Task-9: Report on Vuln-Bank

Prepared for: MuLearn Bootcamp

Prepared By: Atul H

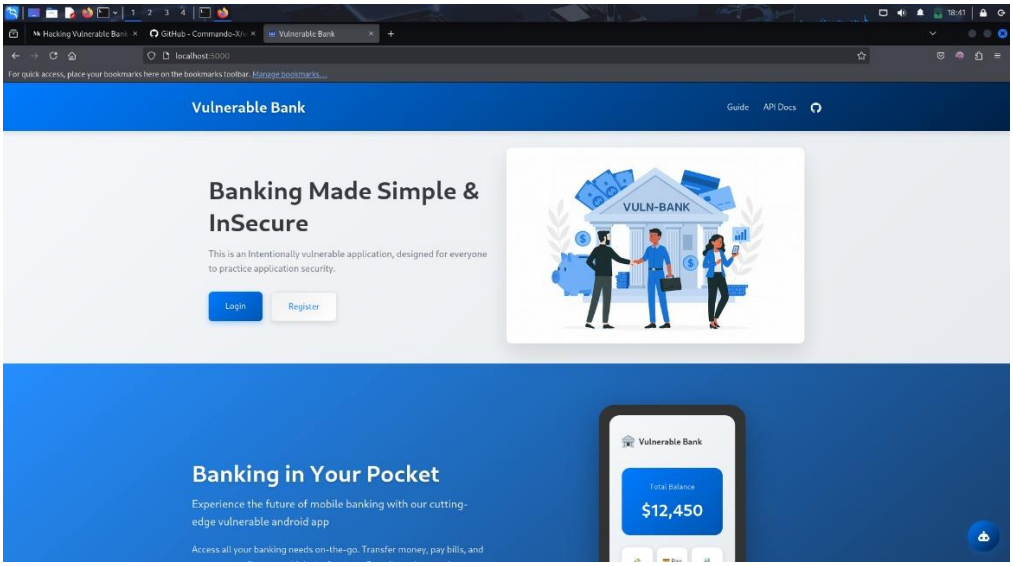
Description: Clone and deploy the purposely vulnerable Bank web app, inspect it for issues, and produce a concise professional vulnerability report.

Repo URL: https://github.com/Commando-X/vuln-bank.git

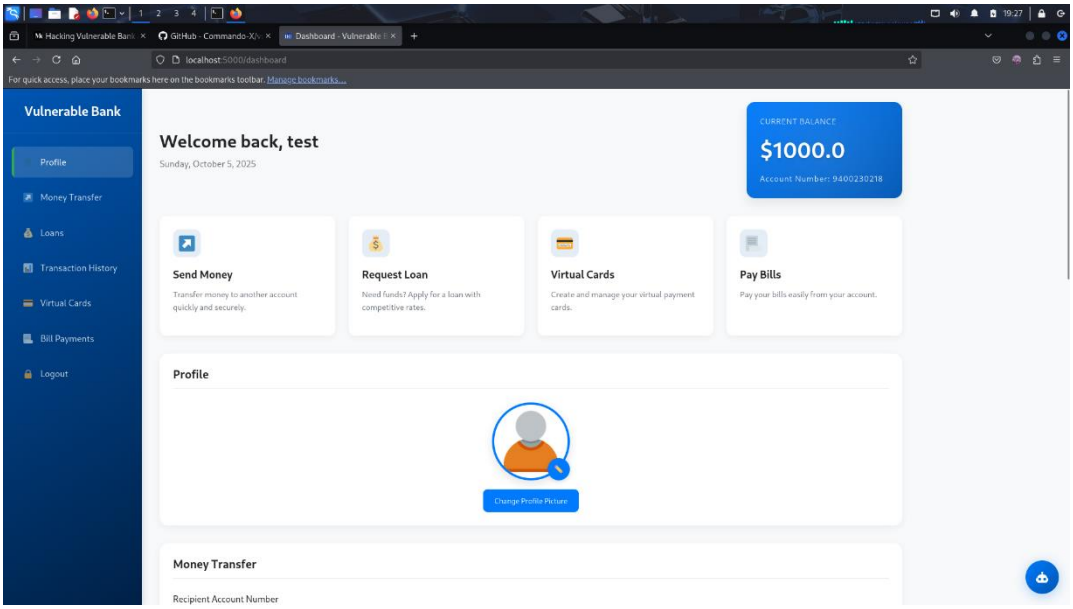
This report focuses on the vulnerabilities identified by using this vuln-bank repo from github. This has a server which runs using the docker service in our localhost.

```
File  Actions  Edit  View  Help

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/


┌──(ferronic㉿ferronic)-[~/vuln-bank]
└─$ sudo usermod -aG docker $USER

┌──(ferronic㉿ferronic)-[~/vuln-bank]
└─$ newgrp docker
┌──(ferronic㉿ferronic)-[~/vuln-bank]
└─$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES

┌──(ferronic㉿ferronic)-[~/vuln-bank]
└─$ sudo docker-compose up --build
WARN[0000] /home/ferronic/vuln-bank/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, pl
ease remove it to avoid potential confusion
[+] Running 0/1
 ⠼ db Pulling                                                                                              2.1s
```

