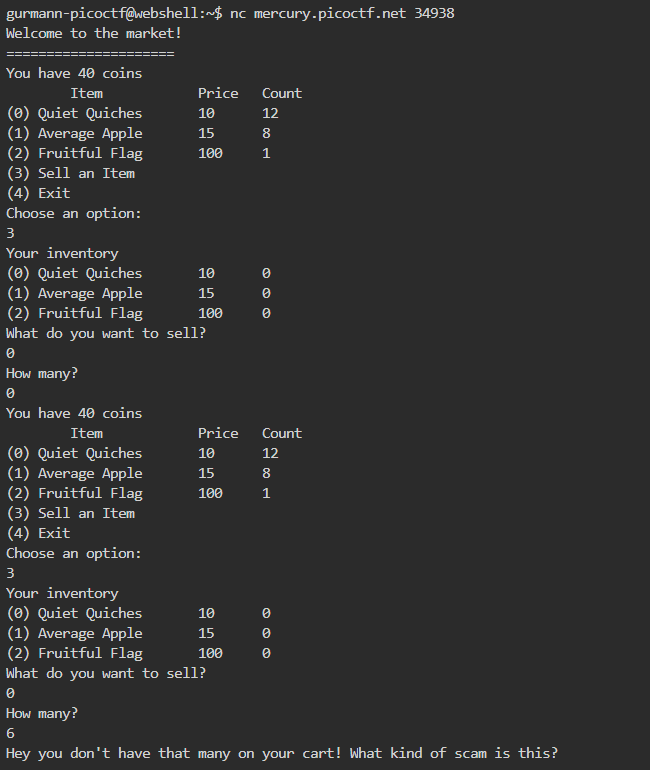
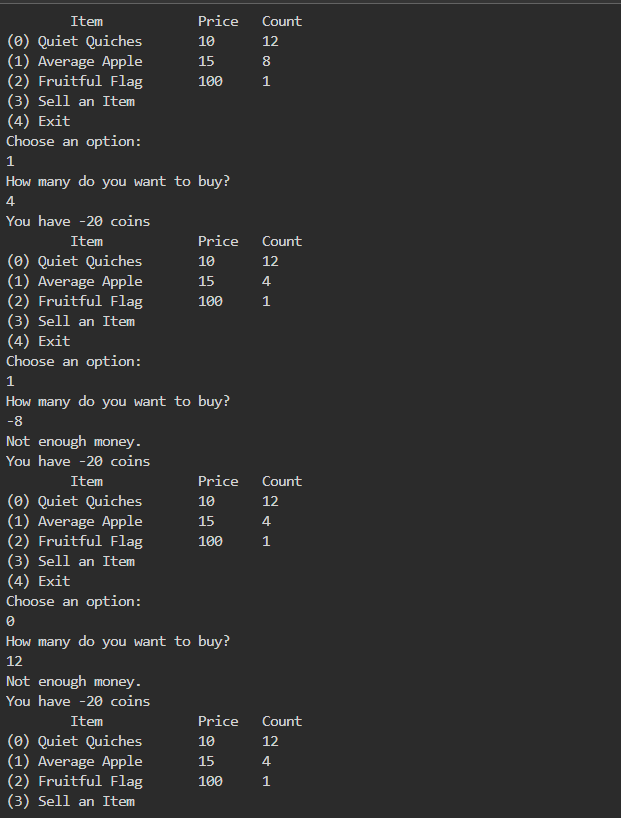
