Name:Ahmed Almass

Lab 6: Attacking Data Stores and Back-End Components

Web Application Security

Contents

[Lab Outcome 3](#_Toc524601709)

[Background Reading 3](#_Toc524601710)

[Required Hardware/Software 3](#_Toc524601711)

[Introduction 3](#_Toc524601712)

[1.0 Numeric SQL Injection 3](#_Toc524601713)

[2.0 String SQL Injection 3](#_Toc524601714)

[3.0 SQL Injection 3](#_Toc524601715)

[4.0 Blind Numeric SQL Injection 4](#_Toc524601716)

[5.0 Blind String SQL Injection 4](#_Toc524601717)

[6.0 Log Spoofing 4](#_Toc524601718)

[7.0 Command Injection 4](#_Toc524601719)

[8.0 Create a SOAP Request 4](#_Toc524601720)

[9.0 WSDL Scanning 4](#_Toc524601721)

[10.0 Web Service SQL Injection 4](#_Toc524601722)

[11.0 Sign-Off – Lab 6: Attacking Data Stores and Back-End Components 5](#_Toc524601723)

Web Application Security

Lab 6: Attacking Data Stores and Back-End Components

# Lab Outcome

Exploit the web application data stores and back-end components using various injection techniques.

Background Reading

Read the textbook sections listed in the Course Schedule.

Required Hardware/Software

* WebGoat v7.1
* Burp

# Introduction

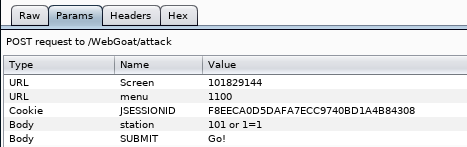
By attacking the data store, you can access web application data, including user accounts, permissions, configuration settings and everything else stored on a system.

In addition, web applications are often the access control to back-end components, such as filesystems and the operating system, as well as any networked resources like web servers and mail servers. Attacking the back-end can provide access to sensitive data and functionality.

# 1.0 Numeric SQL Injection

In WebGoat, complete the **Injection Flaws > Numeric SQL Injection** lesson.

After Intercepting with burp suit I replaced the station with 101 with 101 or 1=1.

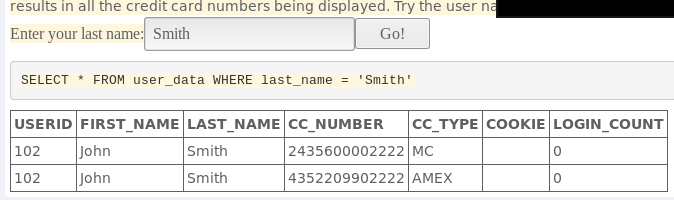


Which prints all the data.

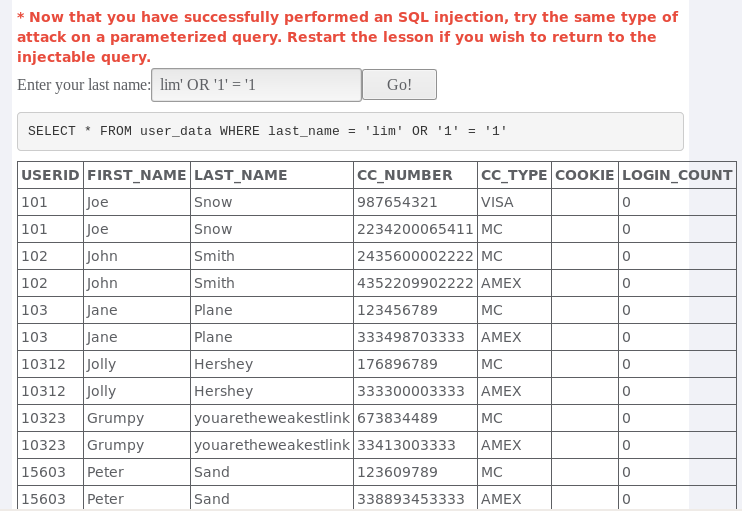
# 

# 2.0 String SQL Injection

In WebGoat, complete the **Injection Flaws > String SQL Injection** lesson.

When typing lastname it brings out credit card number.