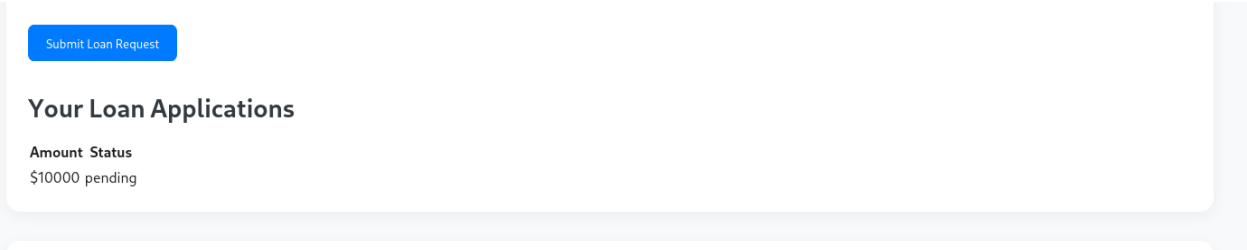
By installing our vuln-bank into the docker by using “git clone https://github.com/Commando-X/vuln-bank.git && cd vuln-bank && docker-compose up –build -d” command, we can access the local site after successfully setting up our machine.



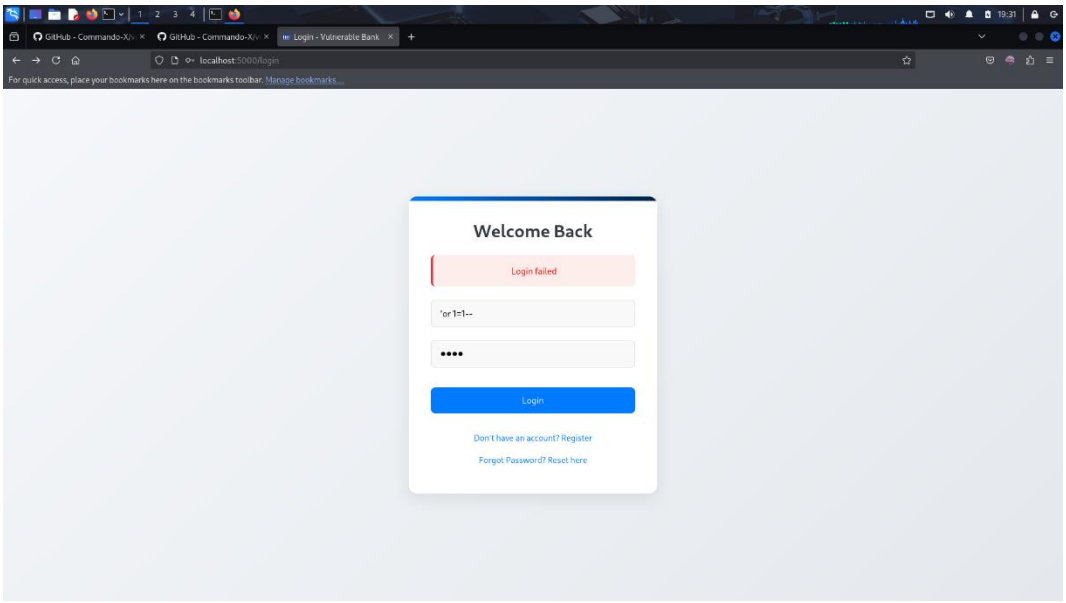
This is the sample interface which pops up at our local host. After setting up we can try to create a new account as the login with admin as username and admin as password doesn't work.



