## Hydrogen

Se trata de parchear el binario para que ejecute el salto que nos de la pista que necesitamos para la contraseña. Se puede hacer con el debugger o directamente sobre las direcciones físicas. La flag está codificada de forma que no se pueda sacar con strings, hay una contraseña falsa para despistar.

## Solución

Ejecutamos con el debugger de radare2 (r2 -Ad hydrogen) y vemos la función main:

```
[0x7f9d8b2df100]> pdf @ main
 327: int
               (int argc, char **argv, char **envp);
           ; var int64_t var_59h @ rbp-0x59
; var int64_t var_58h @ rbp-0x58
           ; var int64_t var_28h @ rbp-0x28
                                                endbr64
                                f30f1efa
                                4889e5
                                                mov rbp, rsp
                                4883ec60
                                                sub rsp, 0x60
                                64488b042528.
                                                mov rax, qword fs:[0x28]
                                488945f8
                                                mov qword [var_8h], rax
                                31c0
                                c745a80000000.
                                                mov dword [var_58h], 0
                                                cmp dword [var_58h], 0
                                837da800
                                741d
                                488d3d800b00. lea rdi, str.Good_here_s_your_hint: ;
                                b8000000000
                                                mov eax, 0
                                                call sym.imp.printf : int printf(co
                                e832fcffff
                                b8000000000
                                                mov eax, 0
                                                call sym.show_hint
                                e807feffff
                                eb0c
                                488d3d7f0b00.
                                                lea rdi, str.you_don_t_want_a_hint__Uhm..._OK
                                e8eafbffff
                                                call sym.imp.puts
```

Se puede ver que en la posición 0x558955591480 se escribe el valor 0 en la variable var\_58h y justo debajo de comprueba si su valor es 0, si lo es se salta las lineas siguientes, entre las que vemos que se realiza una llamada a la función "show\_hint()", por lo tanto tenemos que parchear para que no efectúe ese salto.

Hacemos seek sobre esa dirección y escribimos 0x9090 (son dos *nop*) para sustituir esos dos bytes (*74d1*) y que no haga nada:

```
[0x7f9d8b2df100]> s 0x55895559148b
[0x55895559148b]> wx 9090
```

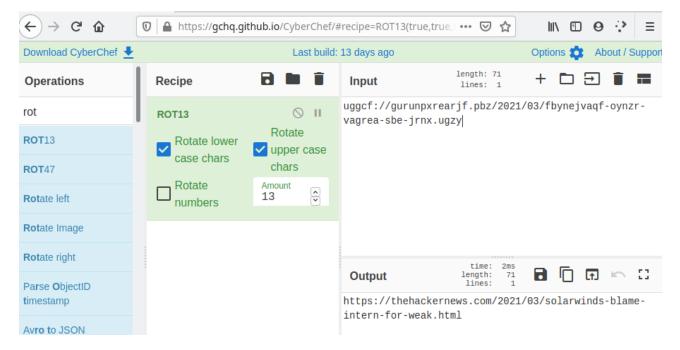
Comprobamos que se ha reemplazado:

```
[0x55895559148b]> pdf @ main
  327: int
                     (int argc, char **argv, char **envp);
                                            @ rbp-0x58
                                            @ rbp-0x54
                  var
                  var
               ; var int64_t var_30h @ rbp-0x30; var int64_t var_28h @ rbp-0x28; var int64_t var_20h @ rbp-0x20; var int64_t var_18h @ rbp-0x18; var int64_t var_8h @ rbp-0x18
                ; var int64
                                          f30f1efa
                                                              endbr64
                                          4889e5
                                                              mov rbp, rsp
                                          4883ec60
                                                              sub rsp, 0x60
                                          64488b042528.
                                                              mov rax, qword fs:[0x28]
                                                              mov qword [var_8h], rax
                                          488945f8
                                          31c0
                                                              mov dword [var_58h], 0
                                          c745a80000000.
                                          837da800
                                                              cmp dword [var_58h], 0
                                          90
                                          488d3d800b00. lea rdi, str.Good_here_s_your_hint:
```

Ejecutamos y ya nos mostrará la pista:

```
[0x55895559148b]> dc
Good, here's your hint: uggcf://gurunpxrearjf.pbz/2021/03/fbynejvaqf-oynzr-vagrea-sbe-jrnx.ugzy
What's the password?:
```

Se puede ver facilmente que es una url codificada con rotación (es rotación 13):



## Entramos al enlace:



Y se puede ver bastante claro qué contraseña va a ser, probamos "solarwinds123":

```
[0x55895559148b]> dc
Good, here's your hint: uggcf://gurunpxrearjf.pbz/2021/03/fbynejvaqf-oynzr-vagrea-sbe-jrnx.ugzy
What's the password?: solarwinds123
Oh, well... Here's_your flag: moonCTF{0661de560d0fbe73ecf5d894e24f0982}
```

Y ya nos devuelve el flag.