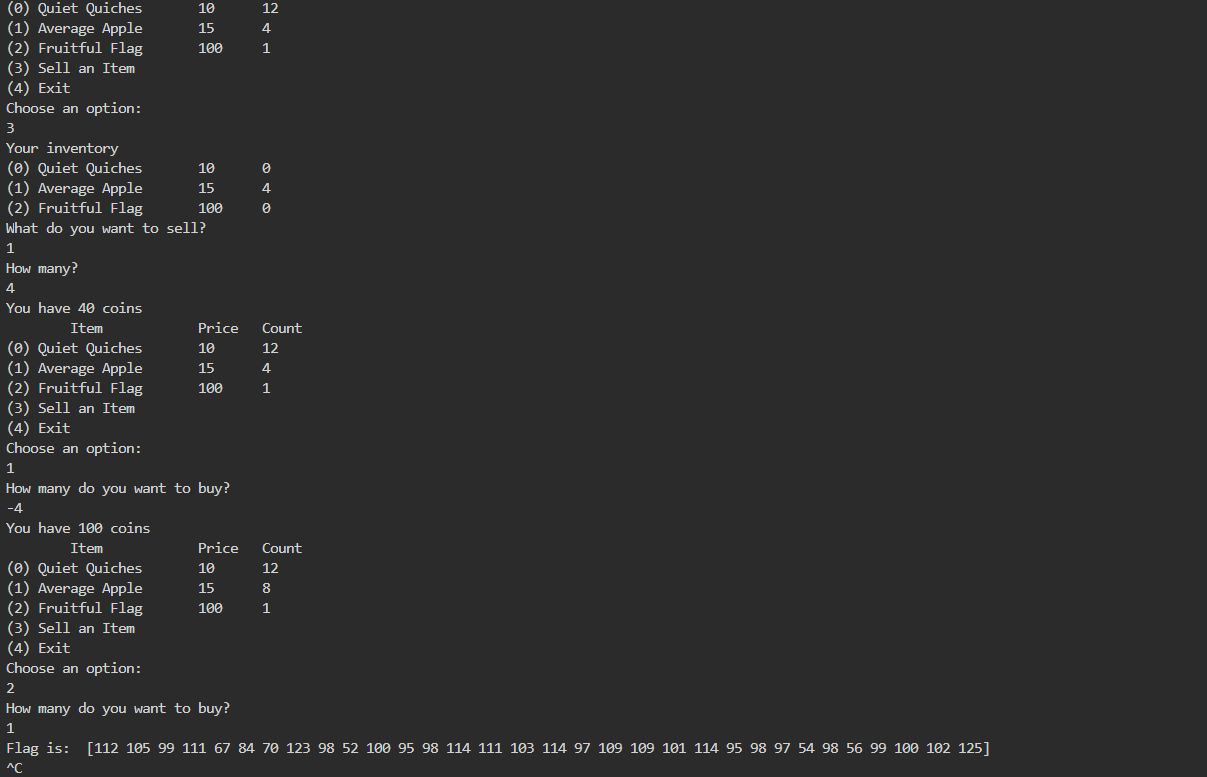
1.shop