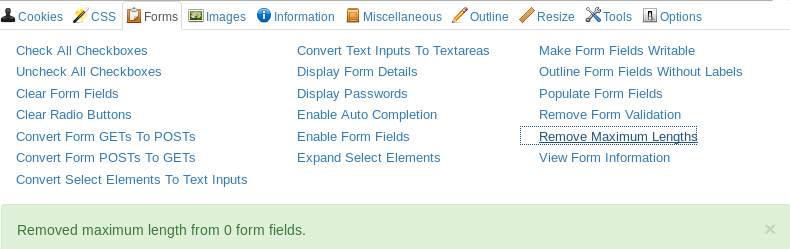
Entering a string that is single quotes sql injection that gives out all the data.



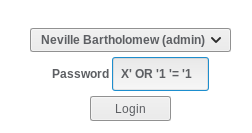
# 3.0 SQL Injection

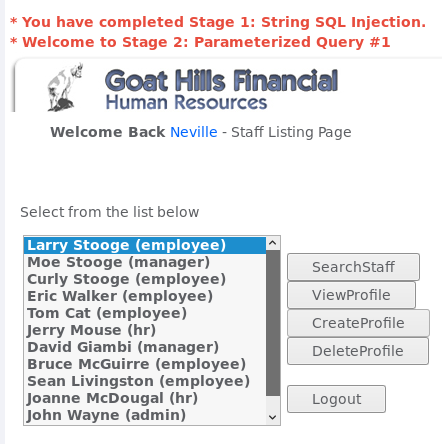
1. In WebGoat, complete the **Injection Flaws > SQL Injection Stage 1** lesson.

I disabled the max string length for the password using web devoloper.



That knowing the admin user I inserted an sql injection to login.



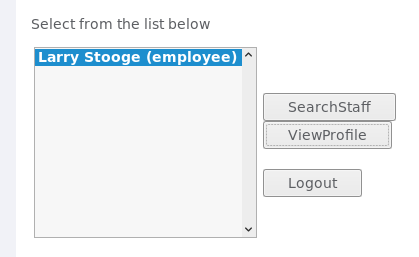


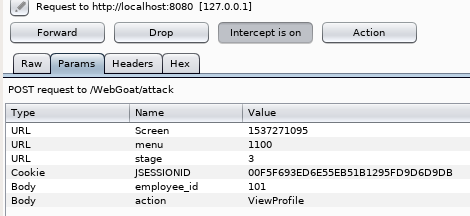
1. In WebGoat, complete the **Injection Flaws > SQL Injection Stage 2** lesson.

Need webgoatdev

1. In WebGoat, complete the **Injection Flaws > SQL Injection Stage 3** lesson.

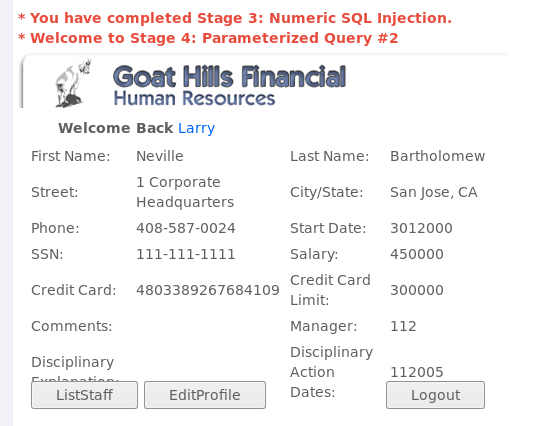
When loging in to a regular user, I intercepted it with burp suite.





I than changed the Value to an sql command to view all users including the admin.





1. In WebGoat, complete the **Injection Flaws > SQL Injection Stage 4** lesson.

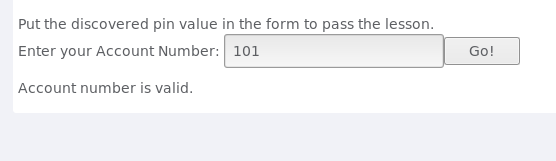
Need webgoatdev

# 4.0 Blind Numeric SQL Injection

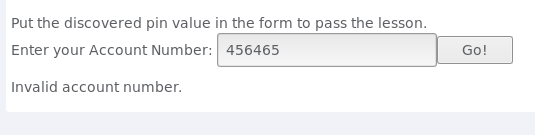
In WebGoat, complete the **Injection Flaws >** **Blind Numeric SQL Injection** lesson.

