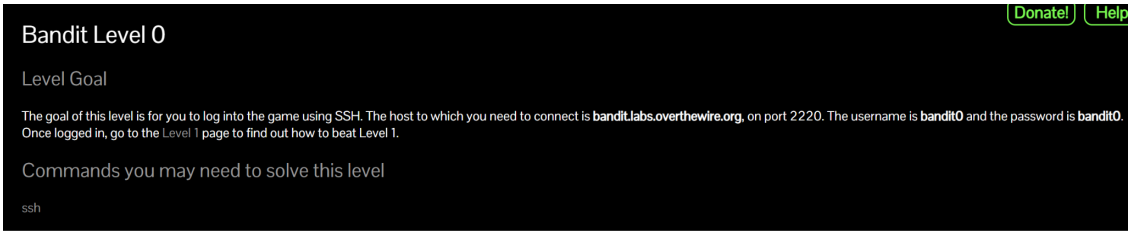
OemOem Challenge CyberSecurity

Bayu Putra Ibana

24/536830/PA/22776

1. OverTheWire Bandit

Level 0



Donate! Help

Bandit Level 0

Level Goal

The goal of this level is for you to log into the game using SSH. The host to which you need to connect is **bandit.labs.overthewire.org**, on port 2220. The username is **bandit0** and the password is **bandit0**. Once logged in, go to the Level 1 page to find out how to beat Level 1.

Commands you may need to solve this level

ssh

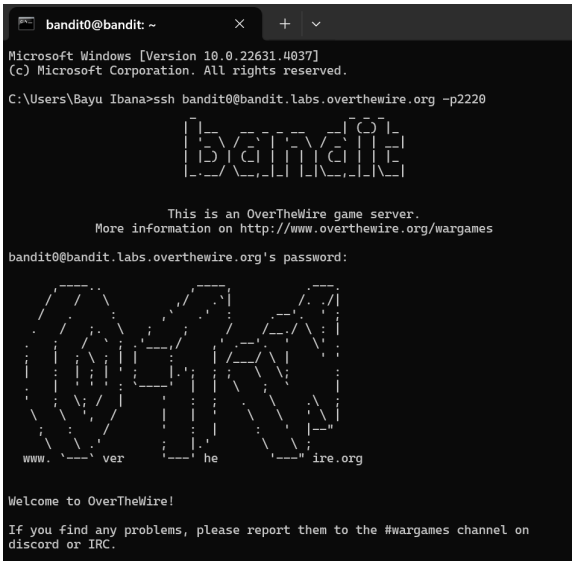
Step 1: Open Windows Powershell, by using Windows Key+R and typing cmd

```
C:\Windows\system32\cmd.e:
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Bayu Ibana>
```

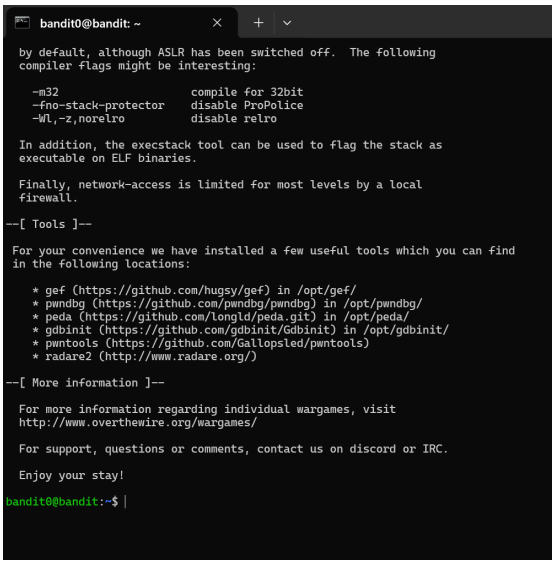
Step 2: Connect to bandit.labs.overthewire.org on port 2220 then enter username bandit0

```
ssh bandit0@bandit.labs.overthewire.org -p2220
```