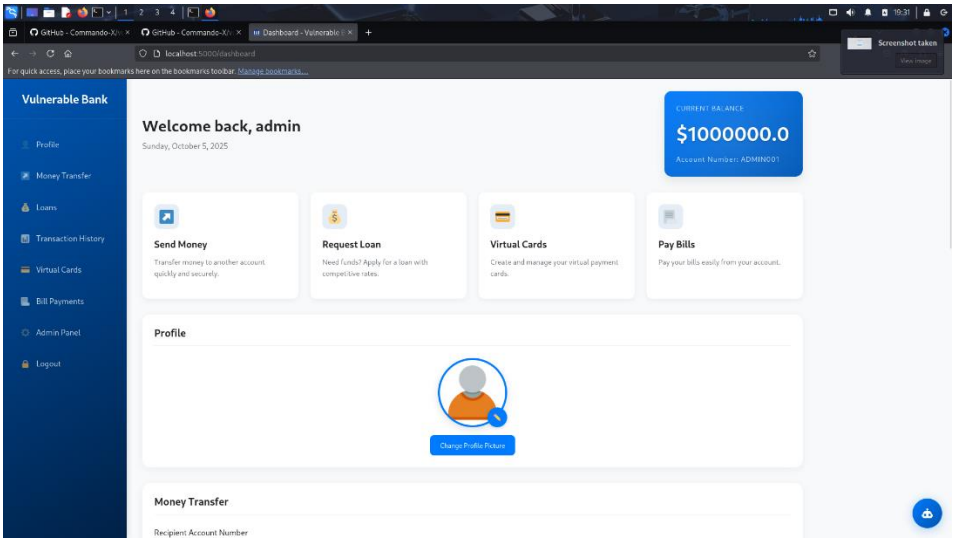
In this we have created(registered) a new account with the username and password as "test". The normal balance which we will have immediately after logging in is $1000. This is a normal user interface. As this is a vulnerable page and following safe practices we can now try to escalate our privilege by logging in as admin. Before we do that lets request loan from this test account.



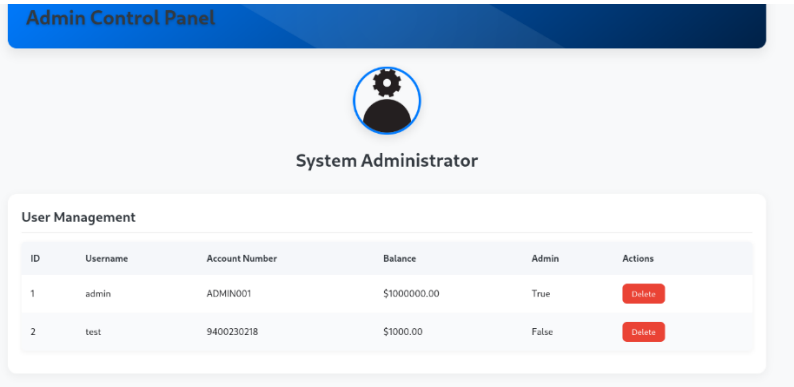
The request status shows pending. For this to get approved the admin should accept this loan application. Now lets try to access the admin page and grant this loan request.



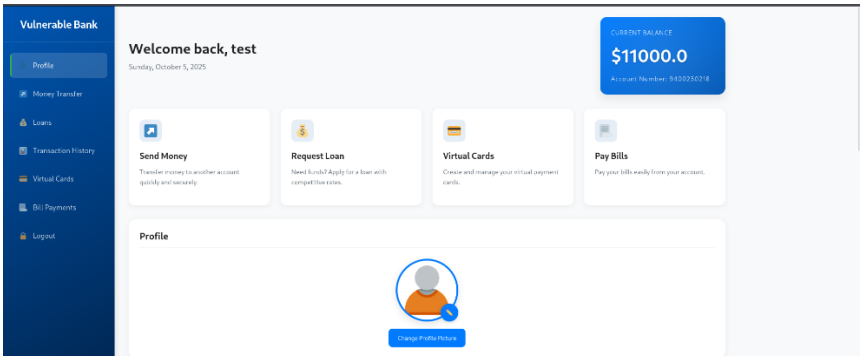
This is the classic SQL injection method to bypass all the false values and making it result in true values to bypass. “‘or 1=1—", password:{anything}(I used 1234). We then successfully bypasses the login page via SQL injection.



