

# Linux\_asm\_jenya writeup

#We use radare2 for finding the password

r2 -A ./main

#there is only a enter password string which we check using iz

#we disass entry0

afl

pdf @ entry0

[0x00401000]> pdf @ entry0

;;-- section..text:

;;-- segment.LOAD1:

;;-- rip:

└ 161: entry0 ();

| 0x00401000 48be322040.. movabs rsi,  
str.Enter\_password:\_Correct\_nWrong\_n ; 0x402032 ; "Enter password:  
Correct\nWrong\n" ; [01] -r-x section size 212 named .text

| 0x0040100a bb10000000 mov ebx, 0x10 ; 16

| 0x0040100f e8b0000000 call fcn.004010c4

| 0x00401014 b800000000 mov eax, 0

| 0x00401019 bf00000000 mov edi, 0

| 0x0040101e 48be002040.. movabs rsi, segment.LOAD2 ; 0x402000

| 0x00401028 ba32000000 mov edx, 0x32 ; '2' ; 50

| 0x0040102d 0f05 syscall

| 0x0040102f e836000000 call 0x40106a

| 0x00401034 e868000000 call fcn.004010a1

```

|      0x00401039    48be422040.. movabs rsi, 0x402042    ; 'B @' ;
"Correct\nWrong\n"
|      0x00401043    bb08000000    mov ebx, 8
|      └─< 0x00401048    eb0f        jmp 0x401059
|      | ; CODE XREF from entry0 @ 0x401096(x)
|      | ; CODE XREF from fcn.004010a1 @ 0x4010c1(x)
|      └─> 0x0040104a    48be4a2040.. movabs rsi, 0x40204a    ; 'J @' ;
"Wrong\n"
|      : | 0x00401054    bb06000000    mov ebx, 6
|      : | ; CODE XREF from entry0 @ 0x401048(x)
|      : └─> 0x00401059    e866000000    call fcn.004010c4
|      : 0x0040105e    b83c000000    mov eax, 0x3c        ; '<' ; 60
|      : 0x00401063    bf00000000    mov edi, 0
|      : 0x00401068    0f05        syscall
|      : ; CALL XREF from entry0 @ 0x40102f(x)
|      : 0x0040106a    48bf002040.. movabs rdi, segment.LOAD2 ;
0x402000
|      : ; CODE XREF from entry0 @ 0x40107c(x)
|      : └─> 0x00401074    803f0a        cmp byte [rdi], 0xa
|      └─< 0x00401077    7405        je 0x40107e
|      | : 0x00401079    48ffc7        inc rdi
|      | : └─< 0x0040107c    ebf6        jmp 0x401074
|      | : ; CODE XREF from entry0 @ 0x401077(x)
|      └─> 0x0040107e    48ffcf        dec rdi
|      : 0x00401081    48be002040.. movabs rsi, segment.LOAD2 ;
0x402000
|      : ; CODE XREF from entry0 @ 0x40109e(x)
|      : └─> 0x0040108b    4839f7        cmp rdi, rsi
|      └─< 0x0040108e    7e10        jle 0x4010a0
|      | : 0x00401090    8a07        mov al, byte [rdi]
|      | : 0x00401092    8a26        mov ah, byte [rsi]
|      | : 0x00401094    38e0        cmp al, ah

```

```

| | |< 0x00401096 75b2 jne 0x40104a
| | | 0x00401098 48ffcf dec rdi
| | | 0x0040109b 48ffc6 inc rsi
| | |< 0x0040109e ebeb jmp 0x40108b
| | ; CODE XREF from entry0 @ 0x40108e(x)
L |> 0x004010a0 c3 ret

```

The code compares our input if it is a palindrome or not .

If it is a palindrome like test ,101,racecar the program prints correct

if it dosen't it prints wrong