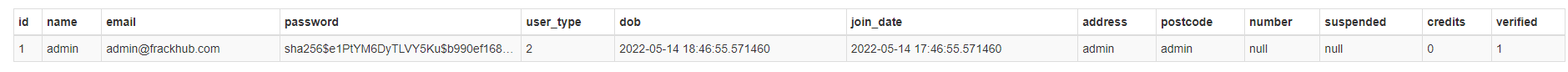
Audit Log:

Initialization – this process is sever intended which initialized the database but for now it is just proof of concept in which the frack hub launches. So, it is localized.

As server side in theory will not be accessible directly by consumer this step is done before the log evaluation.

Database initialization process is now complete now I will check to see if it is encrypted:

Opening the database file we can see that the database is not encrypted however the “password” in the users table is encrypted using sha256



This means in a data leak if the central server was compromised holding the confidential information records, unfortunately the identity of users will be compromised, login credentials will however be secure

Graphical user interface, text, application

Description automatically generated

The user area for the frackhub service is a webservice and as you can see it is a form document, I will demonstrate my attempt to breach this. First I will see what modules are loaded into the service.

Graphical user interface, application

Description automatically generated

Above screenshot shows the modules loaded into the splash screen when running the hosted service. As you can see the resources that are exposed to are the attack surface. This has been kept to a minimal level here. What I will do is attempt a login.

Initiating a login has fetched the identical resources as loading the page:

A screenshot of a computer

Description automatically generated

This means that the database calls are not exposed to the front end

I have tried to login to the email field however it has shown field validation is present

Graphical user interface, text, application

Description automatically generated

This is not a problem to bypass this validation, I have used:



What I have done is changed the type of field from “email” to “text” therefore it will accept any characters, as there are two fields where the Email and Password ids are unknown I will sql inject the term: ' or '1'='1

This means that the SQL table value is always true

After launching this attack, I can see input validation is present as submitting with the characters has shown Chart

Description automatically generated

In the splash screen, this proves that the email field is immune from such attacks

Next, I will try to login assuming of a known i.d: [admin@frackhub.com](mailto:admin@frackhub.com)

Password is not known so I will sql inject with a known email but unknown password

Graphical user interface, text, application

Description automatically generated

The login attempt had failed which shows success that input validation has been made in the backend preventing the table to be conditionally always true

I will be creating a new user account:

I will do this trying to see if age verification is present:

Graphical user interface, text, application

Description automatically generated

Expected that the creation would fail

Actual outcome:

Graphical user interface

Description automatically generated with medium confidence

The form has not allowed the creation of the account meaning the age verification – DOB system is working.

Now filling details over the minimum age of 18 account has been created.

Graphical user interface, application

Description automatically generated with medium confidence

After login I am greeted by this splash screen, I will test if I can add anything to the database and deal with products.

A picture containing icon

Description automatically generated

After trying to add a product into the database using the add item feature, it has given an error. This error is expected as user access control has been implemented, in which each user in the system is manually approved to use the system.

I will now access as the authorized administrator and I have made a verified user account role.

I will try to add the item with an invalid form value:

A picture containing graphical user interface

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Expected output:

To output error that one of the fields are incorrect

Actual output:

A picture containing text, indoor, appliance, screenshot

Description automatically generated

This item has been added to the database for the invalid listing price which shows input validation is not implemented into the price category



Trying to make offer is not possible on the unverified account thus the User Access levels are operational

So what happens next:

Input Validation is a must in all fields, this prevents any malicious listing to be performed.

Database encryption is a must because if a data breach was to happen personal information will not be disclosed.