with the password (bandit0)



```
bandit0@bandit: ~
Microsoft Windows [Version 10.0.22631.4037]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Bayu Ibana>ssh bandit0@bandit.labs.overthewire.org -p2220
                      _                     _ _ _
                     | |__  __ _ _ __   __| (_) |_
                     | '_ \ / _` | '_ \ / _` | | __|
                     | |_) | (_| | | | | (_| | | |_
                     |_.__/ \__,_|_| |_|\__,_|_|\__|


                    This is an OverTheWire game server.
            More information on http://www.overthewire.org/wargames

bandit0@bandit.labs.overthewire.org's password:

      ,----..            ,----,         .---.
     /   /   \         ,/   .`|        /. ./|
    /   .     :      ,`   .'  :     .--'.  ' ;
   .   /   ;.  \   ;    ;     /    /__./ \ : |
   .   ;   /  ` ; : .---.;    ; .--'.  '   \' .
   ;   |  ; \ ; | |   :     : | /___/ \ |    ' '
   |   :  | ; | ' ;   |.'  ; ;   \  \;      :
   .   |  ' ' ' : `----'  |  |  \   ;  `      |
   '   ;  \; /  |     '   :  ;   .   \    .\  ;
    \   \  ',  /      |   |  '    \   \   ' \ |
     ;   :    /       '   :  |     :   '  |--"
      \   \ .'        ;   |.'       \   \ ;
   www. `---` ver     '---' he       '---" ire.org


Welcome to OverTheWire!

If you find any problems, please report them to the #wargames channel on
discord or IRC.
```



```
bandit0@bandit: ~
 by default, although ASLR has been switched off.  The following
 compiler flags might be interesting:

   -m32                    compile for 32bit
   -fno-stack-protector    disable ProPolice
   -Wl,-z,norelro          disable relro

 In addition, the execstack tool can be used to flag the stack as
 executable on ELF binaries.

 Finally, network-access is limited for most levels by a local
 firewall.

--[ Tools ]--

 For your convenience we have installed a few useful tools which you can find
 in the following locations:

   * gef (https://github.com/hugsy/gef) in /opt/gef/
   * pwndbg (https://github.com/pwndbg/pwndbg) in /opt/pwndbg/
   * peda (https://github.com/longld/peda.git) in /opt/peda/
   * gdbinit (https://github.com/gdbinit/Gdbinit) in /opt/gdbinit/
   * pwntools (https://github.com/Gallopsled/pwntools)
   * radare2 (http://www.radare.org/)

--[ More information ]--

 For more information regarding individual wargames, visit
 http://www.overthewire.org/wargames/

 For support, questions or comments, contact us on discord or IRC.

 Enjoy your stay!

bandit0@bandit:~$
```

Level 0-1

Donate! Help!?

Bandit Level 0 → Level 1

Level Goal

The password for the next level is stored in a file called **readme** located in the home directory. Use this password to log into bandit1 using SSH. Whenever you find a password for a level, use SSH (on port 2220) to log into that level and continue the game.

Commands you may need to solve this level

ls , cd , cat , file , du , find

Step 1 :

```
bandit0@bandit:~$ ls
readme
bandit0@bandit:~$ cat readme
Congratulations on your first steps into the bandit game!!
Please make sure you have read the rules at https://overthewire.org/rules/
If you are following a course, workshop, walthrough or other educational activity,
please inform the instructor about the rules as well and encourage them to
contribute to the OverTheWire community so we can keep these games free!

The password you are looking for is: ZjLjTmM6FvvyRnrb2rfNWOZOTa6ip5If

bandit0@bandit:~$
```

Type

“ls”

This command is used to view the list of the contents of the directory.

Step 2:There is a readme file in the directory, type the

“cat readme “

command, which is used to display the content of files in the terminal, in this case we are displaying the contents of the “readme” file which contains the password for the next level.

The Password is => **ZjLjTmM6FvvyRnrb2rfNWOZOTa6ip5If**

Level 1-2

Bandit Level 1 → Level 2

Level Goal

The password for the next level is stored in a file called - located in the home directory

Commands you may need to solve this level

ls , cd , cat , file , du , find

Step 1:

```
bandit0@bandit: ~        Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Bayu Ibana> ssh bandit1@bandit.labs.overthewire.org -p2220
                        _
                       | |__   __ _ _ __   __| (_) |_
                       | '_ \ / _` | '_ \ / _` | | __|
                       | |_) | (_| | | | | (_| | | |_
                       |_.__/ \__,_|_| |_|\__,_|_|\__|


                    This is an OverTheWire game server.
            More information on http://www.overthewire.org/wargames

bandit1@bandit.labs.overthewire.org's password:
```

Open another tab in the windows powershell and type:

```
ssh bandit1@bandit.labs.overthewire.org -p2220
```

then enter the password that we got from the last level to continue. this step is done repeatedly every time we enter a new level.

Step 2

```
--[ More information ]--

  For more information regarding individual wargames, visit
  http://www.overthewire.org/wargames/

  For support, questions or comments, contact us on discord or IRC.

  Enjoy your stay!

bandit1@bandit:~$ ls
-
bandit1@bandit:~$ cat ./-
263JGJPfgU6LtdEvgfWU1XP5yac29mFx
bandit1@bandit:~$
```

Type

```
"ls"
```

the file "-" will be displayed in the directory

Step 3

Type

```
"cat ./-"
```

use the cat command again to display the contents of the file, however since it is named "-" we need to add "./" in the command to open the file. the file also contains the password for the next level: **263JGJPfgU6LtdEvgfWU1XP5yac29mFx**

Level 2-3

Bandit Level 2 → Level 3

Level Goal

The password for the next level is stored in a file called **spaces in this filename** located in the home directory

Commands you may need to solve this level

ls , cd , cat , file , du , find

Step 1
Open another Windows Powershell tab, repeat the process in the previous level, this time connect to :

```
ssh bandit2@bandit.labs.overthewire.org -p2220
```

Step 2

```
bandit2@bandit:~$ ls
spaces in this filename
bandit2@bandit:~$ cat "spaces in this filename"
MNk8KNH3Usiio41PRUEoDFPqfxLPlSmx
bandit2@bandit:~$
```

Type

```
“ls”
```

the file “spaces in this filename” will be displayed in the directory.

