Binghamton University, Watson School of Engineering

Project 4:

SQL Injection Deception

Name: Rahul Verma

Science of Cyber Security – CS 559-01

Prof. Guanhua Yan

December 05, 2022

<u>**Contents**</u>

SQL INJECTION DECEPTION

1. Previous Work:

There are four projects, all of which center around the topic of SQL injection. The first project focuses on the offense demonstration component of the attack, the second on the detection of the SQLI attempt, the third on the implementation of prevention mechanisms to counter the SQLI attack, and this project, which is the fourth and final project, is based on deceiving the attacker rather than preventing the attack. The following is a summary of each project:

The first project consisted of creating a website that was vulnerable to SQLI for demonstration purposes. Here we came across the process of SQLI in action when the attacker enters the input username and password as (1' or '1' = '1' and user<>'user1). The SQL statement or query in the source code gets manipulated and becomes (SELECT * FROM userpass WHERE user='1' or '1' = '1' and user<>'user1'AND pass='1' or '1' = '1' and user<>'user1') and he gets unauthorized access to the database and all of its contents.

The second project consisted of creating a mechanism that detects the SQLI attempts even before the prevention mechanism is used. For this, we used Naïve Bayes Classifier, which was based on the Bayesian decision theory, and classified the input username or password as SQL injection string or legit string. We provided a training set for helping the classifier with the classification of the input string. In the end, it is displayed if SQLI was detected or not.

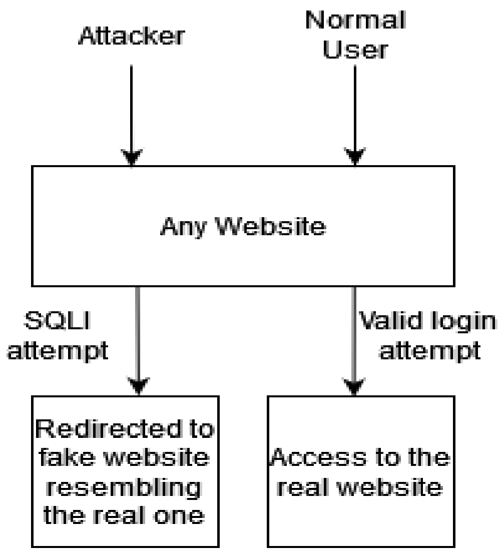
The third project consisted of creating three prevention mechanisms, namely: prepared statements, mysqli_real_escape_string() and input validation. In Prepared statements, we write queries without providing the parameters at the beginning, it is provided later. In mysqli_real_escape_string(), we circumvent the most used thing in SQLI which is special characters. In input validation, only special characters listed as being permitted are allowed.

In this project, which is the fourth and last project, we implement a deception mechanism where rather than blocking the attacker from accessing the database, we redirect him to a fake website that he believes to be legitimate but isn't. He thinks he has obtained access to the server, but all he is doing is visiting a fake website we set up so we can keep track of him all the time and monitor his actions. This mechanism is essential because an attacker who obtains access to susceptible data may disclose it online.

2. <u>Introduction to Deception in SQLI:</u>

Deception in SQLI is, tricking the attackers using lots of fake components throughout the website that resembles real components which here is the fake website("fakevalid.php") the attacker gets redirected to. We can keep track of the attacker and all the methods he used to get the access and take advantage of any vulnerabilities. We can then use all the analyzed information to make the system more secure and reduce the future risks of SQLI attacks.

3. <u>Why is Deception required in SQLI:</u>


Attacker
Normal User
Any Website
SQLI attempt
Valid login attempt
Redirected to fake website resembling the real one
Access to the real website

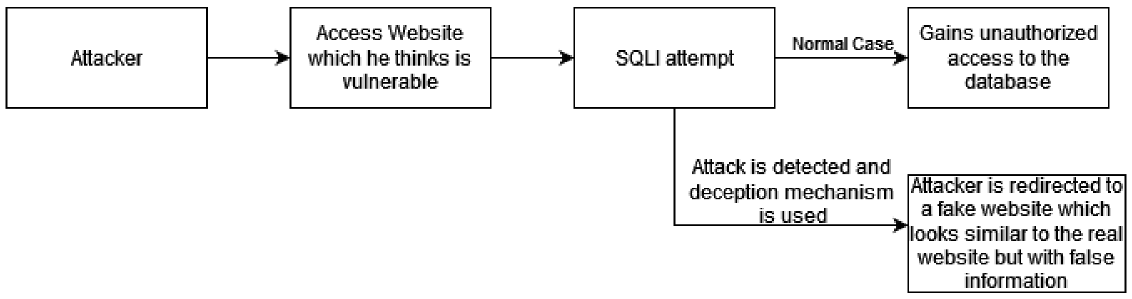
As we can see from the diagrammatic representation above, there are 2 cases:

The first case is the normal case where a user tries to login using the correct username and password in a login form on a particular website. On successful login, the user will be redirected to the account information webpage. Sometimes the user will not be able to enter the information correctly or may enter incorrect information in the login form. In that case, the user will be notified that the entered information is incorrect and is sent back to the login form to attempt the login again.

The second case is where the attacker will try to access the website thinking that it's vulnerable, but as we have already implemented the deception mechanism, this won't be the case. Here the attacker enters an SQL injection query in the login form so that he can access the database consisting of all the information of the users and will try to leak it online or sell the information. But the attacker is redirected to a fake website that resembles the real one.

We can do whatever we can to track the attacker, we can note down the IP Address of the attacker and we can notify the higher authorities to take the required action. This is done so that we can waste as much time of the attacker as possible in exploring the things that we have set up and in the meanwhile, we can analyze the information we have on the attacker and track him.

4. Steps involved in Deception and the end goal:


Attacker
Access Website which he thinks is vulnerable
SQLI attempt
Normal Case
Gains unauthorized access to the database
Attack is detected and deception mechanism is used
Attacker is redirected to a fake website which looks similar to the real website but with false information

The following steps are involved in the process of Deception in SQLI:

Step 1: The attacker accesses the website.

In this step, the attacker tries to access the website and test out if it's vulnerable using his own methods. This can result in various kinds of attacks on the website, one of which can be SQLI attack.

Step 2: Tries to attempt SQLI.

The attacker then tries to attempt SQLI attack by entering various SQLI queries, which can manipulate the source code and give the attacker complete access to the database containing all the user account information.

Step 3: Thinks he has access to the database information.

Once the attacker attempts SQLI, he then gets access to all the information that he thinks is real but is in fact fake information kept in place.

Step 4: Gets redirected to the fake webpage.

