b06705001 資管 3 楊力行 b06705001

0x08

election

memcmp 在比較 token 和 buf 時，會以 buf 的長度為基準，由於 token 的大小為 0xb8，而 buf 的大小為 0xc8，透過爆搜的方式逐 byte 找出 token 之後的 16 bytes， 也就是 canary 和 return address，用 gdb 可發現該 address 為 libc_csu_init 的 address，扣掉 offset 後就可得到 base address。

投票系統的傳送訊息的上限是 uint8 的最大值，也就是 255，而 msg 的大小為 0xe0 也就是 224，透過 GDB 發現 canary 在 0xe8 的位置，因此先把某個候選人的票數灌到 255 接者對他傳訊息，因為可用的空間不大，因而打算用 stack pivot，訊息為 232 個任意的，加上 canary，再加上 buf 的位址，打算把 stack 搬到 buf 去，再加上 leave ret。

接著就是寫 rop chain 了，BUF 的內容是是先透過 login 填入的，先 leak 出 libc_start_main 的位址 在減掉 offset 得到基址，再靠著 libc_csu_init ROP 出 read 接著再寫入我找到的 onegadget 加上 libc 基址後的位址，就成功 get shell。

我本機端能成功的 get shell，但是遠端透過 VPN 光是得出 CANARY 就要用到快 2 分鐘，因而沒有在期限內拿到 FLAG，明天上課會去教室試試看，如果可以的話希望能通融。