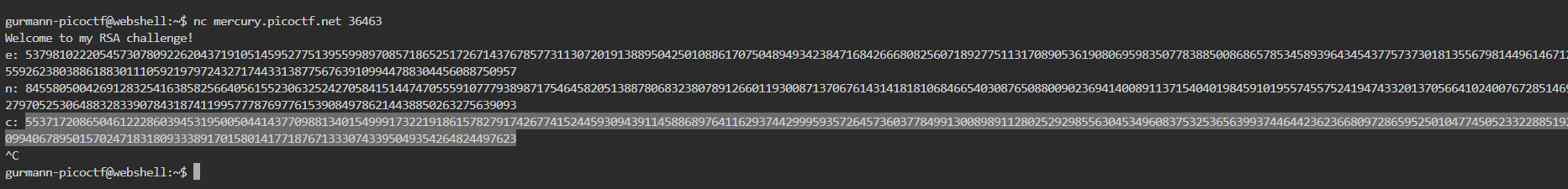


Use webshell to access the market and then figure out how to earn 100 coins to buy flag.

Then we convert this decimal into ascii characters to get the required flag.(Had to refer to convert decimal to ascii).

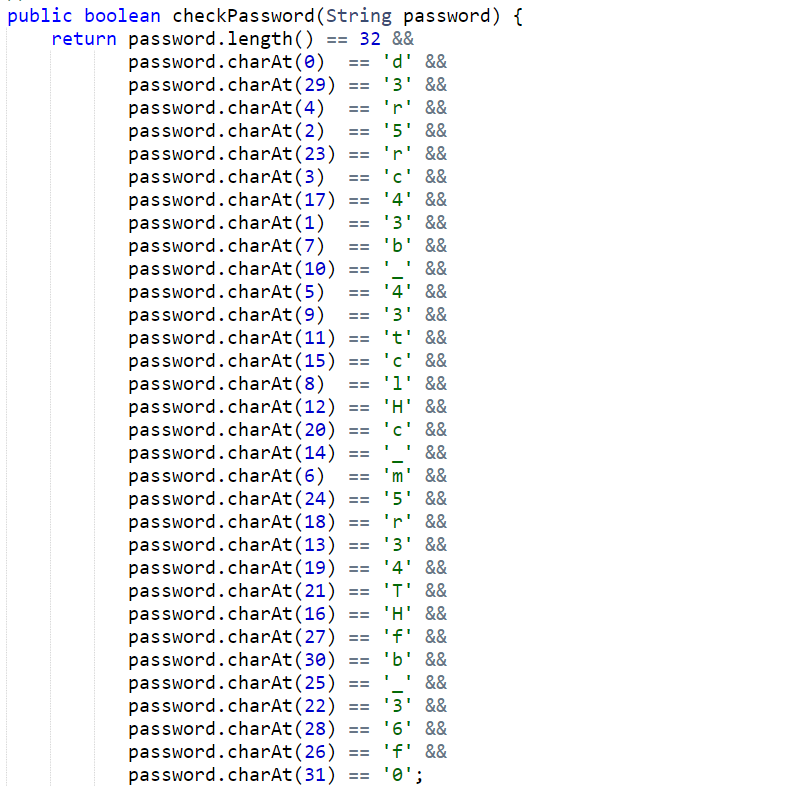
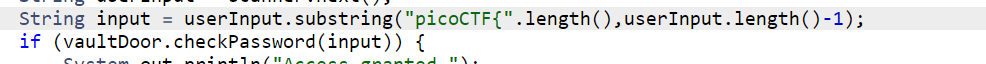
2.

Dachshund attacks



We login to find the value of e, n and c. We then run it through Weiner’s attack (Had to refer online sources) to the pico ctf key.

3.Vault Door 1

In the source code, it says picoCTF as the format. So we just go from 0 to 31 character wise to find the rest.

The charAt() returns/replaces the character at the specified index in a string

4. Vault door 3 :

Graphical user interface, text, application

Description automatically generated

We know length of password is 32 and after running it through loops we get a given statement

After first for loop we get the first 8 characters as “jU5t\_a\_s”

After the 2nd for loop we get the next 8 chars as “1mpl3\_an”

After 3rd loop we get the next 16 characters as “4gr4m\_4\_u\_1fb380”

4th loop does not make any change to the flag.

So the flag becomes picoCTF{jU5t\_a\_s1mpl3\_an4gr4m\_4\_u\_1fb380}

5. Vault door 4:

Calendar

Description automatically generated

We convert this into ascii characters to get the flag. We get picoCTF{jU5t\_4\_bUnCh\_0f\_bYt3s\_c194f7458e}