For this lesson logical operators has to be understood.

Figure below shows that a correct Account number it prints out “ Account number is valid.”



This Figure shows if you put incorrect Account number it prints out “ Invalid account number”.

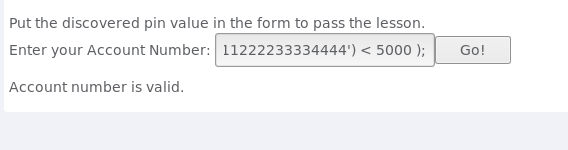


This means that these values have a True or False statement. In which True the value of of the interger exist and for False it doesnt exist.

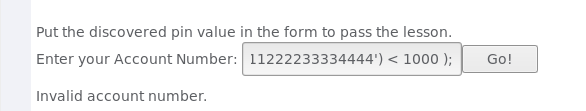
The data base query being used is “ SELECT\*FROM user\_data WHERE userid=accountNumber; in which if the account exit or not.

By using the AND function, we can add a second condition. By using this command we are 101 AND ((SELECT pin FROM pins WHERE cc\_numbers=’1111222233334444’) < [int-value ]);

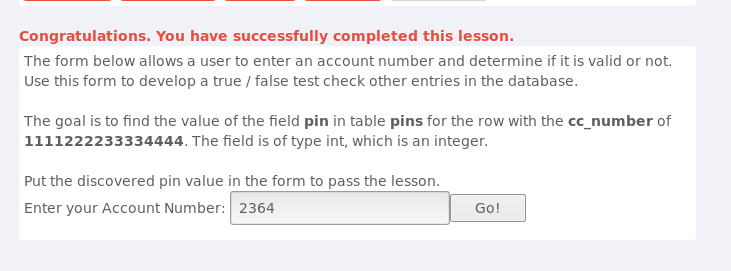
in which will create a loop with a increment condition if its true or not. By replace int-value by 5000 it will show if the account is valid meaning its within that range.



In this figure below i set the int-value to 1000 which is invalid.



Knowing that the range is in between 5000 to 1000 you can eventually find the accurate number by changing the values which will eventually become to 2364



# 5.0 Blind String SQL Injection

In WebGoat, complete the **Injection Flaws > Blind String SQL Injection** lesson.

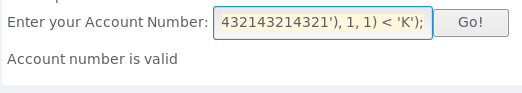
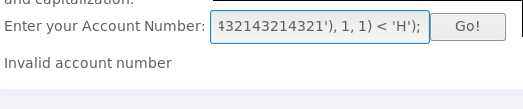
In this Lesson, this is similar to the previous lesson; the only difference is that we are looking for a string not a interger.

Using te same method for data base query, I will be using 101 AND (SUBSTRING((SELECT name FROM pins WHERE cc\_numbers=’4321432143214321’), 1 ,1) < ‘H’);

were the SUBSTRING method is being used which compares the the letters using the boolean to see if its true or false.

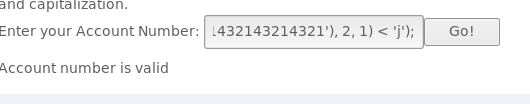
By inputing this:

101 AND (SUBSTRING((SELECT name FROM pins WHERE cc\_numbers=’4321432143214321’),1,1) < ‘H’);

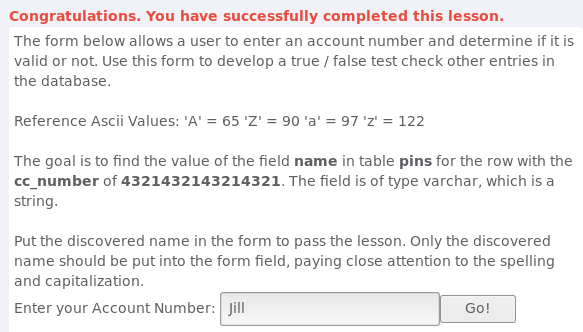
and changing the H with K; will find the range where the first letter would be.

By figureing out the second letter we will have to change the command by changing he 1 to 2 to compare the second letter.