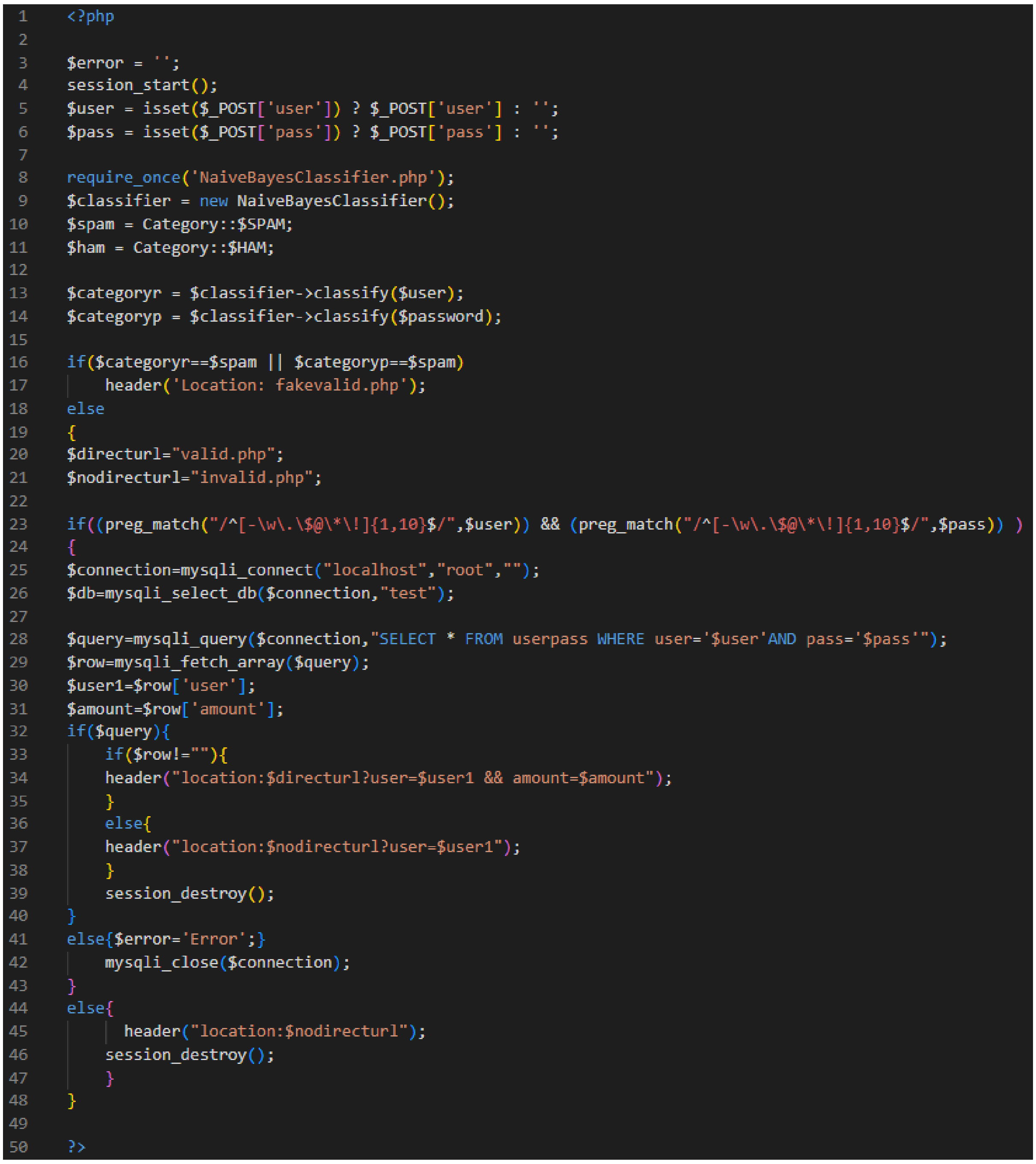
The attacker here is redirected to a fake webpage where he is monitored continuously, and all the displayed information is fake and will result in no loss to any user or organization.

Step 5: Information is noted of the attacker.

We can take note of the attacker's IP Address, notify the higher authorities about the attack, monitor all the actions being performed by the attacker etc.

The end goal is to waste the time of the attacker as much as possible by letting him get fake information or data, improve the defense mechanism by analyzing the attacker's way of finding the vulnerability on the website by tracking the attacker's actions as soon as he is redirected to the fake website. The tracking data can also be utilized to determine the type of data the attacker is seeking.

5. <u>Working and implementation of Deception mechanism in SQLI:</u>



```php
<?php

$error = '';
session_start();
$user = isset($_POST['user']) ? $_POST['user'] : '';
$pass = isset($_POST['pass']) ? $_POST['pass'] : '';

require_once('NaiveBayesClassifier.php');
$classifier = new NaiveBayesClassifier();
$spam = Category::$SPAM;
$ham = Category::$HAM;

$categoryr = $classifier->classify($user);
$categoryp = $classifier->classify($password);

if($categoryr==$spam || $categoryp==$spam)
    header('Location: fakevalid.php');
else
{
$directurl="valid.php";
$nodirecturl="invalid.php";

if((preg_match("/^[-\w\.\$@\*\!]{1,10}$/",$user)) && (preg_match("/^[-\w\.\$@\*\!]{1,10}$/",$pass)) )
{
$connection=mysqli_connect("localhost","root","");
$db=mysqli_select_db($connection,"test");

$query=mysqli_query($connection,"SELECT * FROM userpass WHERE user='$user'AND pass='$pass'");
$row=mysqli_fetch_array($query);
$user1=$row['user'];
$amount=$row['amount'];
if($query){
    if($row!=""){
    header("location:$directurl?user=$user1 && amount=$amount");
    }
    else{
    header("location:$nodirecturl?user=$user1");
    }
    session_destroy();
}
else{$error='Error';}
    mysqli_close($connection);
}
else{
      header("location:$nodirecturl");
    session_destroy();
    }
}

?>
```

As shown above, we have successfully implemented the deception mechanism, the explanation is as follows:

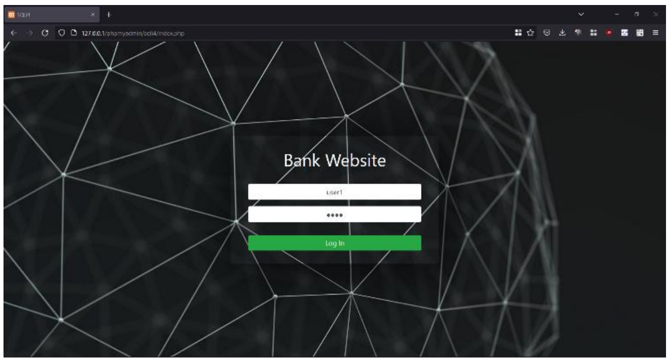
On line 8, we use “NaiveBayesClassifier.php” for the pre-existing implementation of the detection mechanism used in detecting the SQLI attempts, we use it on both the inputs, i.e., username and password.

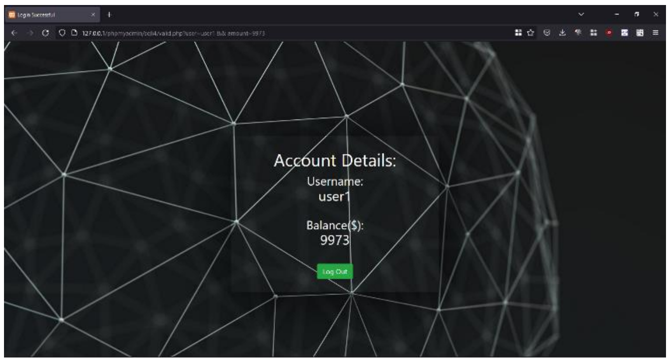
On lines 9, 10 and 11, we use Naïve Bayes Classifier by creating $classifer and for spam and legit keyword training set usage we create $spam and $ham respectively.

On lines 13 and 14, we create $categoryr and $categoryp and classify spam and legit keywords from input username and password respectively.

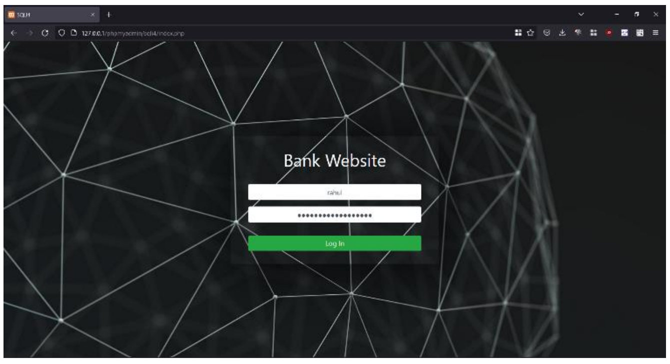
On line 16, as we can see, if any of the input username or password falls under the spam category, it will automatically redirect to fake website which is “fakevalid.php” here.