Step 3
Type

```
“ cat “spaces in this filename” ”
```

This time we are opening the file “spaces in the filename”, which according to the name has spaces in it. Because of this, we must use “ “ after cat to view the contents of the file. It will then reveal the password for the next level: **MNk8KNH3Usiio41PRUEoDFPqfxLPlSmx**

Level 3-4

Bandit Level 3 → Level 4

Level Goal

The password for the next level is stored in a hidden file in the **inhere** directory.

Commands you may need to solve this level

ls , cd , cat , file , du , find

Step 1
Open another Windows Powershell tab, repeat the process in the previous level, this time connect to :

```
ssh bandit3@bandit.labs.overthewire.org -p2220
```

```
bandit3@bandit:~$ ls
inhere
bandit3@bandit:~$ cd inhere
bandit3@bandit:~/inhere$ ls
bandit3@bandit:~/inhere$ ls -a
.  ..  ...Hiding-From-You
bandit3@bandit:~/inhere$ cat "...Hiding-From-You"
2WmrDFRmJIq3IPxneAaMGhap0pFhF3NJ
bandit3@bandit:~/inhere$
```

Step 2
Type

```
"ls"
```

The File "inhere" contains files inside of it, so we must change the directory to view its contents. We will use the "cd" command. type:

```
"cd inhere"
```

Step 3
However, when we type "ls" again, there are no files shown, so we try using "ls -a", this way, files starting with "." will be revealed/not ignored.'

Step 4
Then once the file is revealed we then can open it using

```
cat "...Hiding-From-You"
```

The password for the next level : **2WmrDFRmJIq3IPxneAaMGhap0pFhF3NJ**

Level 4-5

Bandit Level 4 → Level 5

Level Goal

The password for the next level is stored in the only human-readable file in the **inhere** directory. Tip: if your terminal is messed up, try the "reset" command.

Commands you may need to solve this level

ls , cd , cat , file , du , find

Step 1

Open another Windows Powershell tab, repeat the process in the previous level, this time connect to :

```
ssh bandit4@bandit.labs.overthewire.org -p2220
```

```
bandit4@bandit:~$ ls
inhere
bandit4@bandit:~$ cd inhere
bandit4@bandit:~/inhere$ ls
-file00  -file01  -file02  -file03  -file04  -file05  -file06  -file07  -file08  -file09
bandit4@bandit:~/inhere$ file -- *
-file00: data
-file01: data
-file02: data
-file03: data
-file04: data
-file05: data
-file06: data
-file07: ASCII text
-file08: data
-file09: data
```

Step 2

Type "ls" again and change the directory to "inhere" again just like the previous level.

Step 3

Type "ls" to reveal the contents of "inhere" which will result in multiple files. however we are tasked with finding a file with "human-readable text" which is also called an ASCII text.
to do that, type

```
"file -- *"
```

The "file" command is used to determine the type of all the files in the directory. using this will help find the file with ASCII text.
Once done, it will reveal that every file has a type of data except for -file07 which has human-readable text.

Step 4

```
bandit4@bandit:~/inhere$ cat ./-file07
4oQYVPkxZOOEOO5pTW81FB8j8lxXGUQw
bandit4@bandit:~/inhere$
```

Type

```
"cat ./-file07"
```

Just like the previous levels, we use the "cat" command to view the contents of files.
The password for the next level : **4oQYVPkxZOOEOO5pTW81FB8j8lxXGUQw**

Level 5-6

Bandit Level 5 → Level 6

Level Goal

The password for the next level is stored in a file somewhere under the **inhere** directory and has all of the following properties:

human-readable
1033 bytes in size
not executable

Step 1
Open another Windows Powershell tab, repeat the process in the previous level, this time connect to :

```
ssh bandit5@bandit.labs.overthewire.org -p2220
```

Step 2

```
bandit5@bandit:~$ ls
inhere
bandit5@bandit:~$ cd inhere
```

use the “ls” command and change the directory to “inhere”

Step 3

```
bandit5@bandit:~/inhere/maybehere07$ file -- *
-file1:       ASCII text, with very long lines (3662)
-file2:       ASCII text, with very long lines (2487)
-file3:       data
spaces file1: ASCII text, with very long lines (4129)
spaces file2: ASCII text, with very long lines (9063)
spaces file3: data
```

use the “file – *” command again to find files with ASCII text

Step 4

```
bandit5@bandit:~/inhere/maybehere07$ find . -size 1033c ! -executable
./.file2
```

Find the last 2 conditions for the file we are looking for, which is 1033 bytes in size and not executable, using

```
find . -size 1033c ! -executable
```