101 AND (SUBSTRING((SELECT name FROM pins WHERE cc\_numbers=’4321432143214321’), **2** ,1) < ‘letter’);



By continuing this process you’ll eventually find the rest of the letter.



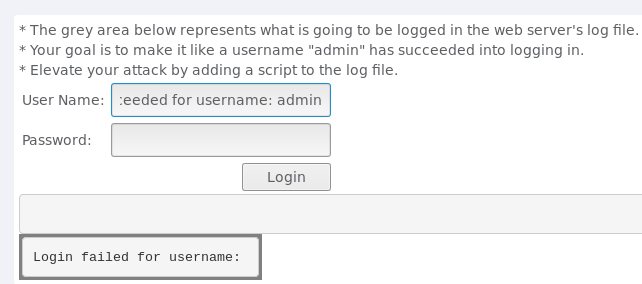
# 6.0 Log Spoofing

In WebGoat, complete the **Injection Flaws > Log Spoofing** lesson.

The objective of this lesson is to inject commands to webserver log file.

By injecting “Smith%0d%0aLogin Succeded for username: admin” into input user.

By doing this I inputed in the logfile that the the admin successfully loggin.





Also with this vulnerbility its possible to inject maliouse script in the log-file.

# 

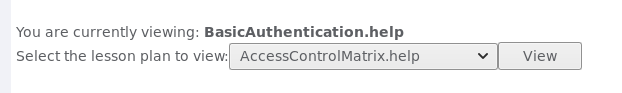
# 

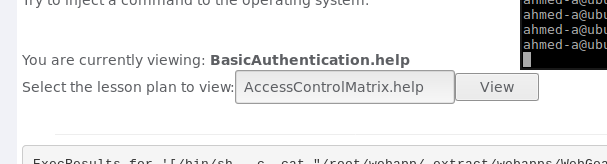
# 7.0 Command Injection

In WebGoat, complete the **Injection Flaws > Command Injection** lesson.

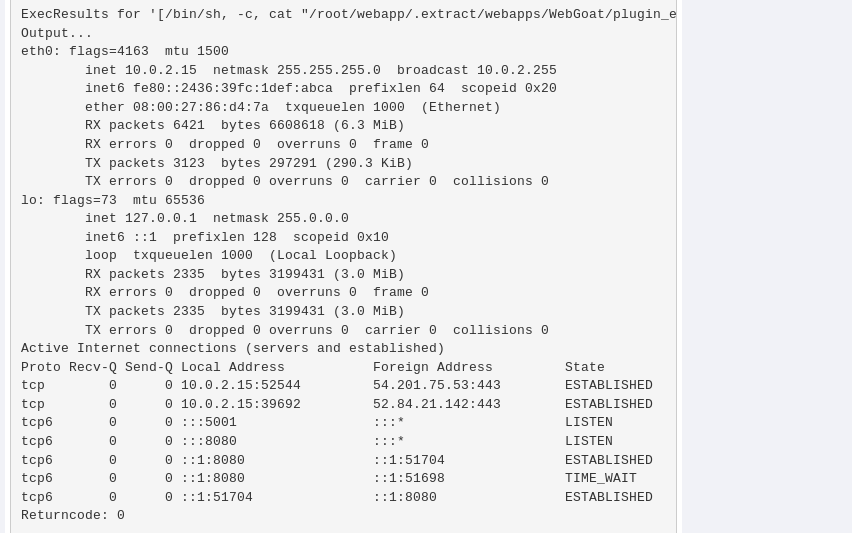
In this lesson, the goal is to inject OS command to the host.

1st I converted the elements to text inputs with web devolper tool extension on firefox.





2nd I appended “ & netstat -ant ifconfig” to the text input. Which gave me the following output.

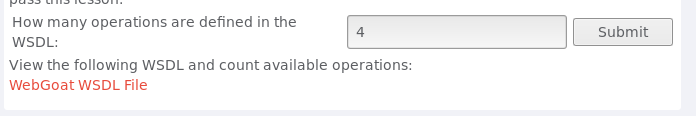


# 8.0 Create a SOAP Request

In WebGoat, complete the **Web Services > Create a SOAP Request** lesson.

