SarCTF

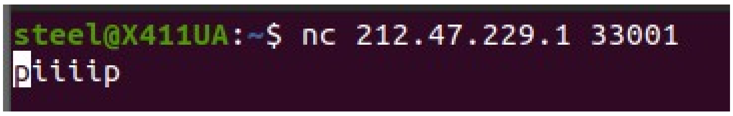
PPC : Mind Palace I

Description : It looks like the situation is hopeless, there is no time to think. However, you can use the mind palace and solve all problems instantly.

nc 212.47.229.1 33001

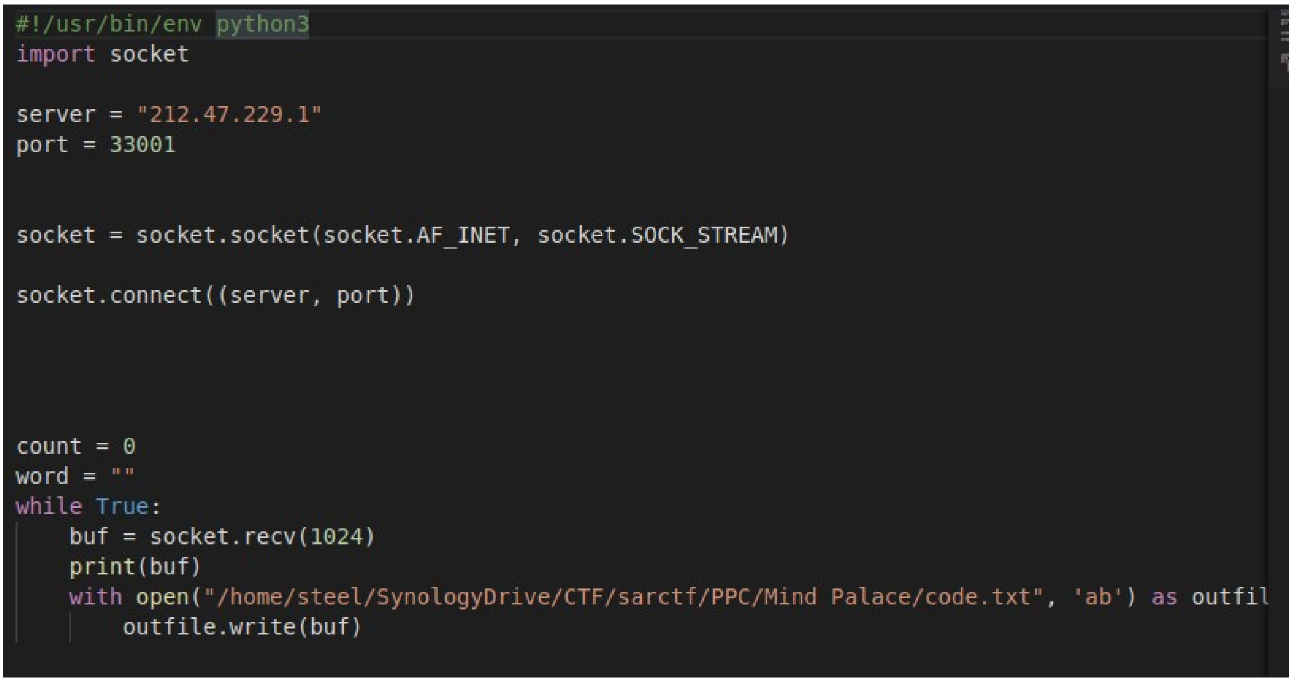
Solution :

The netcat command given in the description shows us some words blinking on the screen.



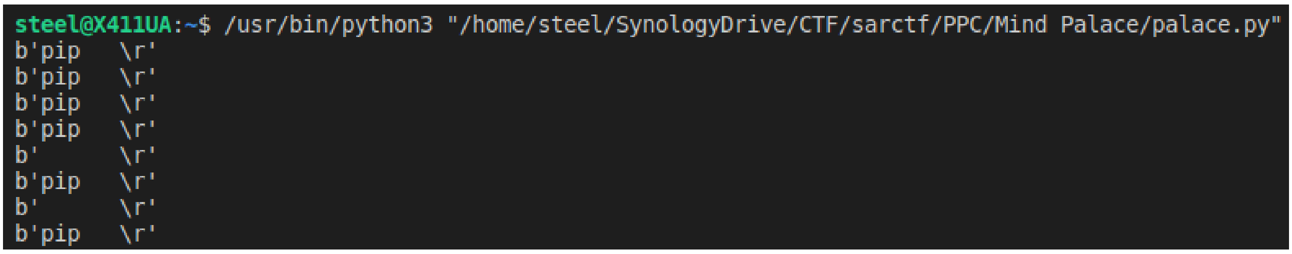
```
steel@X411UA:~$ nc 212.47.229.1 33001
piiiip
```

It seems like an encoded message. Let’s program something to save the message.



```
#!/usr/bin/env python3
import socket

server = "212.47.229.1"
port = 33001


socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

socket.connect((server, port))




count = 0
word = ""
while True:
    buf = socket.recv(1024)
    print(buf)
    with open("/home/steel/SynologyDrive/CTF/sarctf/PPC/Mind Palace/code.txt", 'ab') as outfil
        outfile.write(buf)
```

This script will save everything into a file called ‘code.txt’.