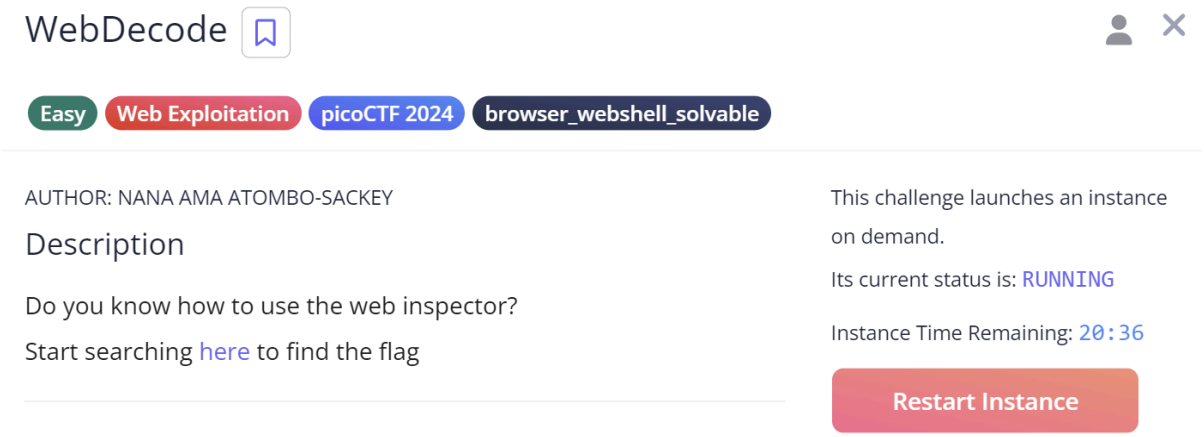
Step 5
Now that we found the file, which is “file2”,

```
bandit5@bandit:~/inhere/maybehere07$ cat ./.file2
HWasnPhtq9AVKe0dmk45nxy20cvUa6EG
```

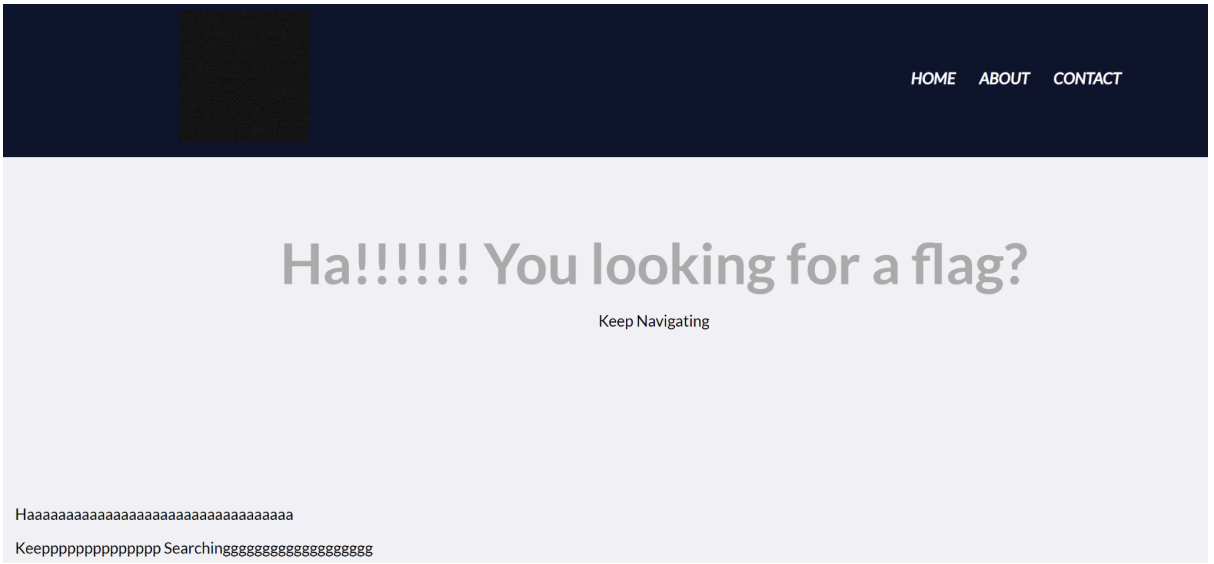
use the “cat” command to view the contents of the file, which is the password for the next level : **HWasnPhtq9AVKe0dmk45nxy20cvUa6EG**

PicoCTF Practice Questions

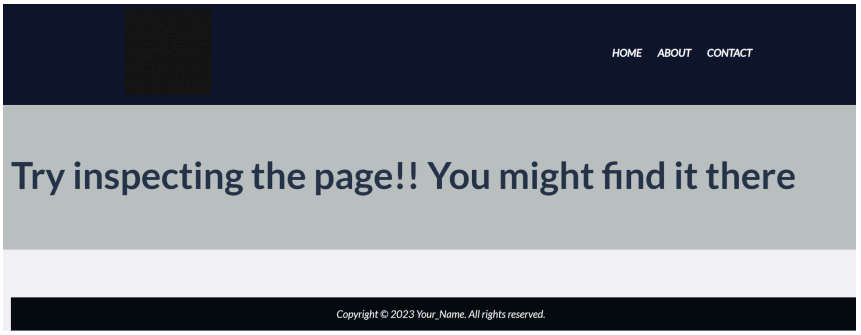
1. WebDecode



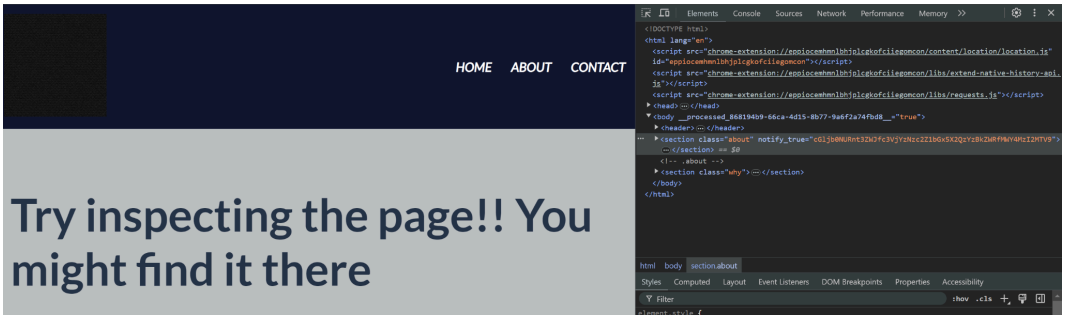
Step 1: start the instance and click “here”