We can now see our admin’s dashboard, he also have admin panel to control the requests and transactions. Lets take a look!

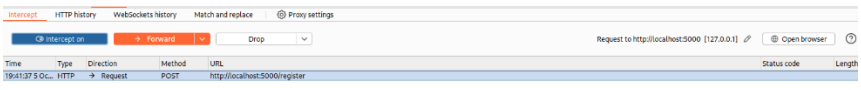


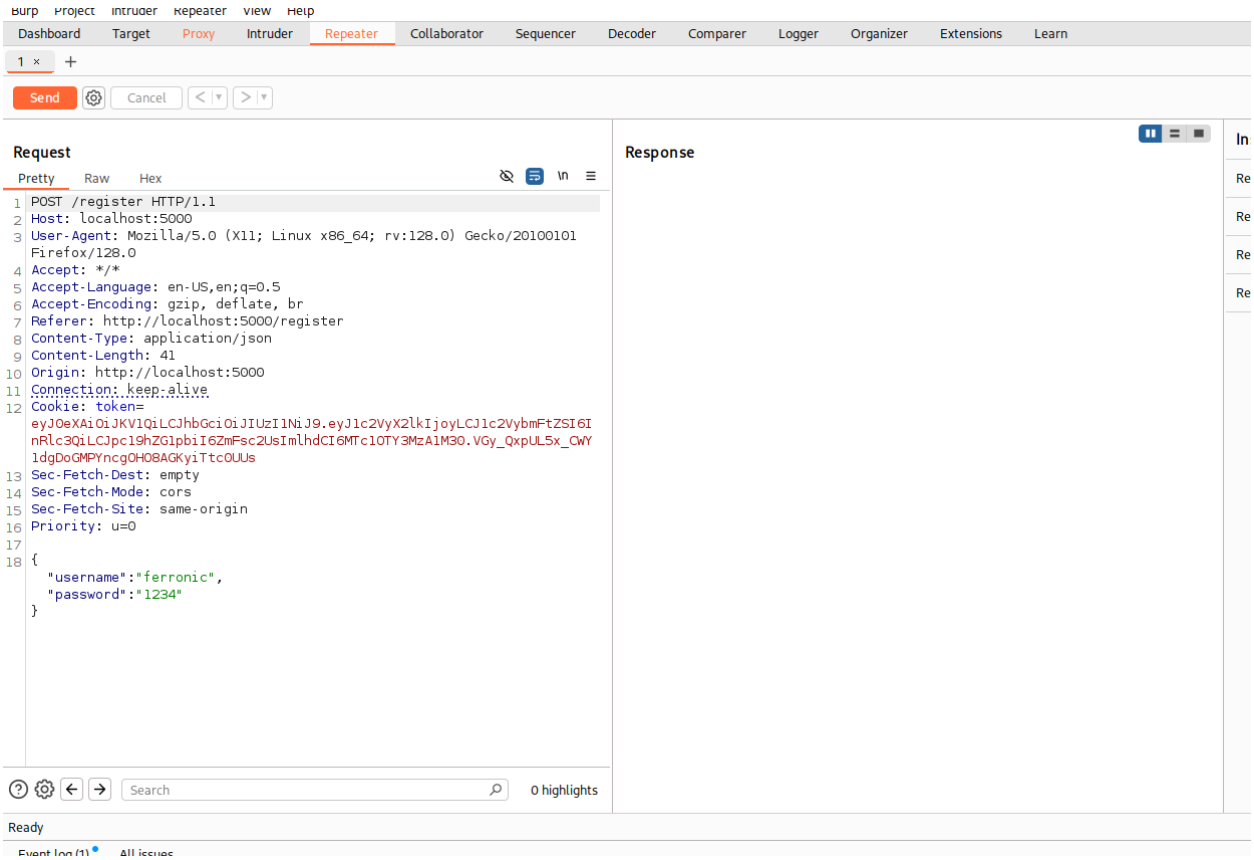
Here we can see the number of users logged in or using our bank services. We can also see the request made by test for the loan. Once the loan is approved we can check the test account and see whether it resulted.



