Authentication and Replay

Code can be found here: https://github.com/c2003-tamu/413
In auth directory

Environment setup

Navigate to auth/ directory
Run command: python3 -m venv venv
Run command: source venv/bin/activate
Run command: pip install -r requirements.txt
Run command: python bank.py
Create another terminal tab to run .sh files that interact with bank

Exploitation

Using the endpoint /unsafetransaction, attackers can make transactions between two bank users without having to authenticate who they are and could skim requests and simply replay them, as is seen in the ./unsafe.sh script.

This endpoint simply takes input from the user as to which user to transfer money from and to and amount, without validating that the transaction is unique or that if the user is authenticated to make such a request.

Fixing the Vulnerabilities

To fix these vulnerabilities, we will implement the following endpoints:

/authenticate
Takes in a username, password, and a nonce. Returns a cookie for the user to use when making safe requests. The cookie has a finite time to live (5 minutes). This endpoint is not vulnerable to replay attacks because it takes in a nonce and adds said nonce to a set where the bank checks against previously used nonces. This endpoint ensures authentication through the use of a password.

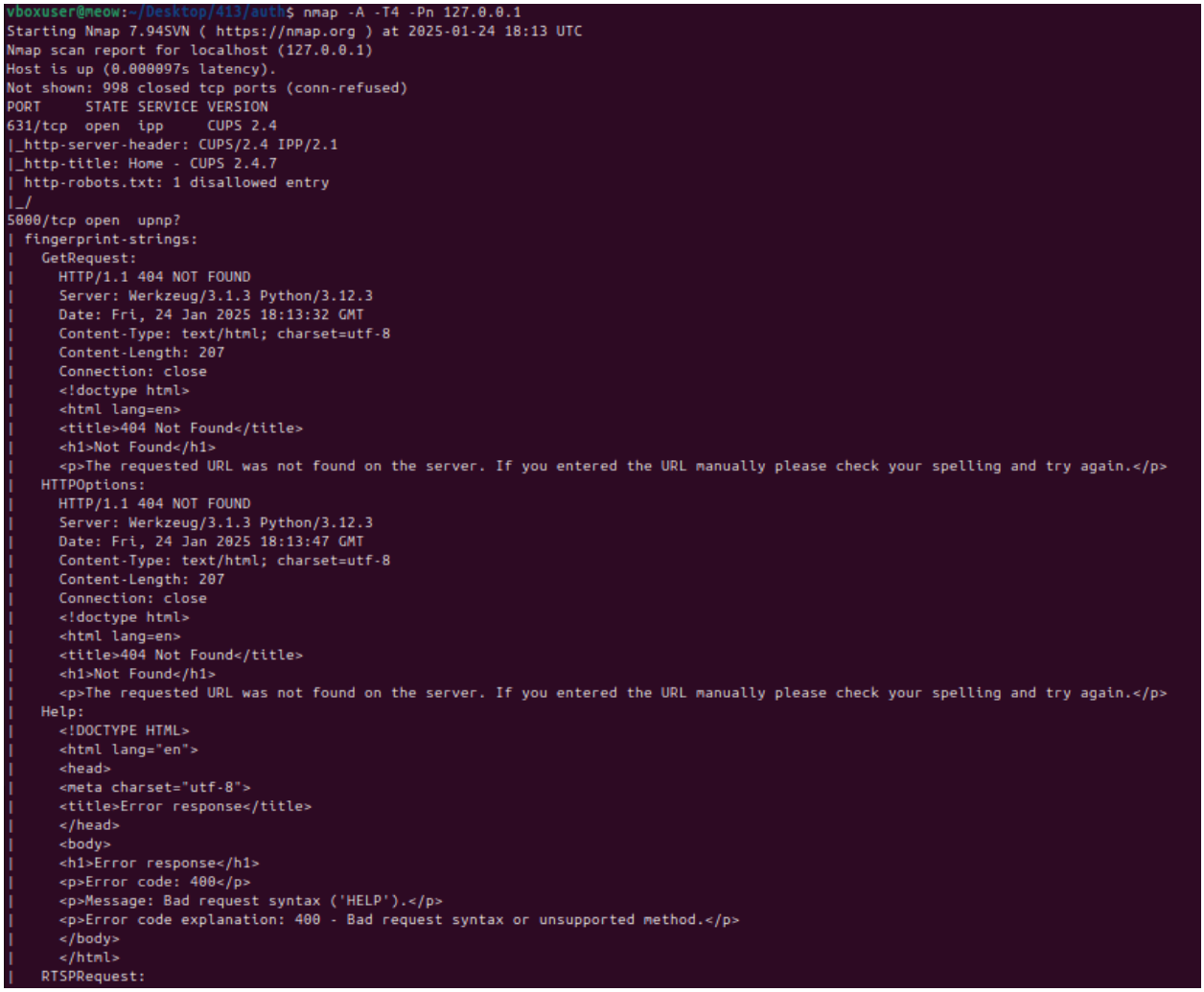
NOTE: It should be noted that if this method were implemented in reality, rather than in a POC, the /authenticate endpoint would likely not receive a plaintext password from the users, there would likely be some type of encoding that happened before sending the password over a network (or we could just use google OAUTH).