5. Vault Door 5:

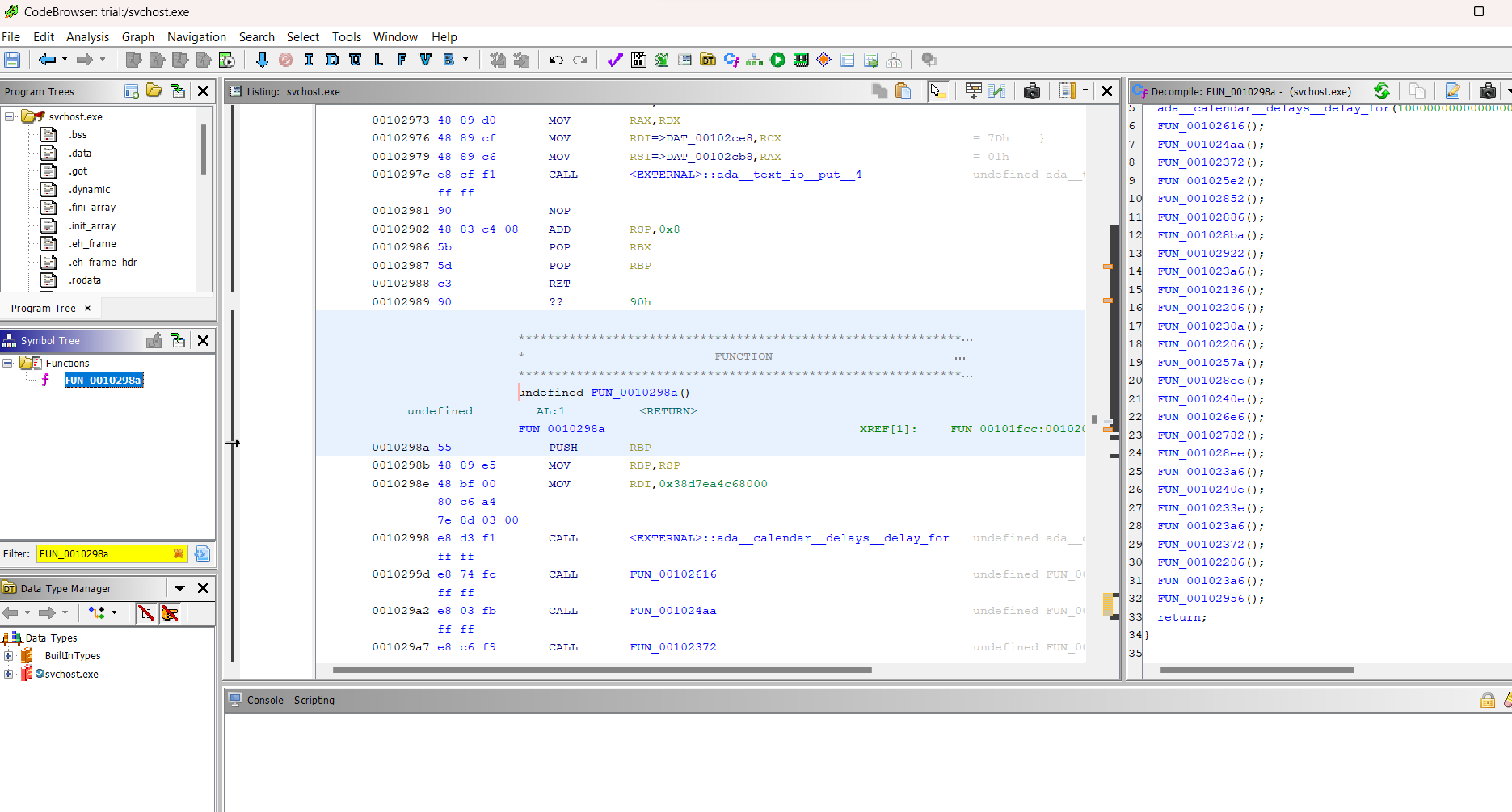
A screenshot of a computer

Description automatically generated with medium confidence

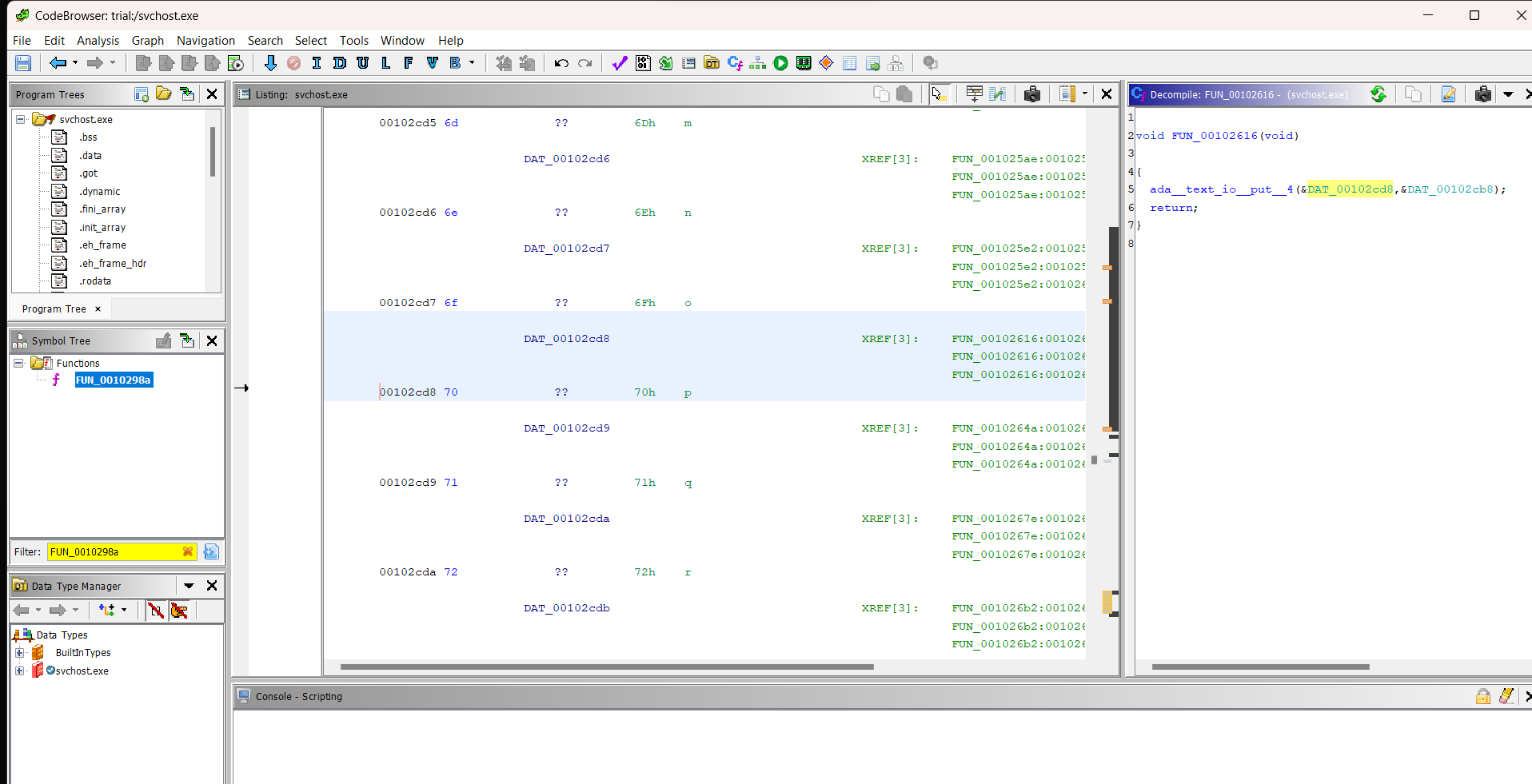
Graphical user interface, text, email

Description automatically generated

We can see that the password is converted to base64 and url decoding is also used. So we used an online decoder to get the output.(Had to refer to the internet).

6. Hurry Up wait

Decompiled the file using ghidra and found the character relevant for each function one-by-one(Had to refer to tutorial).