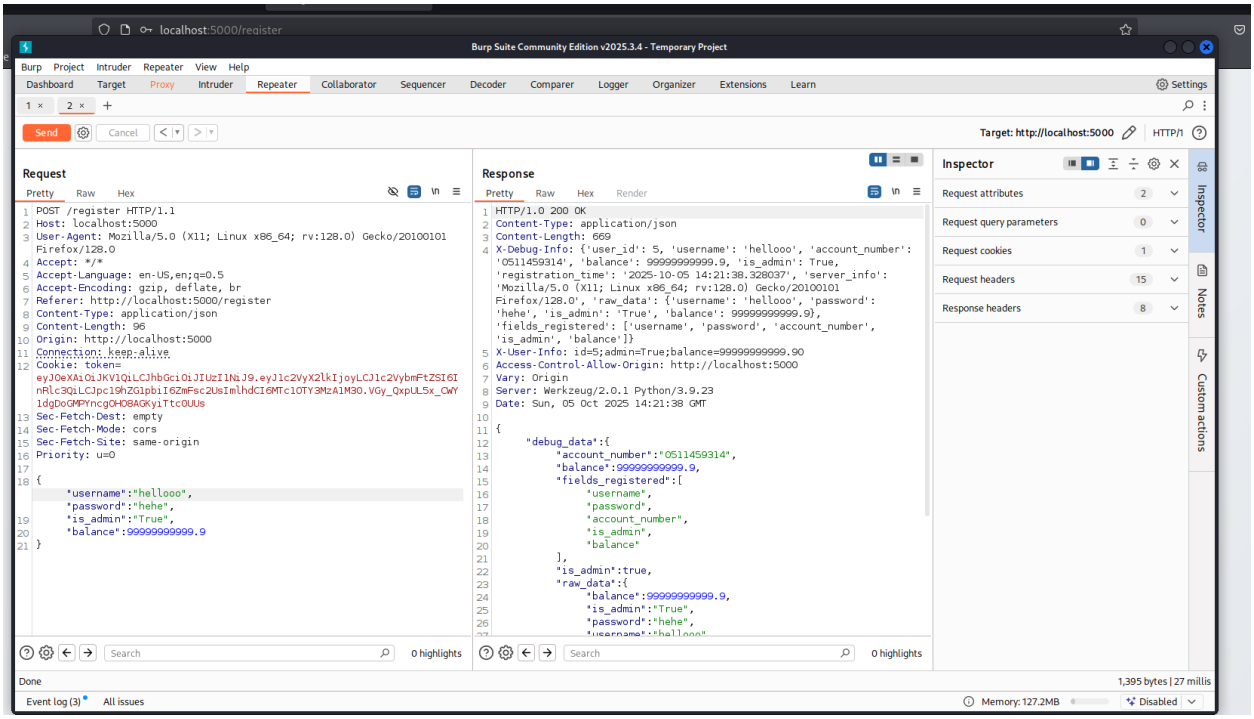
We initially had $1000 and after the loan request been approved by the admin we now have $1000 + $10000 resulting in a total of $11000. We can now try to create a new account with a lot of bank balance without the admin's approval. How? By being an admin!.

Lets try creating a new account, but not in a standard way but using the foxy-proxy to monitor through burp. When we enter the credentials of a new user, it got registered, we can try to change that by sending the proxy to the repeater.

