CS: hw4 - fmtfun4u

學號：b04902053 姓名：鄭淵仁

flag: `FLAG{FEED_MY_TURTLE}`

script

- 可以取得 shell 的 python3 檔案： `fmtfun4u.py`

  (requirements: `pwntools`)

write-up

程式的漏洞

`fmtfun4u` 在讀入了 `buf` 之後馬上用 `printf` 把 `buf` 印出去，所以可以利用 format string 來 leak 位置和寫入資料。另外，利用 &argv → &argv[] → argv[] 的 chain 可以達成任意位置寫入。

leak 出 libc 和 stack 的 address

由於 `RELRO` 的保護是 `FULL` 的，所以要從 `stack` 上殘留的值 leak 出 `libc` 、 `argv` 和 function 的 return address 放的地方。leak 出這三個值之後就可以推算其他需要的 address。

修改迴圈數

為了方便之後大量寫入 format string ，所以就從 `argv` 的 address 推算出跑迴圈的 `i` 變數，把值改成 `0xffff` 就可以幾乎無限次數輸入 format string 了。

> 這個部分因為我是直接蓋 &argv[] 的 address 的後 2 個 byte 來改成 `i` 的 address 來改 `i` 的值的，所以有時候如果 `i` 和 &argv[] 的 address 差別比 2 個 byte 多就會失敗。但是這件事發生的機率不高，頂多再執行一次就會成功蓋到了。 這個部分是可以通過多蓋一次 1 個 byte 就可以解決的，只是我懶得改了 XD

把 `printf` 的 return address 改成 `pop rdi` 、 `";/bin/sh";` 、 `system`

只要達成標題寫的目標，就可以成功執行 `system(";/bin/sh";)` 了。

而為了達成這個目標，首先要把 `";/bin/sh";` 和 `system` 寫到 `printf` 的 return address 後面兩個 address，最後再把 `pop rdi` 蓋到 `printf` 的 return address 上，這樣一來 `printf` return 的時候就會執行我寫入的這三個 rop 了。

但是寫入 `system` 之後，再次執行 `printf` 就會動到 `system` 的其中一個 byte，所以要在蓋 `printf` 的 return address 的同時把 `system` 的那個 byte 改回來。

而為了做到上面的操作，會需要比 `buf` 還要長的字串。所以我就先用 format string 在 `buf` 後面寫入最後要送出的 payload 的 0x10 個 char 後的字串，之後再正常的讓程式讀入 payload 的前 0x10 個 char ，接起來就會是完整的 format string 了。

另外，為了最後能同時蓋到 `printf` 的 return address 和改回 `system` 被改掉的那個 byte，會需要兩個地方放 `printf` 的 return address 和 `system` 的 address。所以我除了把 `printf` 的 return address 放在 `argv[0]` 以外，還在 stack 上找了另一個位子充當 `argv[1]` ，用來放 `system` 的 address 。