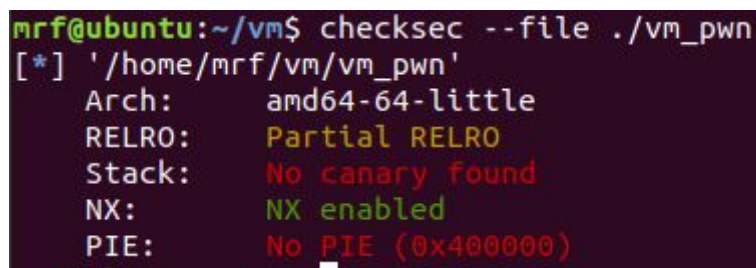


Этот task надо было решать после решения taska на реверс

(https://docs.google.com/document/d/1_dt270kJJhbvJoZH5Ah2fbeJHsDjnJmpplzZE0lDr-A/edit). Сравниваем 2 исполняемых файла, для этого можно использовать программы из интернета, к примеру вот эту https://github.com/CapeLeidokos/elf_diff. И так в файлах отличается лишь одна функция

```
1|int __fastcall int(signed __int16 *a1)
2|{
3|    int result; // eax
4|
5|    result = *a1;
6|    if ( result != 1 )
7|    {
8|        if ( result == 2 )
9|            exit(0);
10|        if ( !result )
11|            result = printf(bytecode + regs);
12|    }
13|    return result;
14|}
```

Как мы видим здесь вырезано прерывание для чтения ввода от пользователя. А также здесь мы можем увидеть уязвимость форматной строки. Воспользуемся программой checksec (<https://github.com/slimm609/checksec.sh>), чтобы узнать какие защитные механизмы включены.



```
mrfr@ubuntu:~/vm$ checksec --file ./vm_pwn
[*] '/home/mrf/vm/vm_pwn'
Arch:      amd64-64-little
RELRO:     Partial RELRO
Stack:     No canary found
NX:        NX enabled
PIE:       No PIE (0x400000)
```

Как мы видим у нас отключён PIE и RELRO (Relocations Read Only) включено частично, что означает, что мы можем писать в GOT таблицу. Но т.к нам нужно получить шелл на сервере, то нам надо как-то получить возможность слить адрес какого-то символа из libc. Но это сделать нельзя, поскольку мы не можем узнать что вывелось в стандартный вывод, а отослать его эксплойтом назад нельзя, потому-что вырезано прерывание для чтения пользовательского ввода. Но мы можем изменить адрес в GOT таблице с помощью инструкций для виртуальной машины, для этого нужно записать их в регистры виртуальной машины, поскольку адреса размером 2 байта, т.е писать слишком далеко мы не можем, а bss находится близко к GOT.

Итак вот план действий:

1. Записываем в регистры байткод, который меняет адреса в GOT.
2. Пишем кучу инструкций, которые не делают ничего, чтобы сделать указатель на текущую инструкцию равным $0x602178 - 0x602030 - 2$ (Адрес регистров в bss - адрес printf в GOT таблице - размер инструкции INT 0), чтобы он указывал на массив с регистрами.
3. Вызываем прерывание, которое переписывает значение глобальной переменной "bytecode" на адрес функции "printf" в GOT.
4. Вычесть из предпоследних двух байт адреса printf в GOT $0x15c$ и поменять значение младшего байта на $0xc2$.
5. Вызвать прерывание.

<https://pastebin.com/ZE4SvYZZ> - код компилятора. <https://pastebin.com/utJL1S9m> - код эксплойта. Код в регистрах - <https://pastebin.com/ExrcMStL>.
<https://pastebin.com/szsQJsM7> - скрипт, который отправляет эксплойт на сервер.

CTF{5up3r_34sy_VM_pwn}