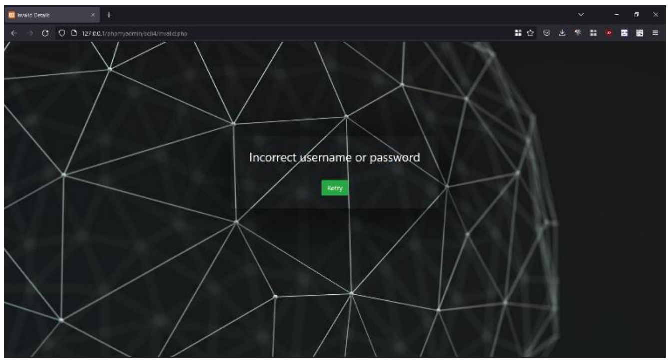
6. <u>Screenshots:</u>



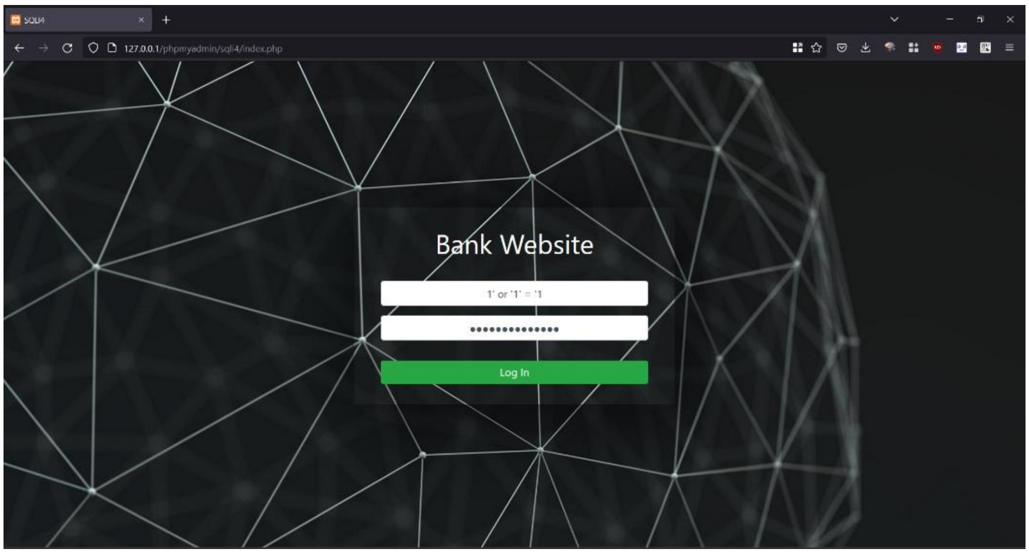

Account Details:
Username:
user1
Balance($):
9973

Entering the correct username and password as shown in the screenshots above, we get access to that particular user’s information and all the account details associated with it.




Incorrect username or password

Entering an invalid username and password as shown in the screenshots above, we are displayed a message “Incorrect username or password”. We are shown a button to retry the login procedure again.


SQLI4

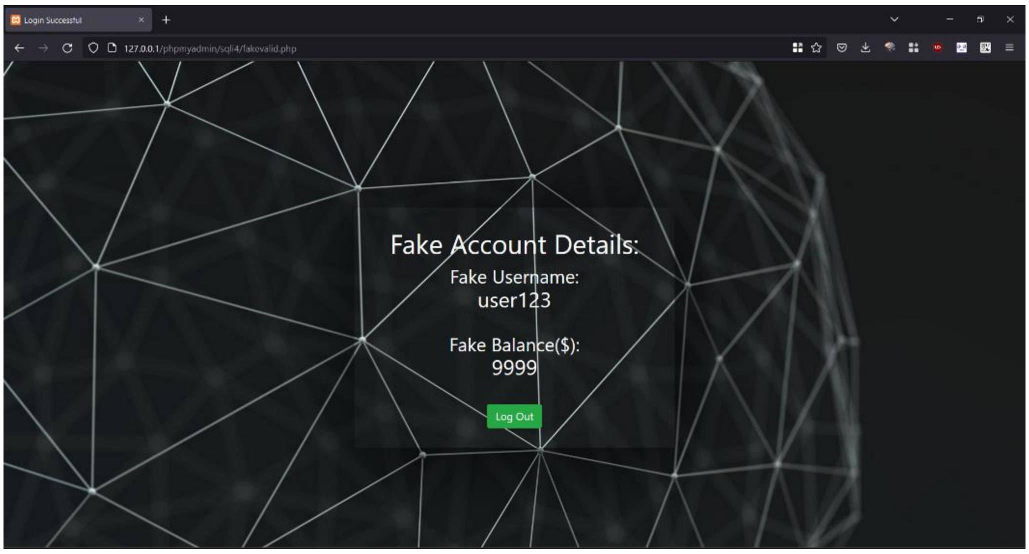
127.0.0.1/phpmyadmin/sqli4/index.php

Bank Website

1' or '1' = '1

••••••••••••••

Log In


Login Successful

127.0.0.1/phpmyadmin/sqli4/fakevalid.php

Fake Account Details:

Fake Username:
user123

Fake Balance($):
9999

Log Out

Here we can see that as soon as the attacker tries to attempt SQLI attack, it is detected by the classifier and is redirected to the fake website, with all the fake account information, so that no one is at a loss if he decides to leak the fake information online. The fake website accessed here is "fakevalid.php" which is accessed whenever the attacker attempts SQLI attack, thus deceiving the attacker.

7. **Bibliography:**

- https://www.forcepoint.com/cyber-edu/deception-technology
- https://www.techtarget.com/whatis/definition/deception-technology
- https://www.tutorialspoint.com/what-is-deception-technology-in-cybersecurity
- https://en.wikipedia.org/wiki/Deception_technology