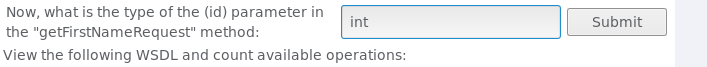
Stage 1:



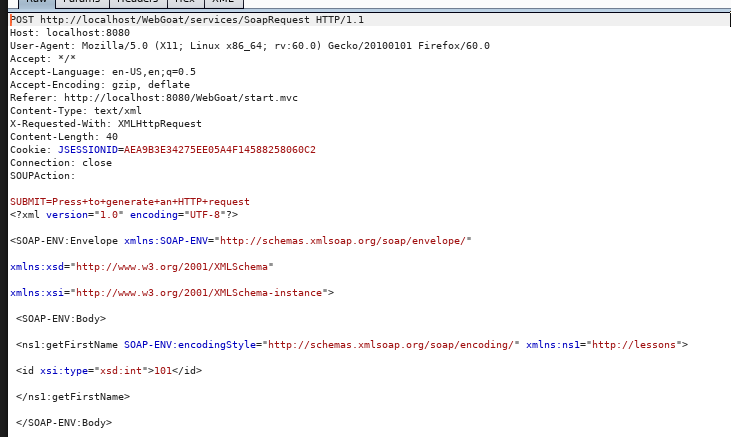


Stage 2:





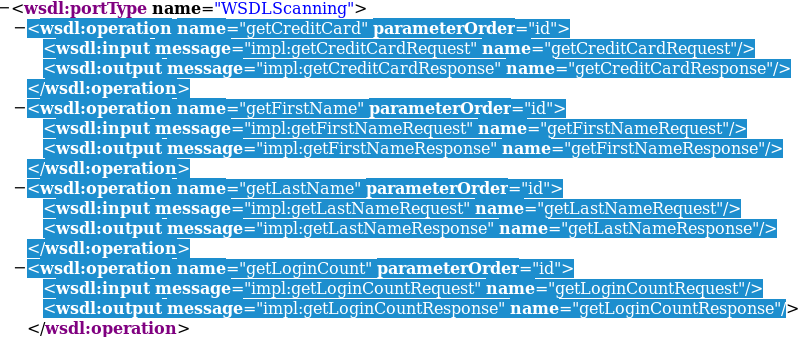
Stage 3



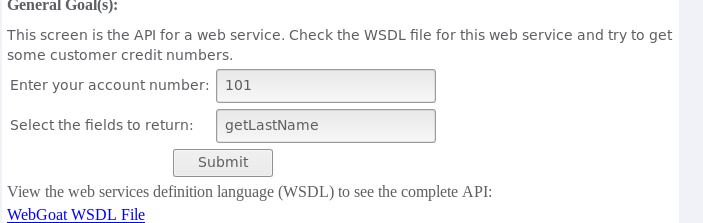
# 9.0 WSDL Scanning

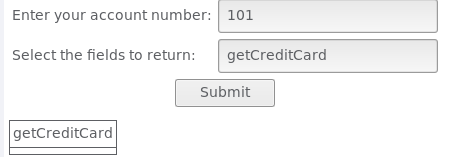
In WebGoat, complete the **Web Services > WSDL Scanning** lesson.

The objetive is to fnd the creditcard number. Checkng the operation name from the code



Used Webdev tool to change element to text





For some reason it didnt show the number.

# 10.0 Web Service SQL Injection

In WebGoat, complete the **Web Services > Web Service SQL Injection** lesson.

Didnt work for me

# 11.0 Sign-Off – Lab 6: Attacking Data Stores and Back-End Components

Detach this page and submit it to your instructor to indicate you have completed all sections.

Name:

Student ID:

|  |  |
| --- | --- |
| **Section** | **Instructor Initials** |
| 1.0 Numeric SQL Injection |  |
| 2.0 String SQL Injection |  |
| 3.0 SQL Injection – 1 |  |
| 3.0 SQL Injection – 2 |  |
| 3.0 SQL Injection – 3 |  |
| 3.0 SQL Injection – 4 |  |
| 4.0 Blind Numeric SQL Injection |  |
| 5.0 Blind String SQL Injection |  |
| 6.0 Log Spoofing |  |
| 7.0 Command Injection |  |
| 8.0 Create a SOAP Request |  |
| 9.0 WSDL Scanning |  |
| 10.0 Web Service SQL Injection |  |