Soluzione

Viene dato in particolare un file *sumiti* binario senza estensione, come al solito da analizzare senza eseguire nessuna patch. L'esecuzione chiede semplicemente di inserire un nome e poi termina.
Controllando il file con *radare2*, possiamo notare diverse cose:

- decrypt, che esegue un loop e una XOR tra variabili in memoria per decriptare la flag
- create_panino, che chiama time, srand e controlla lo stack (canary), ritornando una divisione per 9999
- check_sandwitch, che controlla una serie i caratteri in successione, affinché siano almeno 8 e compaia la stringa F4nTa151A

In IDA si vede subito che l'input che vogliamo è SumitiLover1234:

```
mov     eax, 0
call    print_intro
lea     rdi, aWeHaveManyCust ; "We have many customers these days... Di"...
mov     eax, 0
call    _printf
lea     rax, [rbp+s1]
mov     rsi, rax
lea     rdi, aS          ; "%s"
mov     eax, 0
call    ___isoc99_scanf
lea     rax, [rbp+s1]
lea     rsi, s2          ; "SumitiLover1234"
mov     rdi, rax         ; s1
```

continuando l'esecuzione viene eseguita la funzione *check_sandwitch* che accetta come input una stringa lunga 8 e controlla che per ogni carattere della stringa corrisponde al codice ascii.
Utilizzando in alternativa a prima la funzione chr di python e riordinando è possibile ottenere
la stringa richiesta
- print(chr(70),chr(52),chr(110),chr(84),chr(97),chr(53),chr(49),chr(65))
2) input : F4nTa51A

Il file è un eseguibile PIE, pertanto è in atto una rilocazione dinamica degli indirizzi.

Viene richiesto un pin casuale che poi viene eseguito in XOR e diviso come detto sopra, usando gdb è possibile creare un breakpoint al momento dell'assegnazione alla variabile panino
mov [rbp+var_44], eax

Si deve eseguire e fallire il programma almeno una volta. Poi avremo gli indirizzo con 0x0000555555555.
Metteremo un break subito dopo la *create_panino*, come segue:

```
   0x000055555555567e <+223>:   call   0x5555555554ae <create_panino>
   0x0000555555555683 <+228>:   mov    DWORD PTR [rbp-0x44],eax
   0x0000555555555686 <+231>:   lea    rdi,[rip+0xf53]        # 0x555555
```

Sul terminale quindi si inseriscano, le seguenti istruzioni (considerando di stampare "eax" perché abbiamo fatto la "mov"):

gdb sumiti
b 0x0000555555555683*
r
SumitiLover1234
F4nTa51A
print $eax
convertire da hex in decimale *(nel mio caso è stato 0x3a0 in decimale 0928)*
c
Inserire il pin
quindi ottenere la flag : SPRITZ{TwO_EuRo_PleAs3}

```
Legend: code, data, rodata, value

Breakpoint 1, 0x0000555555555683 in main ()
gdb-peda$ print $eax
$3 = 0x3a0
gdb-peda$ c
Continuing.
Your panino is ready, but if you want it, guess IN ORDER the FOUR ingred
ients he used, writing them as a 4 digit number (e.g., 0123): 0928
Oh you did it!!! Here your wonderful panino:
SPRITZ{TwO_EuRo_PleAs3}
[Inferior 1 (process 8068) exited normally]
Warning: not running
gdb-peda$
```