FUN\_00102616();-P

FUN\_001024aa();-i

FUN\_00102372();-c

FUN\_001025e2();-o

FUN\_00102852();C

FUN\_00102886();T

FUN\_001028ba();F

FUN\_00102922();{

FUN\_001023a6();-d

FUN\_00102136();1

FUN\_00102206();5

FUN\_0010230a();a

FUN\_00102206();5

FUN\_0010257a();m

FUN\_001028ee();\_

FUN\_0010240e();f

FUN\_001026e6();t

FUN\_00102782();w

FUN\_001028ee();\_

FUN\_001023a6();e

FUN\_0010240e();a

FUN\_0010233e();b

FUN\_001023a6();7

FUN\_00102372();8

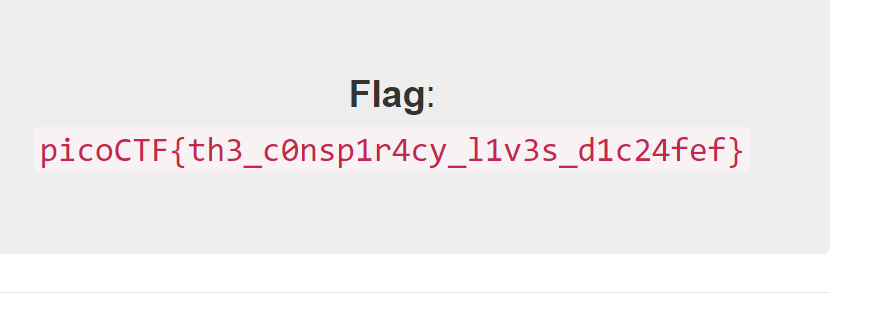
