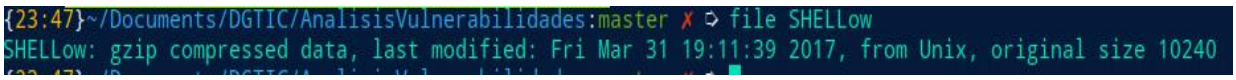
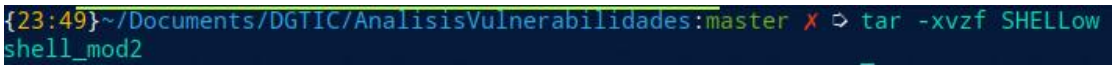
Hernández González Ricardo Omar
Examen

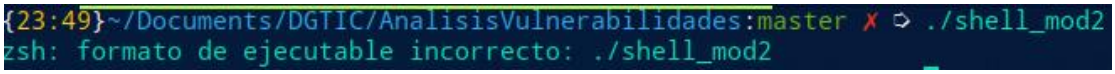
Lo primero que haremos será hacerle un **file** al archivo descargado para saber de qué se trata:



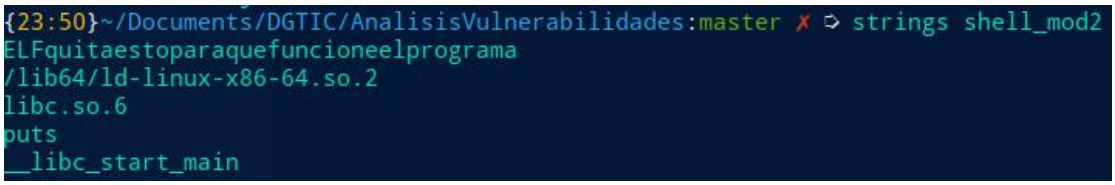
Podemos observar que es un comprimido, por lo cual lo descomprimimos con **tar**:



Observamos que nos descomprime un archivo, lo ejecutamos para ver que pasa:

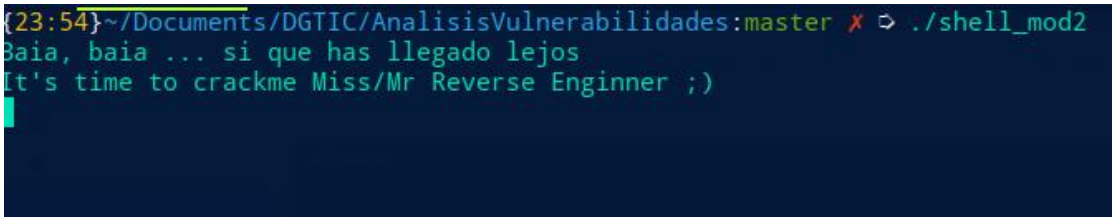


No se puede ejecutar, por lo que procedemos a ver que contiene internamente. Al hacerle un **strings** vemos que tiene una parte de más:

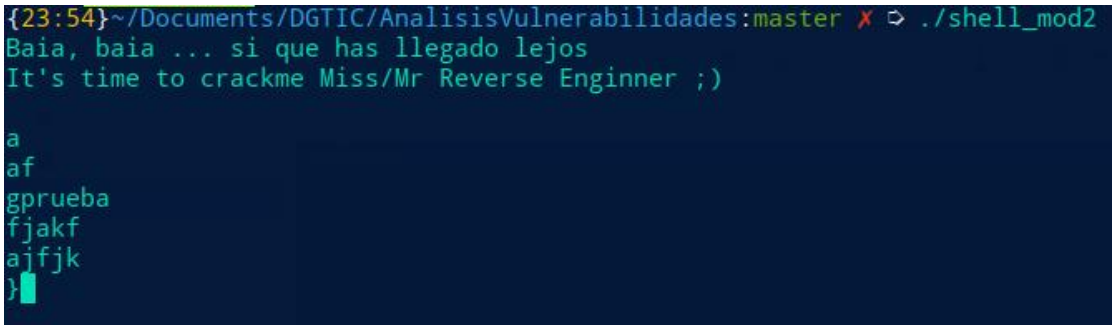


Por lo que podemos deducir que al ejecutable se le agregaron cosas, se las quitamos, esto lo podemos hacer de varias formas, en este caso lo haremos con un editor:

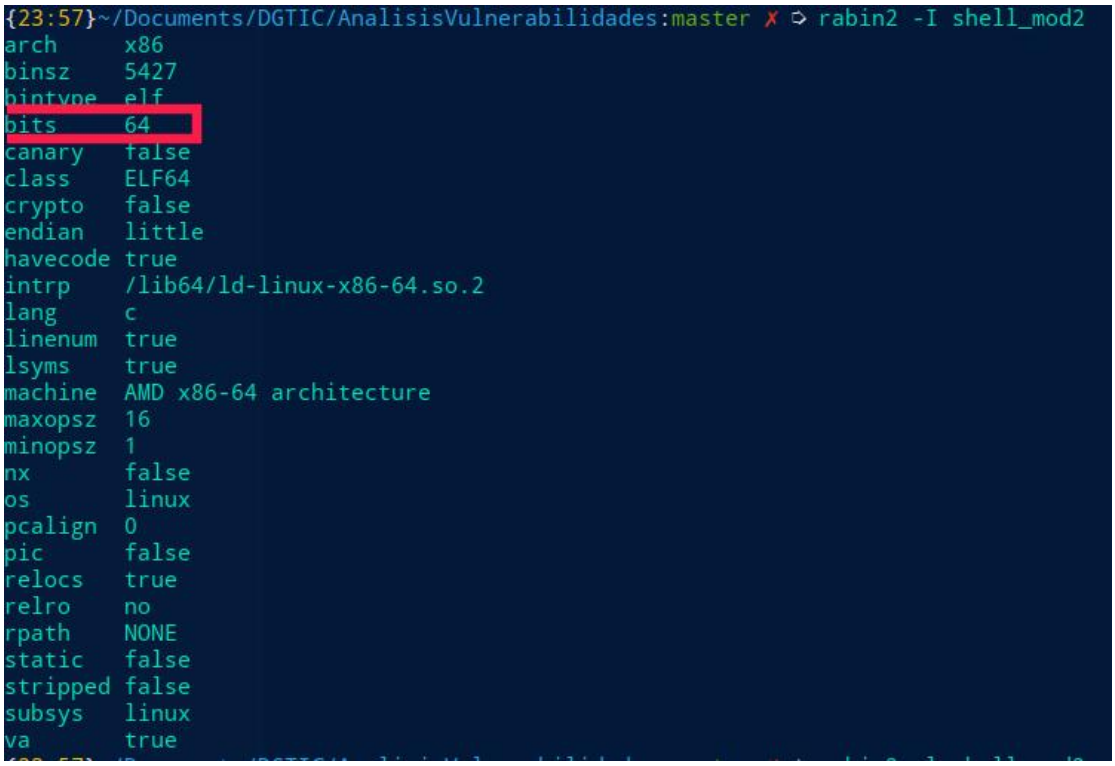
Lo guardamos y lo volvemos a ejecutar:



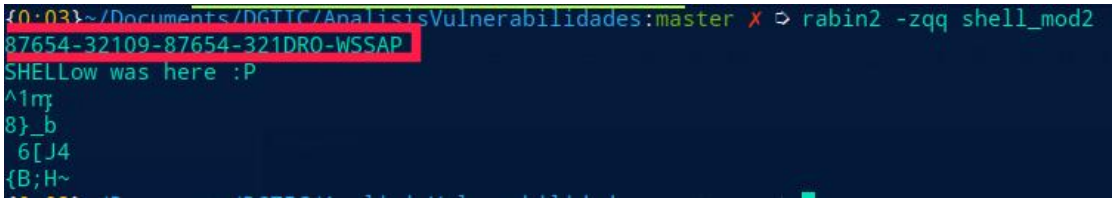
Ahora podemos observar que ya se ejecuta pero que está a la espera de algo, por lo que probamos en escribir texto y vemos que pasa:



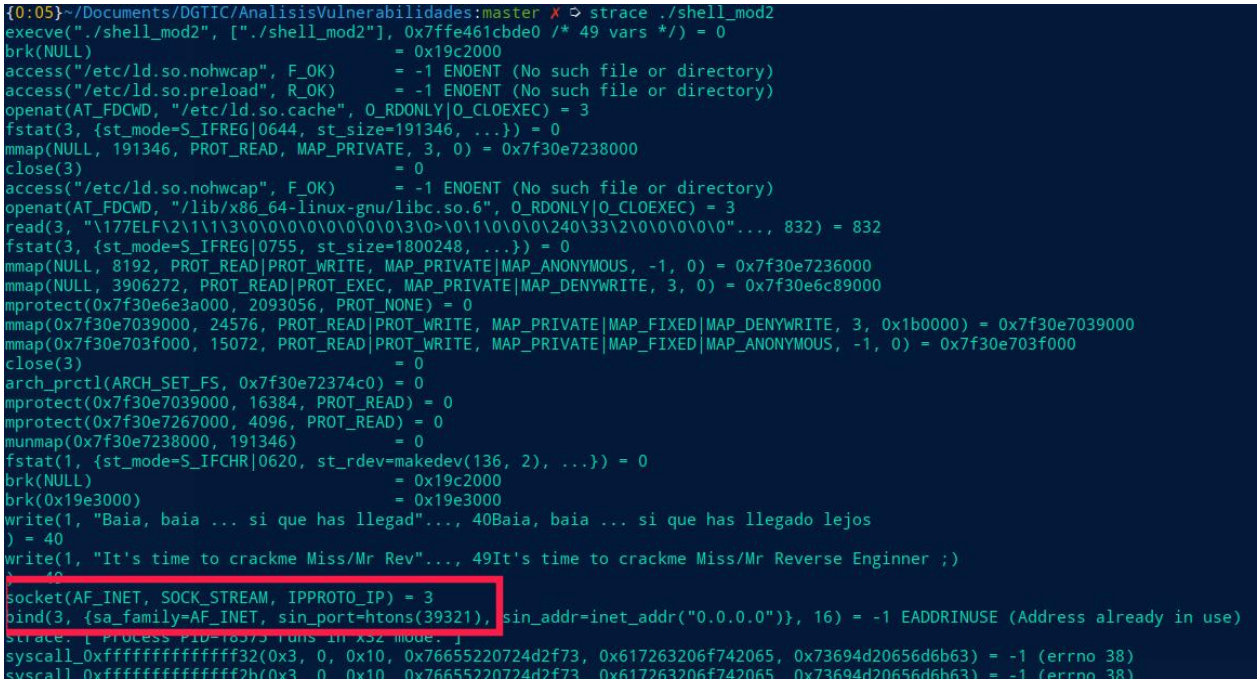
Entonces podemos pensar que la onda va por otro lado. Procedemos a ver la información del archivo con **rabin2**:



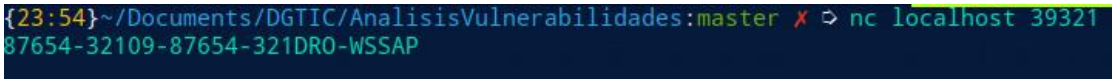
Donde solo tendremos que estamos tratando con mnemónicos de 64 bits. Después veremos sus cadenas:



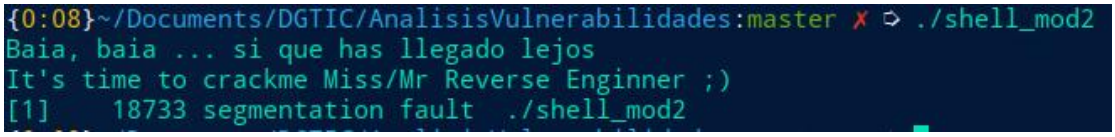
Podemos ver que se trata de algún tipo de serial, por lo cual nos da más información de qué tenemos que hacer, ahora solo falta el saber cómo está compuesto el serial y como se lo pasamos. Posteriormente hacemos el analisis con **strace:**



Con esto ya nos damos cuenta que se trata de un socket y que el programa está a la espera a una conexión en el puerto 39321, por lo que probamos conectarnos con **netcat** para ver que pasa:

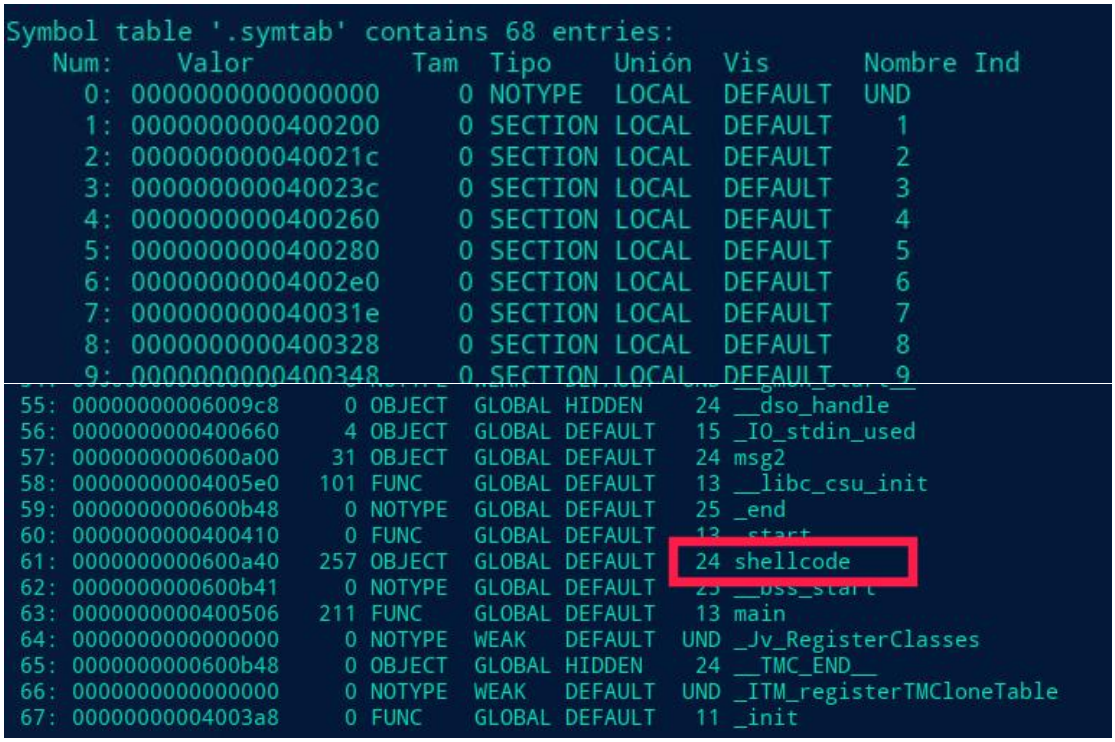


Tratamos de pasarle el serial que encontramos anteriormente, pero observamos que en donde se ejecuta se muere el programa y nos marca un segmentation fault, por lo que tenemos que seguir buscando más información.

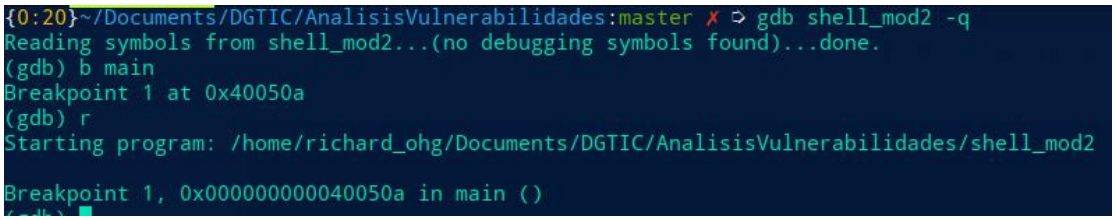


Después a esto, procedemos a usar **radelf**, con la opción **-a** para ver todo, esto nos arroja la siguiente información:

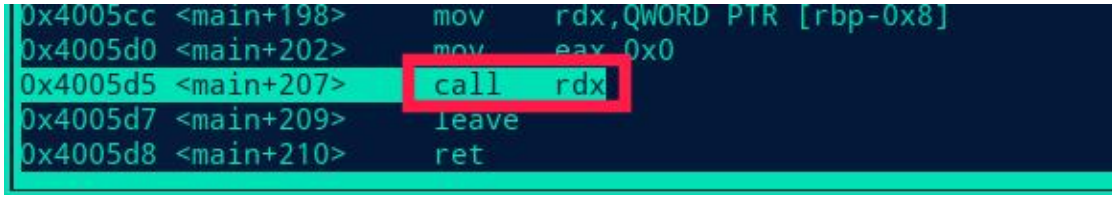
Más abajo, en la tabla de símbolos, observamos que hay una parte del código que se llama **shellcode**, esto también fue encontrado cuando hicimos **strings**, pero aquí encontramos más información acerca de ellos, vemos que es un objeto y por el nombre suponemos hace algo que queremos obtener con el serial.

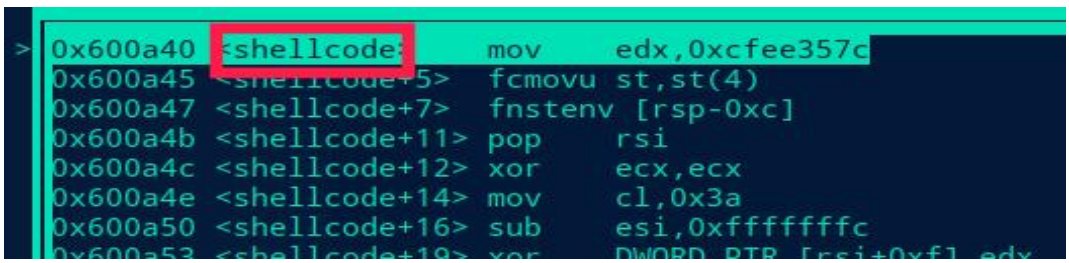


Por último, después de toda la información obtenida anteriormente, procederemos a debugear el programa, para esto usaremos gdb, ponemos un break en el main y lo ejecutamos:



Ponemos el **layout asm** y **layout regs** para ver los registros y las instrucciones, ahí observaremos que llega al punto donde hace un call a un registro y se pasa a la zona del shellcode:



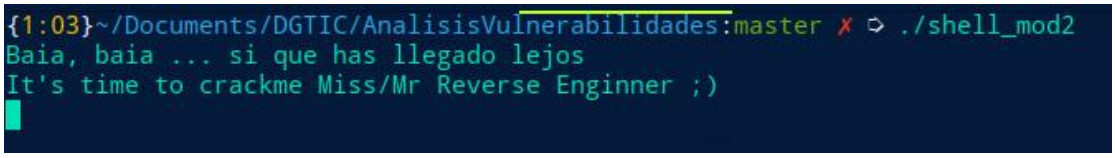


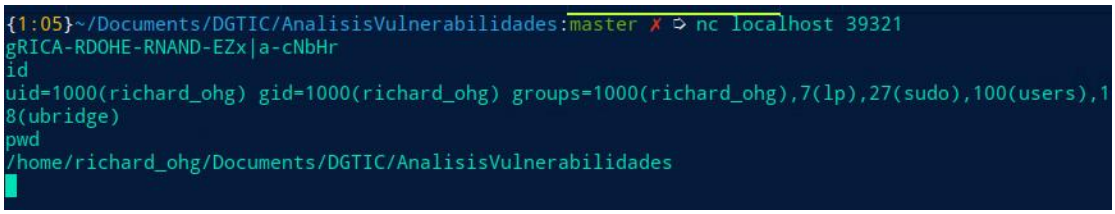
Ahí es donde veremos las comparaciones que necesitamos para el serial. Podemos observar, mediante el análisis del código ensamblador, que el total se caracteres debe ser 29, que debe llevar entre cada 5 caracteres un guion medio '-' y por último, que la suma de todos los caracteres debe ser 2272.

Conclusión:

Con esto, podemos decir que el ejecutable abre un socket con puerto 39321. Que tiene que recibe la cadena desde el socekt. Que la cadena es un serial con caracteristicas específicas. El serial debe contener 29 caracteres, debe tener un guión entre cada 5 caracteres y la suma de dichos caracteres debe ser 2272.

Una entrada válida es esta:





y con esto obtendremos una shell.