Week0- Write-up

Vishnu Vardhan Ciripuram
N14912012
vc2499

Challenge Vault0 :

Objective : locate the secret_vault function within the vault0 binary and submit its address in decimal format.

Steps used to Solve:

1. **Examining the Binary:** I used file vault0 to learn that the binary was an ELF 64-bit LSB executable, not stripped, and without PIE (Position Independent Executable), meaning its memory addresses were fixed.
2. **Finding secret_vault with nm:** Using "nm vault0 | grep secret_vault" , I found the secret_vault function located at 0x401236.
3. **Converting to Decimal:** Since the address needed to be in decimal, I converted 0x401236 to 4198966.
4. **Submission:** 4198966

```
┌──(kali㉿kali)-[~]
└─$ nc offsec-chalbroker.osiris.cyber.nyu.edu 1230
Please input your NetID (something like abc123): vc2499
hello, vc2499. Please wait a moment ...
Can you tell me the address of the secret vault?

> 4198966
Lucky me! that's my favorite vault!

Here's your flag, friend: flag{Th3_g00d_0ld_d4ys_0f_N0_PIE!_8b7be907bc7619be}
```

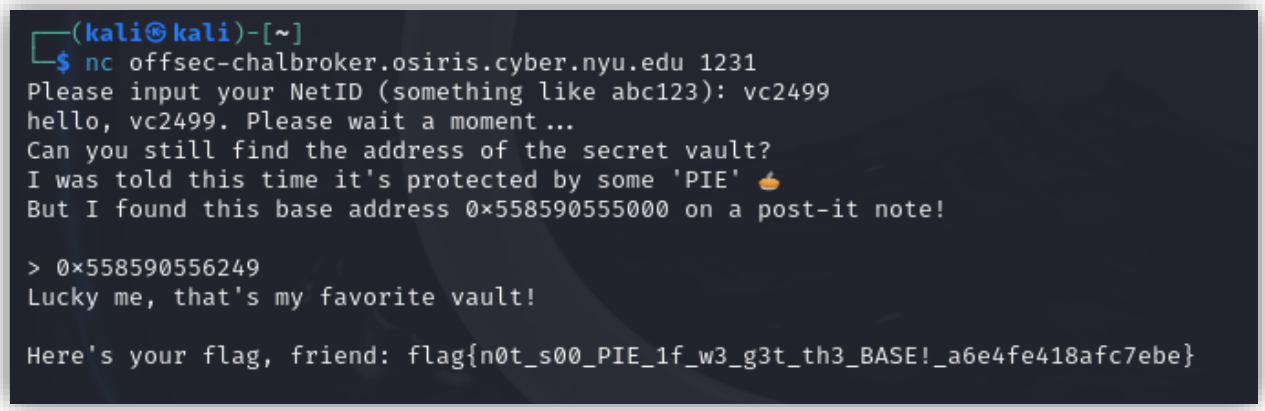
Challenge Vault1:

Objective : Find the address of the secret_vault function, but this time the binary was protected by PIE (Position Independent Executable).

Steps used to Solve:

1. **Given Base Address:** The challenge provided the base address which was 0x558590555000.
2. **Finding the secret_vault Offset:** Using the nm command (nm vault | grep secret_vault), I found the offset of the secret_vault function, which was 0x1249.

3. **Calculating the Final Address:** To find the address of secret_vault, I added the base address to the offset:
   Address = 0x558590555000 + 0x1249 = 0x558590556249
4. **Submission:** 0x558590556249