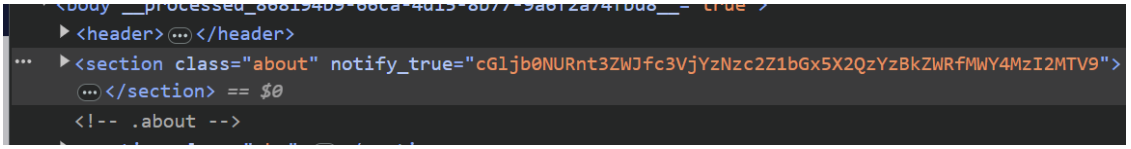
Step 2: when we open the page, there is a text “Keep Navigating” so we try to navigate to the other tabs. lets try the “About” tab



Step 3: Inspect the page according to the hint by right clicking, and then go to inspect

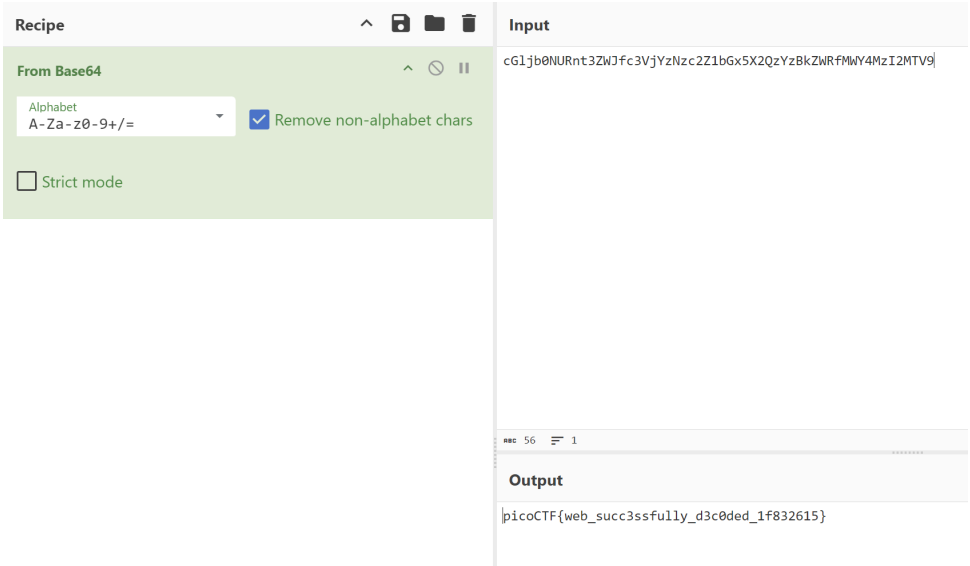


Step 4:



In the inspector, we found a suspicious string text, this string might be encoded so we will try to decode the data, with CyberChef

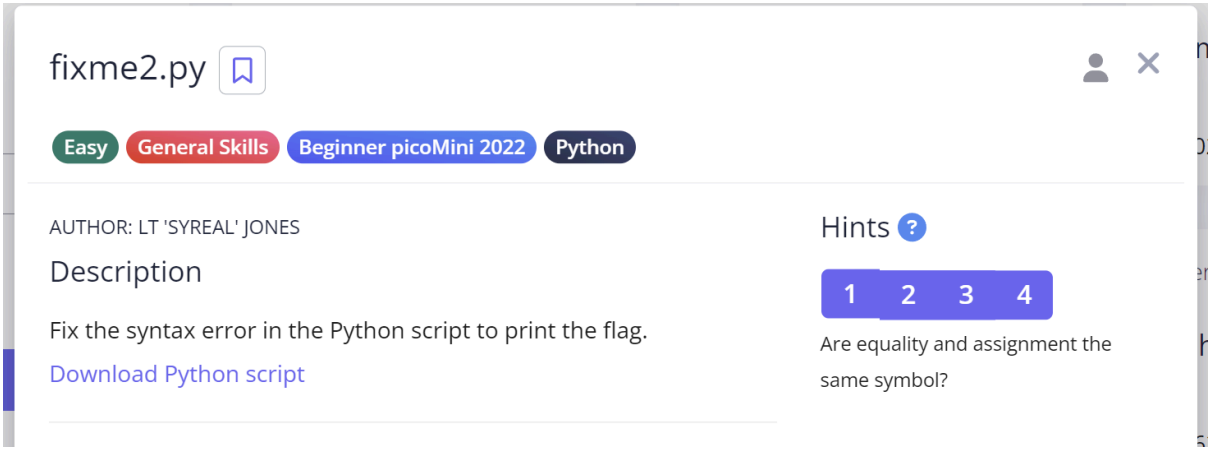
Step 5:



Choose the recipe **From Base64,** because we are decoding a base64 string to its base format. from that, we get the flag:

picoCTF{web_succ3ssfully_d3c0ded_1f832615}

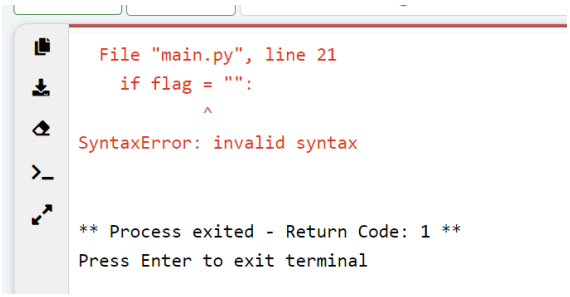
2. fixme2.py



Step 1:
Download the Python script and open it

```python
def str_xor(secret, key):
    #extend key to secret length
    new_key = key
    i = 0
    while len(new_key) < len(secret):
        new_key = new_key + key[i]
        i = (i + 1) % len(key)
    return "".join([chr(ord(secret_c) ^ ord(new_key_c)) for (secret_c,new_key_c) in zip(secret,new_key)])


flag_enc = chr(0x15) + chr(0x07) + chr(0x08) + chr(0x06) + chr(0x27) + chr(0x21) + chr(0x23) + chr(0x15) + chr(0x58) + chr(0x18) + chr(0x11) + chr


flag = str_xor(flag_enc, 'enkidu')

# Check that flag is not empty
if flag = "":
  print('String XOR encountered a problem, quitting.')
else:
  print('That is correct! Here\'s your flag: ' + flag)
```

Step 2:
Try to run the python file, we can see that there is an error in the script



```
  File "main.py", line 21
    if flag = "":
            ^
SyntaxError: invalid syntax


** Process exited - Return Code: 1 **
Press Enter to exit terminal
```

Step 3:

The error comes from the invalid syntax, it is currently “=” which is an assignment operator, we will change that to a comparison operator “==”. with this, the code when run will then check for a flag. If there is a flag then the code will print out the flag.

```
# Check that flag is not empty
if flag == "":
  print('String XOR encountered a problem, quitting.')
else:
  print('That is correct! Here\'s your flag: ' + flag)
```

Ln: 21, Col: 11

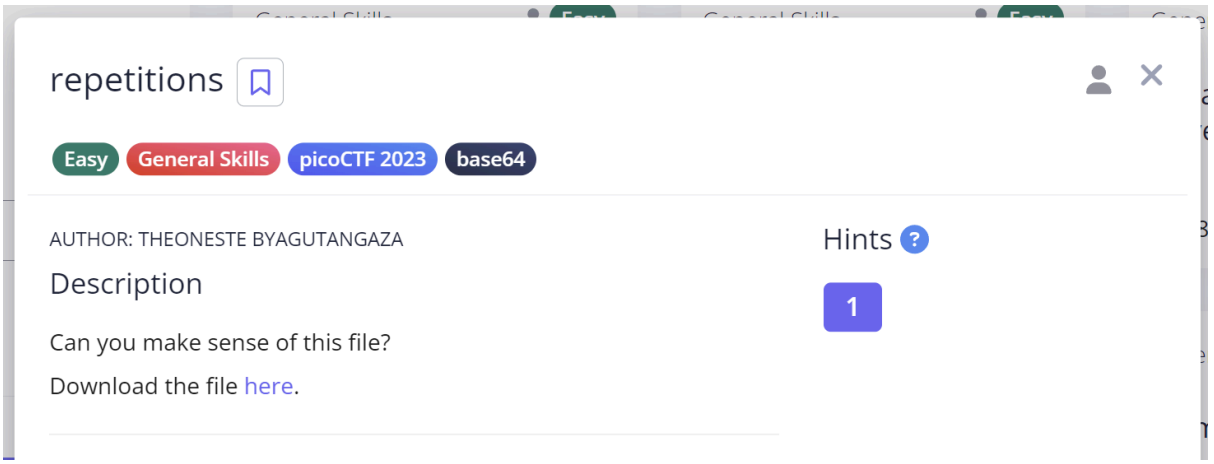
Run | Share | Command Line Arguments

```
That is correct! Here's your flag: picoCTF{3qu4l1ty_n0t_4551gnm3nt_e8814d03}
```