/safetransaction

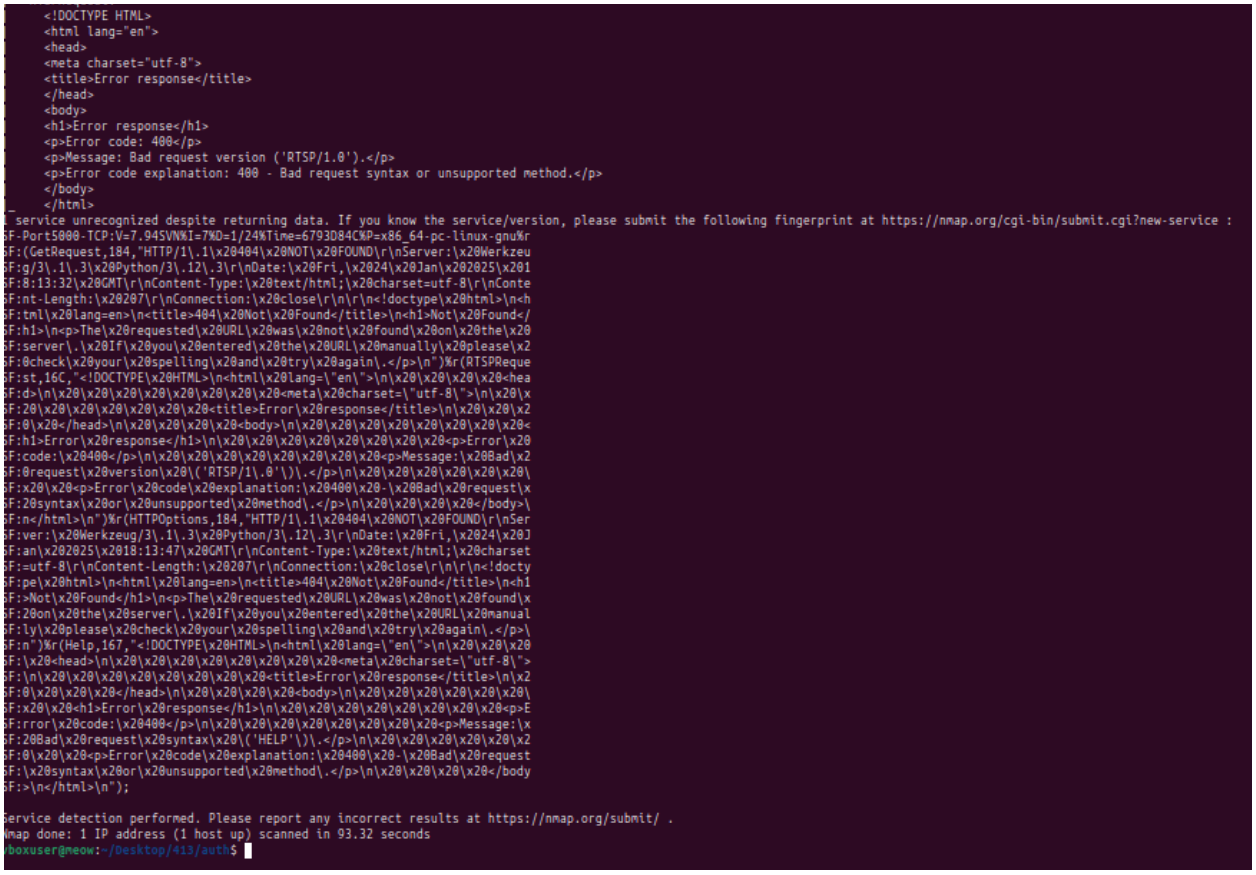
Takes in from id, to id, amount, cookie, nonce, timestamp. If all information is valid, the amount is transferred from one user to another. In order for the information to be valid, the cookie must be correct and not expired, the nonce must be unique to that transaction, and the timestamp must not be more than 5 minutes before the current time.This endpoint is not vulnerable to replay attacks because each transaction requires a nonce that can only be used once. This endpoint ensures authentication because it relies on the user to have authenticated and to have received a valid cookie to use along with requests.

The use of these safe endpoints can be seen in ./safe.sh

NMAP



```
vboxuser@meow:~/Desktop/413/auth$ nmap -A -T4 -Pn 127.0.0.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-24 18:13 UTC
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000097s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT     STATE SERVICE VERSION
631/tcp  open  ipp     CUPS 2.4
|_http-server-header: CUPS/2.4 IPP/2.1
|_http-title: Home - CUPS 2.4.7
| http-robots.txt: 1 disallowed entry
|_/
5000/tcp open  upnp?
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.1 404 NOT FOUND
|     Server: Werkzeug/3.1.3 Python/3.12.3
|     Date: Fri, 24 Jan 2025 18:13:32 GMT
|     Content-Type: text/html; charset=utf-8
|     Content-Length: 207
|     Connection: close
|     <!doctype html>
|     <html lang=en>
|     <title>404 Not Found</title>
|     <h1>Not Found</h1>
|     <p>The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.</p>
|   HTTPOptions:
|     HTTP/1.1 404 NOT FOUND
|     Server: Werkzeug/3.1.3 Python/3.12.3
|     Date: Fri, 24 Jan 2025 18:13:47 GMT
|     Content-Type: text/html; charset=utf-8
|     Content-Length: 207
|     Connection: close
|     <!doctype html>
|     <html lang=en>
|     <title>404 Not Found</title>
|     <h1>Not Found</h1>
|     <p>The requested URL was not found on the server. If you entered the URL manually please check your spelling and try again.</p>
|   Help:
|     <!DOCTYPE HTML>
|     <html lang="en">
|     <head>
|     <meta charset="utf-8">
|     <title>Error response</title>
|     </head>
|     <body>
|     <h1>Error response</h1>
|     <p>Error code: 400</p>
|     <p>Message: Bad request syntax ('HELP').</p>
|     <p>Error code explanation: 400 - Bad request syntax or unsupported method.</p>
|     </body>
|     </html>
|   RTSPRequest:
```



From these screenshots, we can see what version of python is being used for the bank application, the OS and version of the host machine, all open ports, and the fact that some of the endpoints that are checked by nmap are not set up. This could provide attackers with a more specific way they are able to exploit the host system for this banking application.