FUN\_00102206();e

FUN\_001023a6();4

FUN\_00102956();}

=picoCTF{d15a5m\_ftw\_eab78e4}

**Logon**

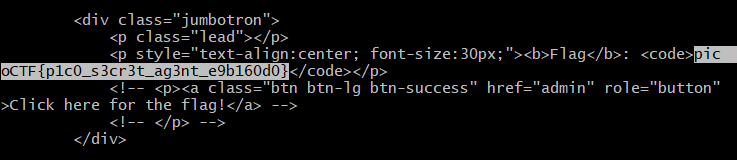


Picobrowser

User agent- identifies platform used by server to access website

We use curl to modify user agent header to “picobrowser” and it works,



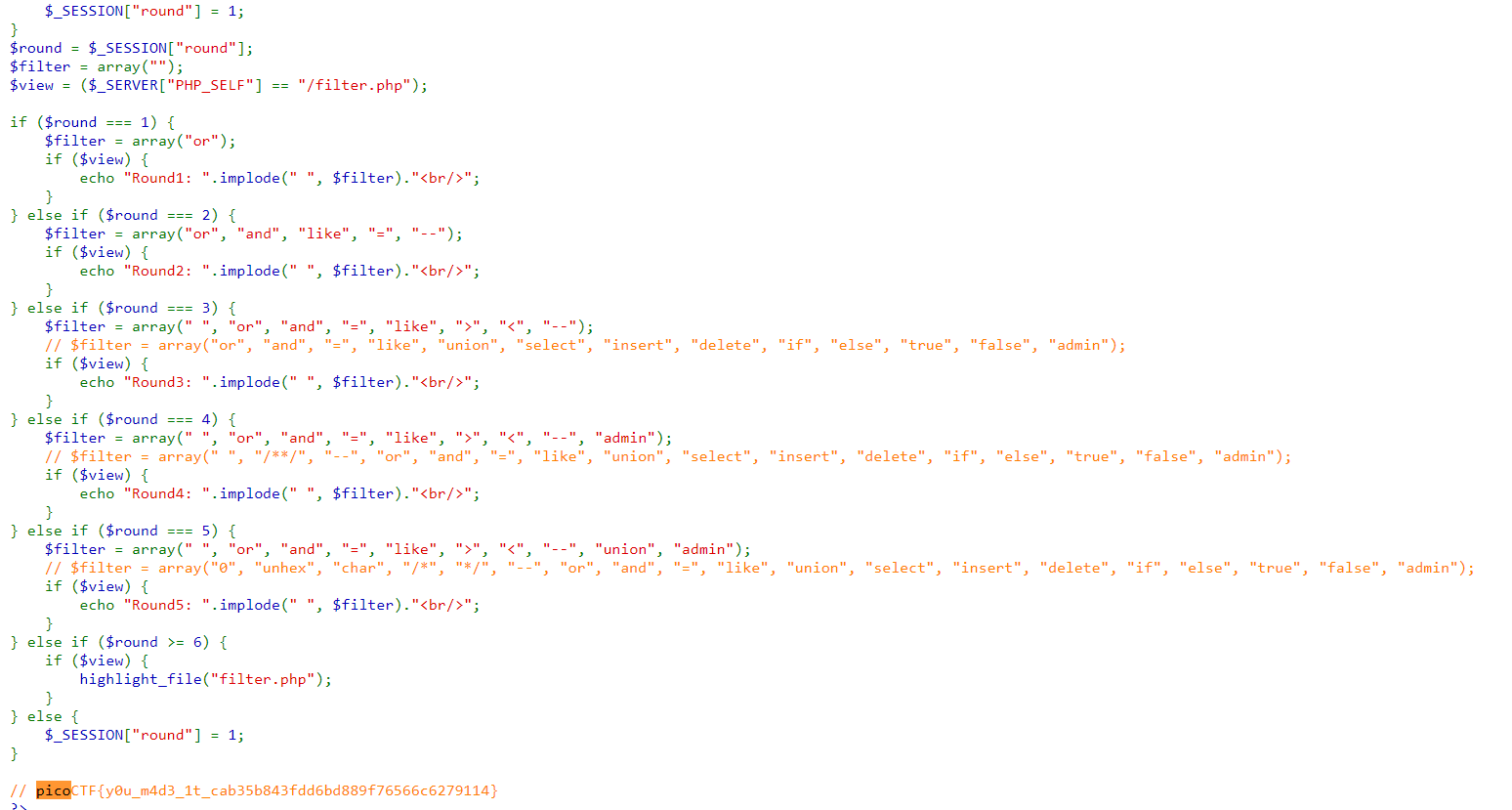


Web Gauntlet

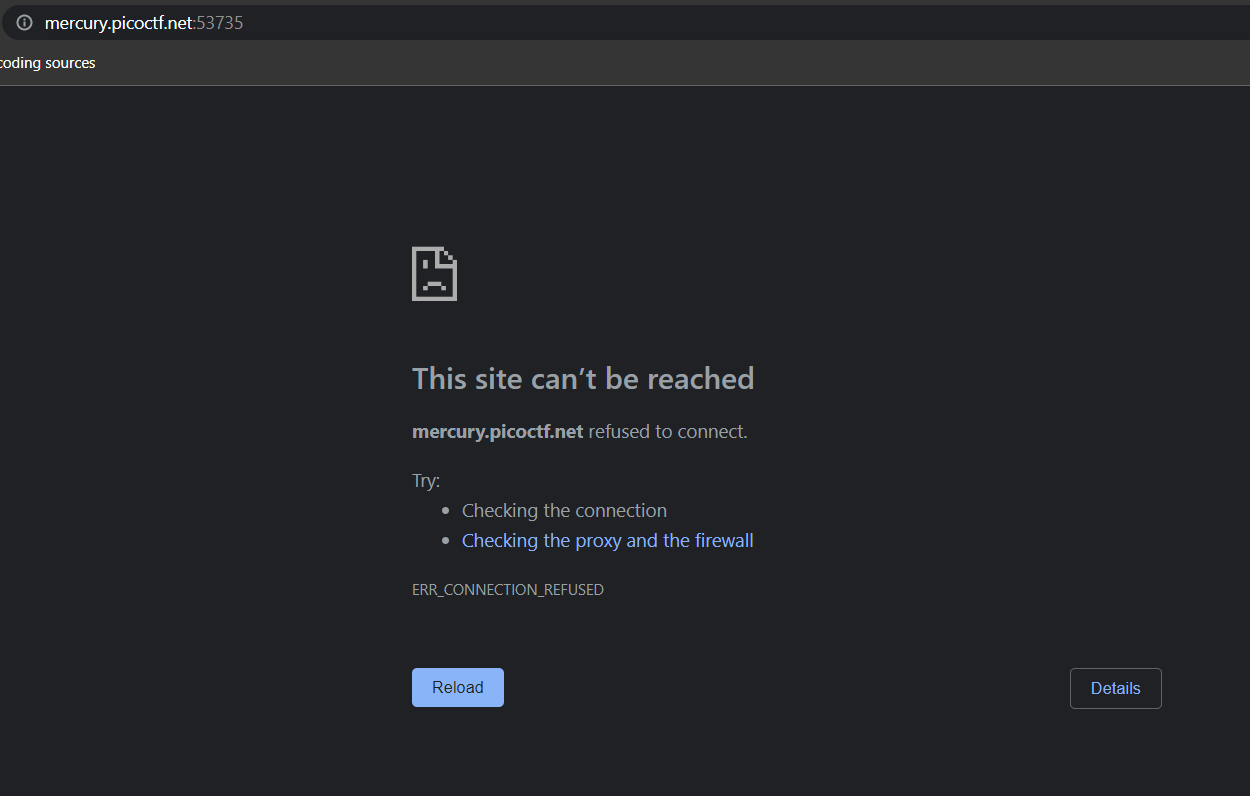
Learned about SQL injection, (<https://www.youtube.com/watch?v=ZQj5tSwaG0k>).