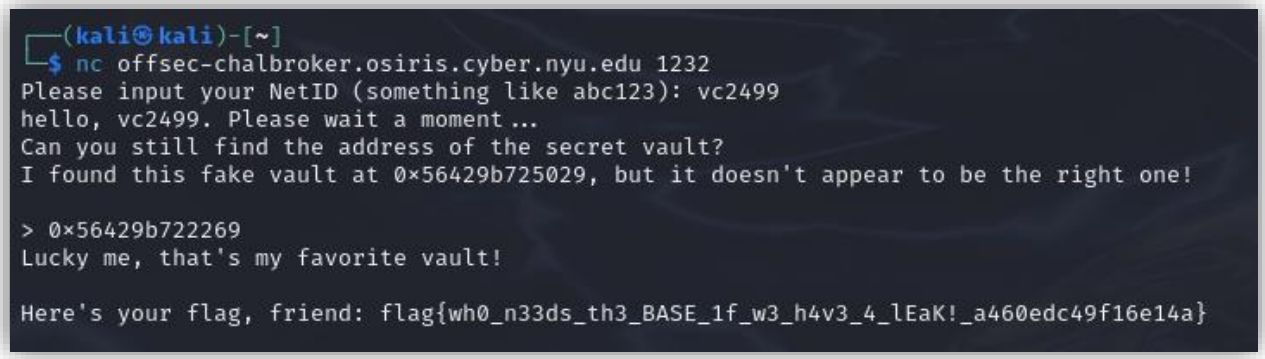


Challenge Vault2:

Objective: find the address of the secret_vault function, where the address to fake_vault is given.

Steps used to Solve:

1. **Finding offset**: I used the nm command to find the offsets for both fake_vault and secret_vault:
   Offset of fake_vault: 0x4029
   Offset of secret_vault: 0x1269
2. **Calculated the Base Address**: The fake vault's address was provided as 0x56429b725029. I calculated the base address by subtracting the fake_vault offset:
   Base Address = 0x56429b725029 - 0x4029 = 0x56429b721000
3. **Find secret_vault Address**: calculated the address of secret_vault by adding its offset to the base address:
   secret_vault_address = 0x56429b721000 + 0x1269 = 0x56429b722269
4. **Submission**: 0x56429b722269



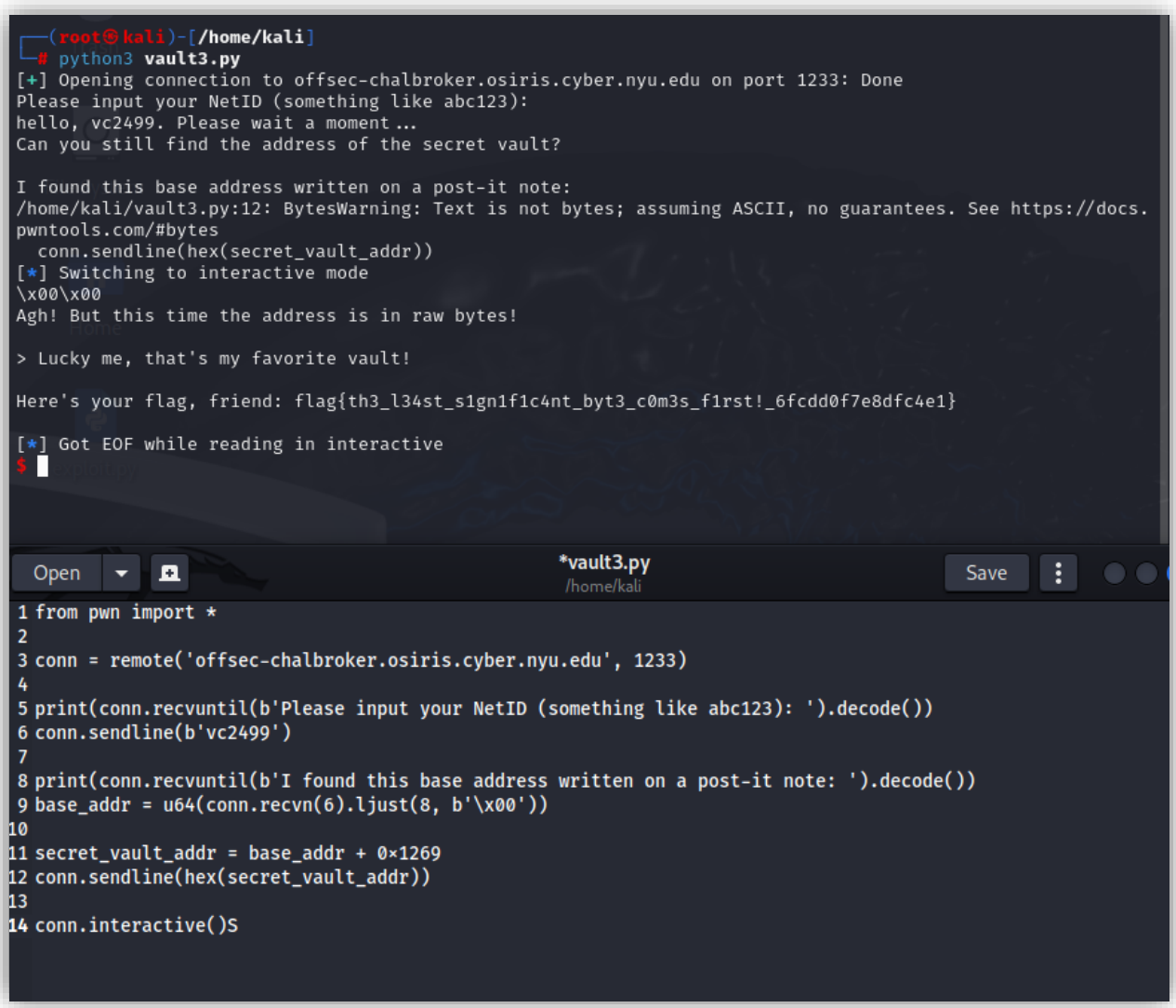
Challenge Vault3:

Objective: Need to find the secret vault address, but now the base address is given in raw format.

Steps used to Solve:

I wrote a script using pwntools to automate this process.

1. **Create Connection**: I used pwntools to establish a connection to the server:
   conn = remote('offsec-chalbroker.osiris.cyber.nyu.edu', 1233)
2. **Send NetID**: I sent my NetID as requested by the server:
   conn.sendline(b'vc2499')
3. **Calculated the Base Address**: After receiving a 6-byte base address from the server, I padded it to 8 bytes for 64-bit conversion:
   base_addr = u64(conn.recvn(6).ljust(8, b'\x00'))
4. **Find secret_vault Address**: I calculated the address of secret_vault by adding its offset (0x1269) which I found by running the "nm vault3 | grep secret_vault" command to the base address:
   secret_vault_addr = base_addr + 0x1269
5. **Submission**: I sent the calculated address to the server in hexadecimal:
   conn.sendline(hex(secret_vault_addr))
   conn.interactive()



Challenge Vault4:

Objective: find the address of the secret_vault function, where the address to fake_vault is given, but this time its in raw format.

Steps used to Solve:

I wrote a script using pwntools to automate this process.

1. **Create Connection**: I used pwntools to establish a connection to the server:
   conn = remote('offsec-chalbroker.osiris.cyber.nyu.edu', 1234)
2. **Send NetID**: I sent my NetID as requested by the server:
   conn.sendline(b'vc2499')

3. **Calculated the Base Address**: After receiving a 6-byte fake vault address from the server, I padded it to 8 bytes for 64-bit conversion:
   fake_vault_addr = u64(conn.recvn(6).ljust(8, b'\x00'))
   base_addr = fake_vault_addr - 0x4030
4. **Find secret_vault Address**: I calculated the address of the secret_vault by adding its offset (0x4038) to the base address:
   secret_vault_addr = base_addr + 0x4038
5. **Converting and Submission**: I used p64() to convert the address back too raw form (the server expected input in raw form) and sent first 6 bytes to the server:
   conn.send(p64(secret_vault_addr)[:6])
   conn.interactive()