## REVMEM

Lo scopo è indovinare l'input corretto. Ad esempio se facciamo ./revmem asdasd lui ci dirà che il flag è sbagliato.

**SOLUZIONE**

Apro il file con Ghidra.
Se cerco il main non c'è un main, quindi come trovo il main quando non c'è un main?
Cerco main e poi vado su _libc_start_main sotto a symbol tree (sopra la ricerca) poi nel codice (centrale) vedo chi sta chiamando _libc_start_main e per fare ciò faccio tasto destro sopra _libc_start_main (quello nel commento grigio THUNK FUNCTION) e poi references→show references to _libc_start_main e clicco sopra CALL qword…
Ora nel codice a destra ho la funzione _libc_start_main e clicco nel suo primo parametro (FUN_001011da) che sarà il main.
Ora ho il main: abbiamo _s1 che è il flag generato da una funzione (che possiamo vedere cliccandoci) che viene confrontato col mio input (param2) e 0x1 che è la lunghezza del flag.
_s1 viene calcolato e tenuto in memoria questo significa che in qualche punto della memoria c'è il flag. Quindi posso runnare il binary, break allo string compare e poi guardare ai parametri della string compare perché uno dei due è il flag.
Possiamo fare gdb ./revmem e poi fare b strncmp e infine fare run asdasd (basta mettere qualche input maggiore di 2 → deve essere maggiore di 2 perché lo richiede nel codice) e dopo aver fatto un po' di si (comando di gdb) vedo il flag in chiaro su gdb sia nello stack ma anche nei registri.

**SECONDA SOLUZIONE (PIÚ RAPIDA)**

C'è un comando che ci permette di trace all the library call che è ltrace.
Quindi posso fare ltrace ./revmem asdasd (va bene qualsiasi input basta che sia maggiore di 2) e vediamo subito il flag.
Il comando **ltrace** in Linux è utilizzato per intercettare e tracciare le chiamate alle funzioni di libreria dinamica effettuate da un programma in esecuzione. In sostanza, **ltrace** consente di vedere quali funzioni di libreria dinamica vengono chiamate da un'applicazione e con quali argomenti.