Spacing error- can be removed by replacing spaces with /\*\*/

||-used for concatenation



X MARKS THE SPOT



**XSS:**

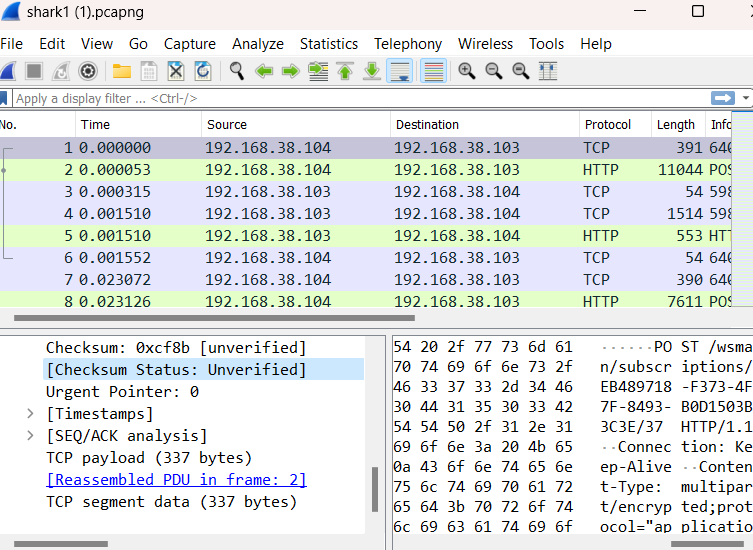
Cross Site Scripting: Form of injecting script into a webpage. Most browsers have an auditor to prevent this.

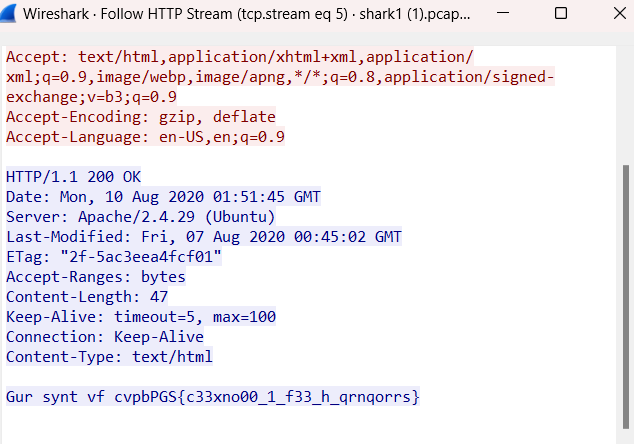
Assembly Language

1. MOV- MOV [destination] [source]- used to move data from source to destination
2. ADD- ADD [destination] [source] used to add two values
3. SUB -used to subtract
4. CMP-used to compare two values. If they are equal, Zero Flag=1 else 0
5. TEST- used to see if a register is 0 or not. If 0 then ZF=1.
6. JMP-used to jump to a particular code (JMP[DESTINATION])
7. JE- Jump If- used to jump to a line of code if ZF=0
8. JNE- Used to jump if ZF!=0
9. RET- Return
10. Jg-Jump if greater

Wireshark Do Doo…(had to follow tutorial)

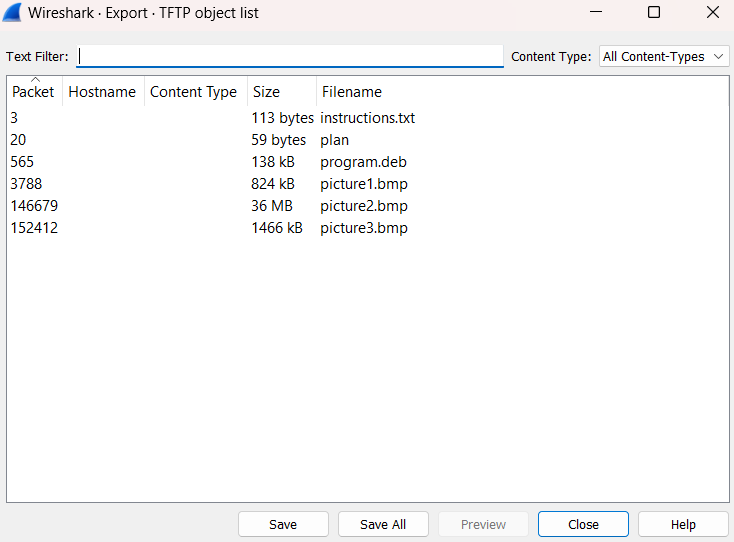
We open the file using wireshark.





Then we search for the packet we need to get the ctf

Trivial Flag Transfer Protocol:



We export all objects from the TFTP.

A picture containing table

Description automatically generated

From this we find the password as DUEDILIGENCE from plan.txt.

We use steghide to manually attack each of the files to find the flag.