databázové oprávnenia (práca s konkrétnou databázou)
    - napr. CREATE, DROP, ...
  - tabuľkové oprávnenia (práca s konkrétnou tabuľkou)
    - napr. SELECT, CREATE, UPDATE, DROP, ...
- priv_level
  - `*.*` - globálne oprávnenia
  - `db_name.*` - databázové oprávnenia
  - `db_name.tbl_name` – tabuľkové oprávnanie
- Príklady:

```
GRANT ALL PRIVILEGES ON test.* TO 'user'@'localhost';

GRANT SELECT, UPDATE, DELETE, INSERT on eshop.items TO 'user'@'localhost';
```

- Zobrazenie oprávnení – SHOW GRANTS;
- Odobratie oprávnení – REVOKE

Oprávnenia - zabezpečenie

- Použitie princípu “least privilege”
- Nepoužívať vstavaný root účet
- Neprideľovať administrátorské oprávnenia na celú DB inštanciu
- Obmedzenie prístupu z konkrétnych host-ov
  - localhost, aplikačný server
- Pridelenie prístupu používateľom iba na konkrétne DB podľa potreby
- Pridelenie iba potrebných oprávnení v rámci DB
  - často postačuje povolenie SELECT, UPDATE, DELETE

Ukladanie prihlasovacích údajov

- Prihlasovacie údaje do DB
  - Neukladať v zdrojovom kóde / komentároch
  - V konfiguračnom súbore mimo adresára web root
  - Zabezpečenie prístupu pomocou vhodných oprávnení
  - Neukladať v repozitároch
- Heslá do aplikácie v DB
  - Neukladať v plaintext-e
  - Použitie bezpečnej hashovacej funkcie a soli
  - Odporúča sa kontrola voči zoznamom uniknutých hesiel

Zabezpečenie konfigurácie

- Podkladový OS by mal byť zabezpečený a aktualizovaný
- Pravidelný inštalácia bezpečnostných aktualizácií
- DB služba by mala bežať pod samostatným používateľom s nízkymi oprávneniami
- Odstránenie preddefinovaných a testovacích účtov a databáz
- Pravidelné zálohy, ideálne zašifrované
- Ukladanie a kontrola auditných záznamov

SQL Injection

- Zraniteľnosť SQL Injection vzniká v prípade, že vstup od používateľa je v aplikácii zapracovaný do SQL dopytu do DB.
- Vkladanie metaznakov
- Injektovanie validných SQL príkazov

```
$login = $_GET["login"];

$pw = $_GET["pw"];

$query = "SELECT * FROM users WHERE
login='$login' AND password='$pw'";

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

SQL Injection – príklad 1

- Narušenie dopytu použitím metaznakov (znakov rezervovaných v SQL)
  - ‘, “, `,(,), \, --, /*, */, #, %
- Dopyt skončí s chybou
  - chybová hláška môže prezradiť ďalšie informácie

```
SELECT * FROM users WHERE login='$login' AND
password='$pw'";

http://www.stranka.sk/login.php?login='

SELECT * FROM users WHERE login='' AND password='$pw'";
```

error: You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near "'" at line 1
client_info: mysqlnd 5.0.12-dev - 20150407 - $Id: 38fea24f2847fa7519001be390c98ae0acafe387 $
host_info: Localhost via UNIX socket

) Query: SELECT username FROM accounts WHERE username='"; (0) [Exception]

SQL Injection – príklad 2

- Zmena pôvodného dopytu vložením komentárov

```
SELECT * FROM users WHERE login='$login' AND password='$pw'"
```

```
http://www.stranka.sk/login.php?login=admin'-- &pw=123
```

```
SELECT * FROM users WHERE login='admin' -- ' AND
password='123'"
```

Poznámka: Rôzne SRBD sa môžu pri komentároch správať rozdielne: napr. MariaDB/MySQL vyžaduje za "--" znak medzery, Oracle a SQL Server nie

Poznámka2: Pri vkladaní špeciálnych znakov do URL je potrebné ich zakódovanie, napr. medzera → +

SQL Injection – príklad 3

- Zmena pôvodného dopytu použitím tautológie

```
SELECT * FROM users WHERE login='$login' AND
password='$pw'
```

```
http://www.stranka.sk/login.php?login=' OR 1=1-- &pw=123
```

```
SELECT * FROM users WHERE login='' OR 1=1-- ' AND
password='123'"
```

Poznámka: Pri vkladaní špeciálnych znakov do URL je potrebné ich zakódovanie, napr. medzera → +

UNION based SQL Injection

- Operátor UNION umožňuje zlúčiť výsledky viacerých SELECT dopytov
- Podmienka: rovnaký počet stĺpcov vo výstupe
- Postup:
  1) Zistenie počtu stĺpcov
  2) Identifikácia stĺpcov vo výstupe
  3) Pripojenie požadovaných informácií

UNION based SQL Injection - príklad

myapp.users

id	username	password	email
1	admin	admin123	admin@localhost
2	alice	Pa55w0rd	alice@mail.com
3	bob	superman	bob@mai

myapp.articles

authorId	name	title	text
1	John	Java How-to	...
2	Jack	Cracking passwords	...

- `SELECT * FROM articles WHERE authorId=$aid`
- `http://www.myapp.sk/article.php?authorId=1 UNION ALL SELECT * FROM users; --`
- `SELECT * FROM books WHERE authorId=1 UNION ALL SELECT 1, 2, 3, 4 FROM users; --`
- `SELECT * FROM books WHERE authorId=1 UNION ALL SELECT 1, username, password, 4 FROM users; --`

authorId	name	title	text
1	John	Java How-to	...
1	admin	admin123	admin@localhost
2	...		

Obrana proti SQL Injection

- **Použitie ORM**
- Prepared statements (parametrizovanie dopytov)
- Stored procedures
- Validácia vstupu na základe whitelist-u
- Ošterenie všetkých vstupov od používateľa
- Princíp “Least privilege”

Zdroje

- SQL Statements
  - https://mariadb.com/kb/en/sql-statements/
- SQL Language Structure
  - https://mariadb.com/kb/en/sql-language-structure/
- Configuring MariaDB with Option Files
  - https://mariadb.com/kb/en/configuring-mariadb-with-option-files/
- Database Security Cheat Sheet
  - https://cheatsheetseries.owasp.org/cheatsheets/Database_Security_Cheat_Sheet.html
- SQL Injection Prevention Cheat Sheet
  - https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html