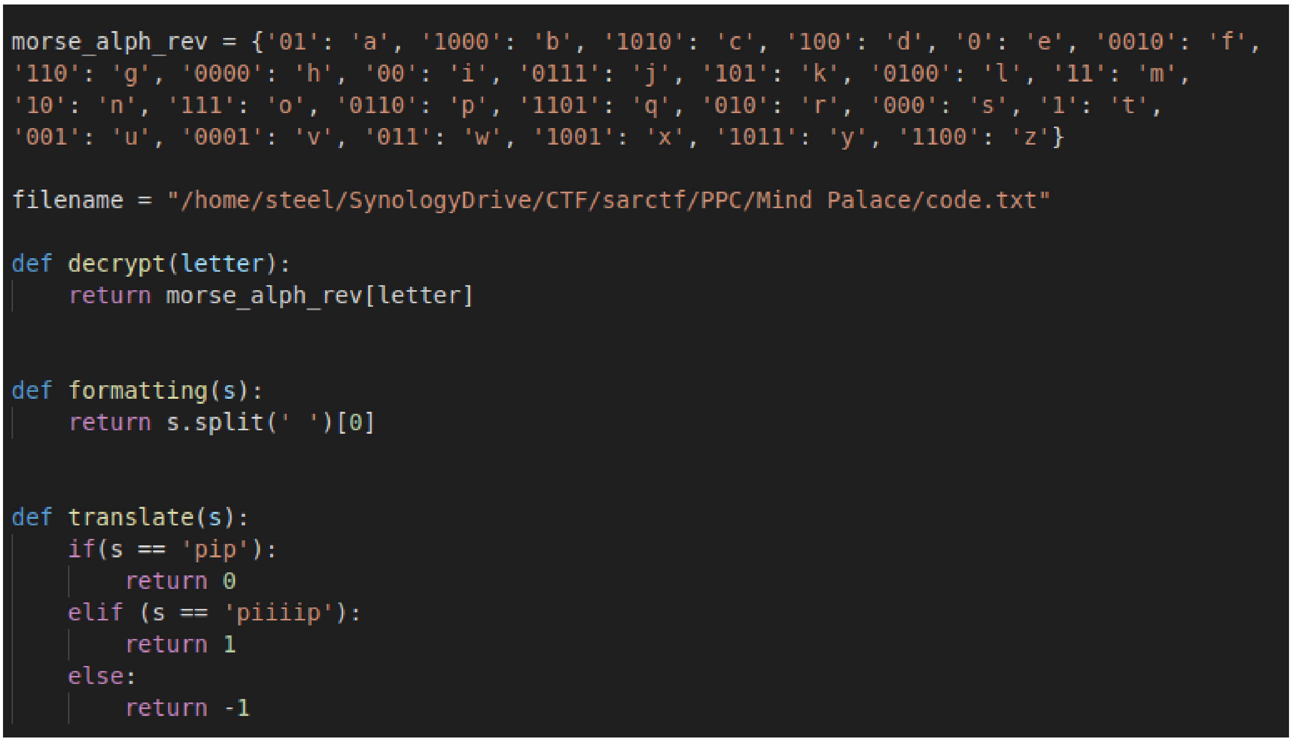


```
steel@X411UA:~$ /usr/bin/python3 "/home/steel/SynologyDrive/CTF/sarctf/PPC/Mind Palace/palace.py"
b'pip   \r'
b'pip   \r'
b'pip   \r'
b'pip   \r'
b'      \r'
b'pip   \r'
b'      \r'
b'pip   \r'
```