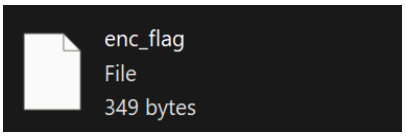
Once you run the code, the flag will be printed

picoCTF{3qu4l1ty_n0t_4551gnm3nt_e8814d03}

3. repetitions



Step 1: let's look at the file by downloading it

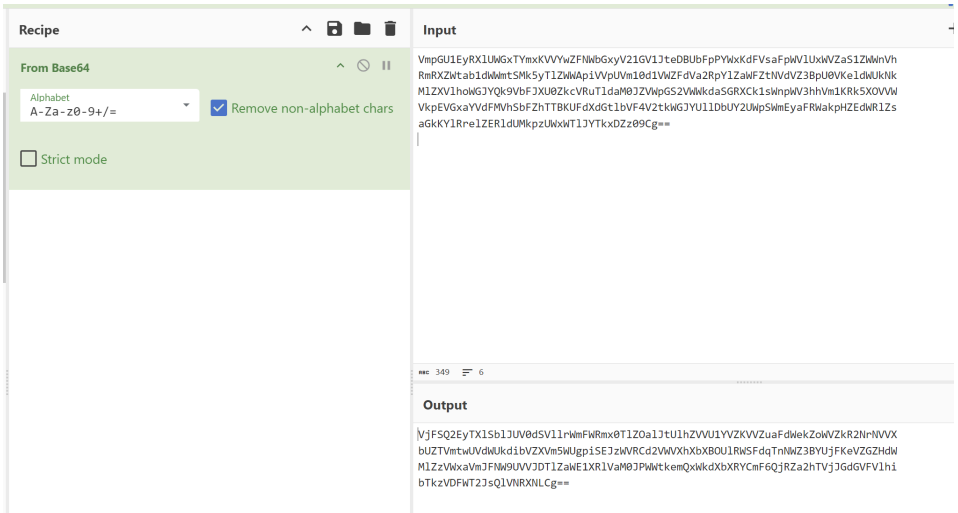


File Edit View

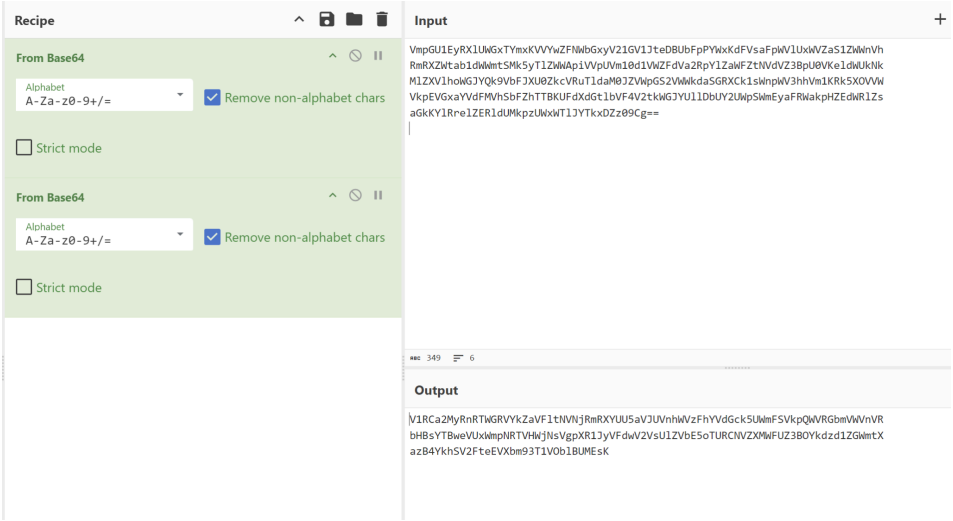

```
VmpGU1EyRXlUWGxTYmxKVVYwZFNWbGxyV21GV1JteDBUbFpPYWxKdFVsaFpWVlUxWVZaS1ZWWnVh
RmRXZWtab1dWWmtSMk5yTlZWWApiVVpUVm10d1VWZFdVa2RpYlZaWFZtNVdVZ3BpU0VKeldWUkNk
MlZXVlhoWGJYQk9VbFJXU0ZkcVRuTldaM0JZVWpGS2VWWkdaSGRXCk1sWnpWV3hhVm1KRk5XOVVW
VkpEVGxaYVdFMVhSbFZhTTBKUFdXdGtlbVF4V2tkWGJYUllDbUY2UWpSWmEyaFRWakpHZEdWRlZs
aGkKYlRrelZERldUMkpzUWxWTlJYTkxDZz09Cg==
```


When opened with Notepad, we get this long text.