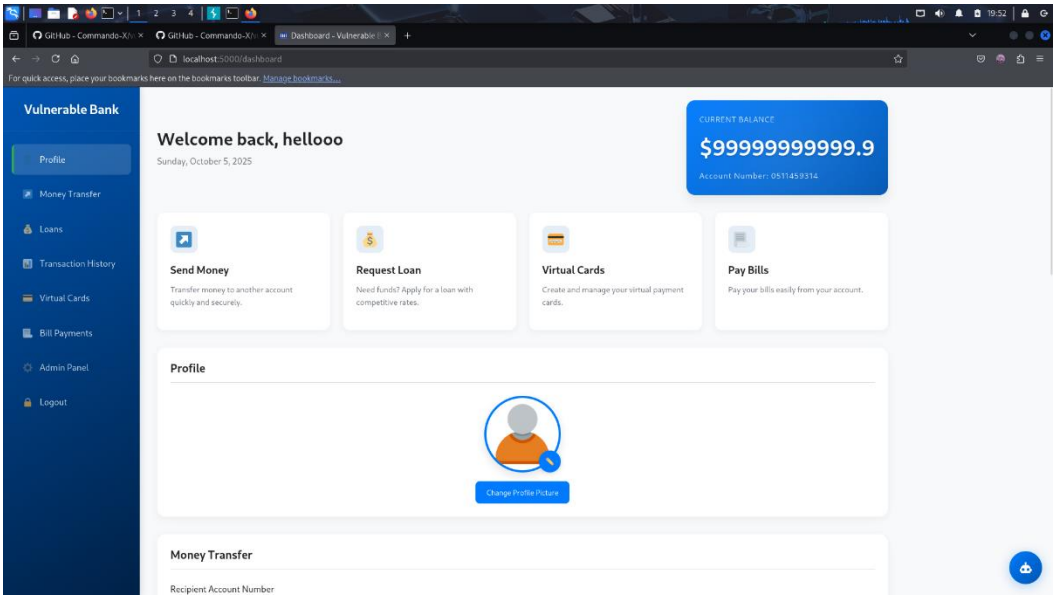




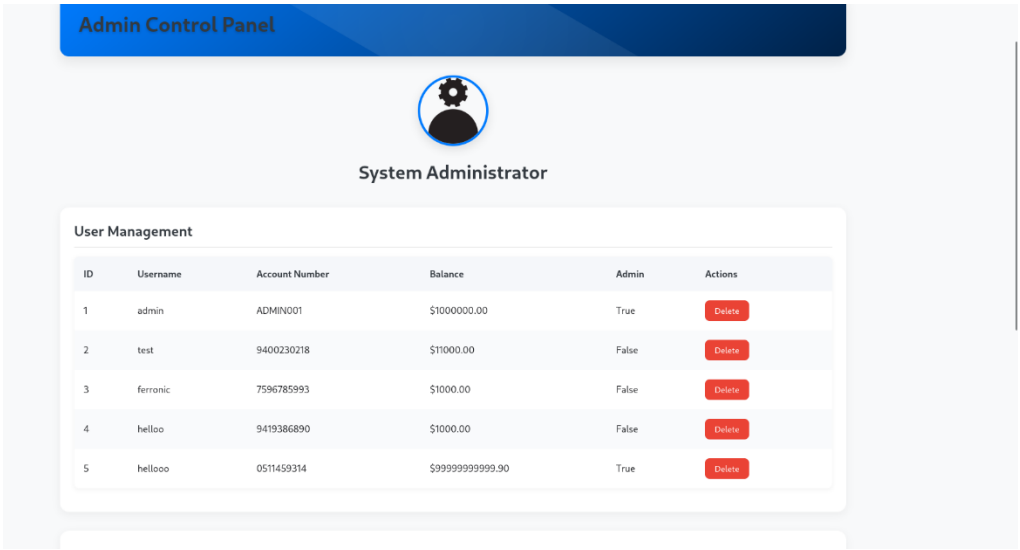
When we press send we can see a 200 ok message, showing that we are not admin, we can set the “Is_admin” to true and the change the “balance” to as much amount as we need, and after changing the username press send.



There we go! We got our new account created with that much amount as our balance. Lets try logging in.



we got an infinite amount of money here, also we have the admin privilages, on the left side we can see our admin panel.



when we monitor, we can see all the users and admins. We can also remove anyone from this category, even we can replace the actual admin.

The Vulnerable Bank app exhibits a range of security flaws that could allow the hackers to gain unwarranted access, commit financial fraud, and even take advantage of the AI chatbot. It is crucial that every security flaw be fixed as it can avert the loss of user funds, safeguard user privacy, and maintain the trust of the organization. It is highly recommended to fix these problems in order to have a strong security stance in the future.