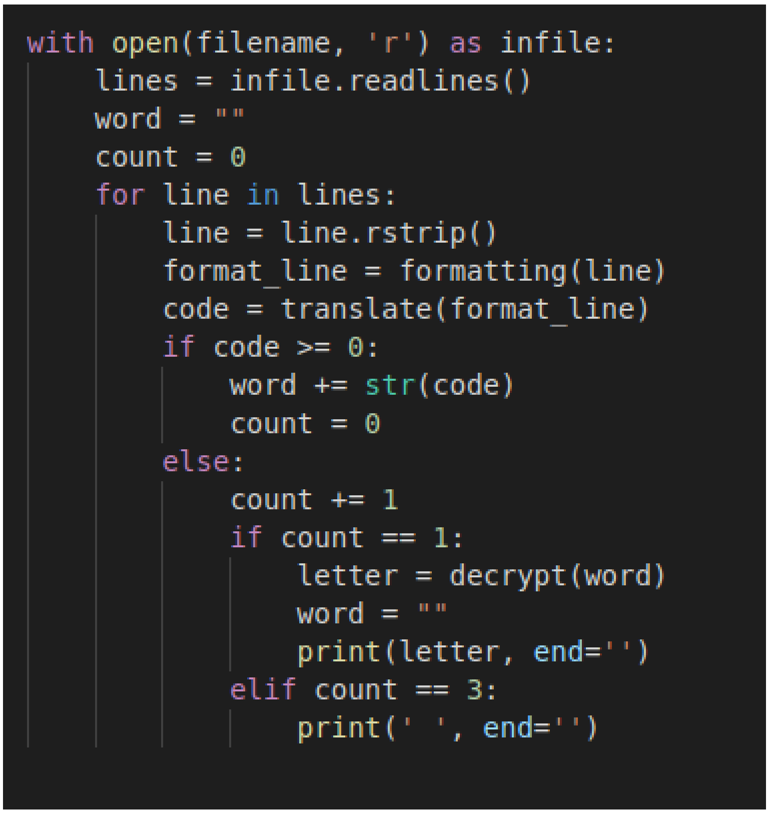
There are three types of lines sent by the server: nothing, ‘bip’ and ‘biiiip’. We can see it as a short signal, a long signal and no signal at all. It seems like morse code. We can try to decrypt it.

There are two types of ‘no signal’: one line with nothing, and three lines in a row with nothing. We can think that the first one separates letters and the second one separate words.

In this program, one long signal will be ‘1’ , one short signal will be ‘0’ and no signal will be ‘-1’. So the string ‘1010’ will be one long, one short, one long and one short in a row. Each combination will be stored with the corresponding letter in an array. Every time three -1 are spotted in a row, a space is printed on the screen.

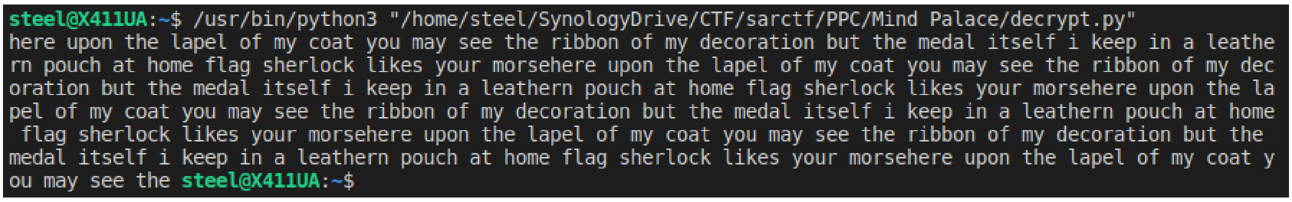


```
morse_alph_rev = {'01': 'a', '1000': 'b', '1010': 'c', '100': 'd', '0': 'e', '0010': 'f',
'110': 'g', '0000': 'h', '00': 'i', '0111': 'j', '101': 'k', '0100': 'l', '11': 'm',
'10': 'n', '111': 'o', '0110': 'p', '1101': 'q', '010': 'r', '000': 's', '1': 't',
'001': 'u', '0001': 'v', '011': 'w', '1001': 'x', '1011': 'y', '1100': 'z'}

filename = "/home/steel/SynologyDrive/CTF/sarctf/PPC/Mind Palace/code.txt"

def decrypt(letter):
    return morse_alph_rev[letter]


def formatting(s):
    return s.split(' ')[0]


def translate(s):
    if(s == 'pip'):
        return 0
    elif (s == 'piiiip'):
        return 1
    else:
        return -1
```



```python
with open(filename, 'r') as infile:
    lines = infile.readlines()
    word = ""
    count = 0
    for line in lines:
        line = line.rstrip()
        format_line = formatting(line)
        code = translate(format_line)
        if code >= 0:
            word += str(code)
            count = 0
        else:
            count += 1
            if count == 1:
                letter = decrypt(word)
                word = ""
                print(letter, end='')
            elif count == 3:
                print(' ', end='')
```

This code translates the morse code in the saved file.



```
steel@X411UA:~$ /usr/bin/python3 "/home/steel/SynologyDrive/CTF/sarctf/PPC/Mind Palace/decrypt.py"
here upon the lapel of my coat you may see the ribbon of my decoration but the medal itself i keep in a leathe
rn pouch at home flag sherlock likes your morsehere upon the lapel of my coat you may see the ribbon of my dec
oration but the medal itself i keep in a leathern pouch at home flag sherlock likes your morsehere upon the la
pel of my coat you may see the ribbon of my decoration but the medal itself i keep in a leathern pouch at home
 flag sherlock likes your morsehere upon the lapel of my coat you may see the ribbon of my decoration but the
medal itself i keep in a leathern pouch at home flag sherlock likes your morsehere upon the lapel of my coat y
ou may see the steel@X411UA:~$
```

We can see the flag in the text:

flag{sherlock_likes_your_morse}