NeverLAN CTF

Programming : Robot Talk

Value : 200 pts

Difficulty : Easy

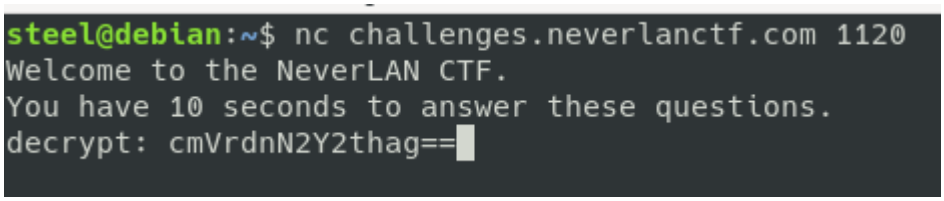
Description : Your flag will be in the normal flag{flagGoesHere} syntax.

Attachment : This server only gives the flag to bots. You'll need to convince it that you're a bot by answering it's challenges.

challenges.neverlanctf.com:1120

Solution :

The port given can’t be accessed with Firefox. Let’s try netcat to see what it returns.



That’s a little game where we need to decrypt base64 in 10 seconds. Let’s do a bot for it.

```python
import socket
import base64

server = "challenges.neverlanctf.com"
port = 1120

def decrypt(s):
    l = s.split(' ')
    print(l[1])
    res = base64.b64decode(l[12])
    return res

socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
socket.connect((server, port))

buf = socket.recv(1024).decode("utf-8")
print(buf)

res = decrypt(buf)
print(res)
socket.send(res)
buf = socket.recv(1024).decode('utf-8')
print(buf)
```

This programs receives the encoded string, decode it, send the answer to the server and receives the response of the server, maybe the flag.

```
steel@X411UA:~$ /usr/bin/python3 /home/steel/SynologyDrive/CTF/Neverlan/Programming/robot-talk-example.py
Welcome to the NeverLAN CTF.
You have 10 seconds to answer these questions.
decrypt: d2RhbWhpaGF3eg==
b'wdamhihawz'
Awesome, continuing.
```

That’s not the flag. It seems that we’ll need to do a loop in the program.

```python
def decrypt_first(s):
    l = s.split(' ')
    print(l[1])
    res = base64.b64decode(l[12])
    return res

def decrypt(s):
    l = s.split(' ')
    print(l[1])
    res = base64.b64decode(l[2])
    return res

def chall(socket, buf):
    continuing = True
    while continuing:
        res = decrypt(buf)
        print(res)
        socket.send(res)
        buf = socket.recv(1024).decode('utf-8')
        print(buf)

        if not 'continuing' in buf:
            continuing = False
```

The first message is different, so an other function is used. The core of the program looks now like this.
With this program, we can decrypt all the encoded strings and earn the flag!

```
b'qemaxecqca'
Awesome, continuing.
decrypt: ZmdhZHhlc25zYQ==
continuing.
decrypt:
b'fgadxesnsa'
Awesome, continuing.
flag{Ant1_hum4n}
continuing.
flag{Ant1_hum4n}
Traceback (most recent call last):
  File "robot-talk.py", line 43, in <module>
    chall(socket, buf)
  File "robot-talk.py", line 22, in chall
    res = decrypt(buf)
  File "robot-talk.py", line 16, in decrypt
    res = base64.b64decode(l[2])
IndexError: list index out of range
steel@debian:~/CTF/Neverlan/Programming$
```

flag{Ant1_hum4n}