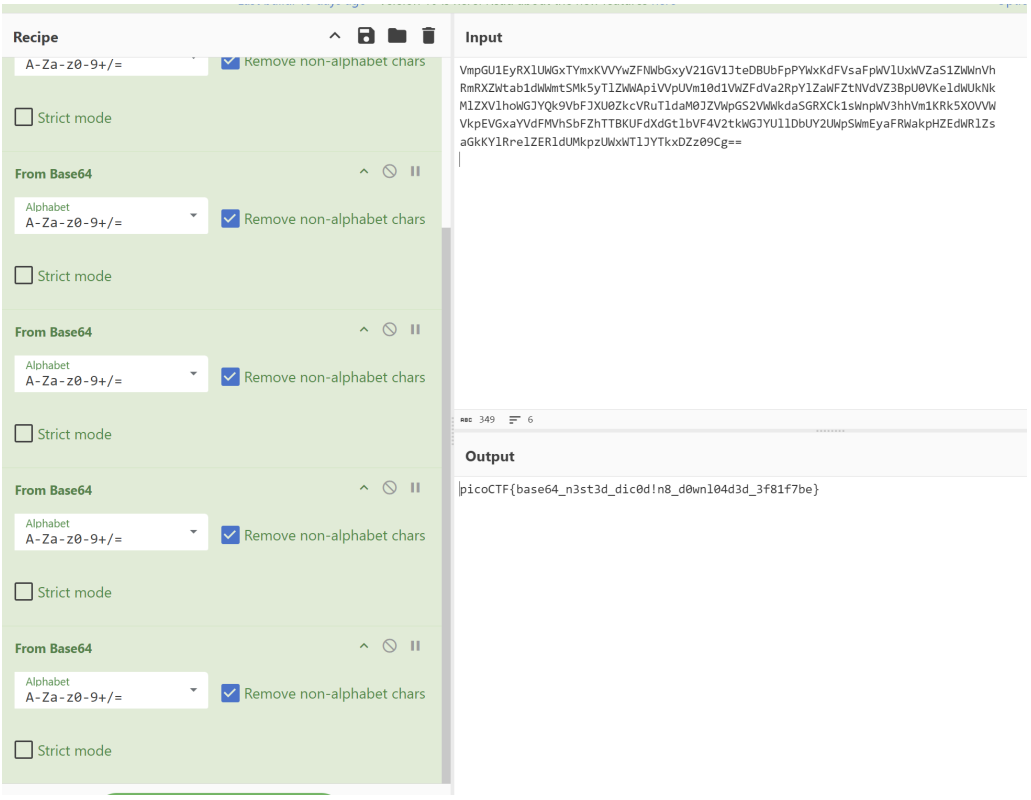
Step 2: Since this challenge is in the base64 category, let's try and decode this text using CyberChef, with the recipe of **From Base64.**



Step 3: when decoded, the flag has not been discovered yet. however it seems like we could still decode the text again with the same recipe.



Step 4: There is a bit of a pattern where the text gets shorter but it looks like it can be decoded again and again still, so we'll repeat the process again.



Just like the name of the challenge, "repetitions" eventually when we repeat the recipe to decode the text again and again the flag is finally discovered after repeating the process 6 times.

picoCTF{base64_n3st3d_dic0d!n8_d0wnl04d3d_3f81f7be}

4. convertme.py

convertme.py

Easy | General Skills | Beginner picoMini 2022 | base | Python

AUTHOR: LT 'SYREAL' JONES

Description

Run the Python script and convert the given number from decimal to binary to get the flag.

Download Python script

Step 1: download the python script and open it

```python

import random



def str_xor(secret, key):
    #extend key to secret length
    new_key = key
    i = 0
    while len(new_key) < len(secret):
        new_key = new_key + key[i]
        i = (i + 1) % len(key)
    return "".join([chr(ord(secret_c) ^ ord(new_key_c)) for (secret_c,new_key_c) in zip(secret,new_key)])


flag_enc = chr(0x15) + chr(0x07) + chr(0x08) + chr(0x06) + chr(0x27) + chr(0x21) + chr(0x23) + chr(0x15) + chr(0x5f) + chr(0x05) + chr(0x08) + chr(0x2a) + chr(0x1c) +


num = random.choice(range(10,101))

print('If ' + str(num) + ' is in decimal base, what is it in binary base?')

ans = input('Answer: ')

try:
  ans_num = int(ans, base=2)

  if ans_num == num:
    flag = str_xor(flag_enc, 'enkidu')
    print('That is correct! Here\'s your flag: ' + flag)
  else:
    print(str(ans_num) + ' and ' + str(num) + ' are not equal.')

except ValueError:
  print('That isn\'t a binary number. Binary numbers contain only 1\'s and 0\'s')

```

Step 2: Run the file

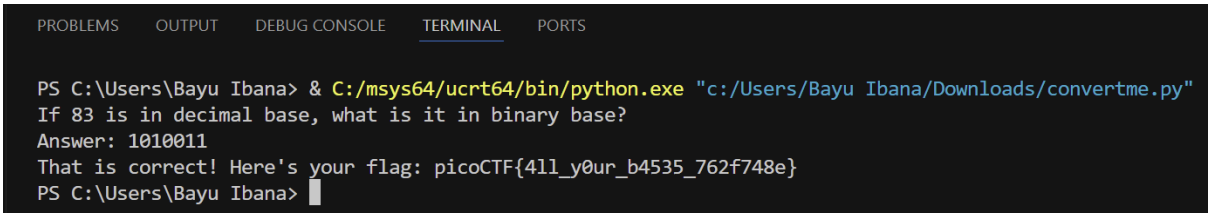
```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\Bayu Ibana> & C:/msys64/ucrt64/bin/python.exe "c:/Users/Bayu Ibana/Downloads/convertme.py"
If 83 is in decimal base, what is it in binary base?
Answer:
```

When we run the terminal, it will ask us to answer the question “If 83 is in the decimal base, what is it in binary base?”

Step 3: So, we convert 83 from decimal to binary using an online converter,

In this case, I used RapidTables.

Step 4: Paste the binary number in the terminal to answer the previous question,



we discovered the flag after getting the password correct:

picoCTF{4ll_y0ur_b4535_762f748e}