MIxto de strings

Al ejecutar el binario, se observa que no imprime la flag, lo que indica que el reto debe resolverse mediante análisis estático.

```
└─$ chmod +x reverse_strings

┌──(vfric㉿vfric)-[~/…/CTF-cyberminds-
└─$ ./reverse_strings
Bienvenido.
Este programa NO mostrará la flag.
Analiza el binario para encontrarla.
```

Se analiza la función main usando radare2:

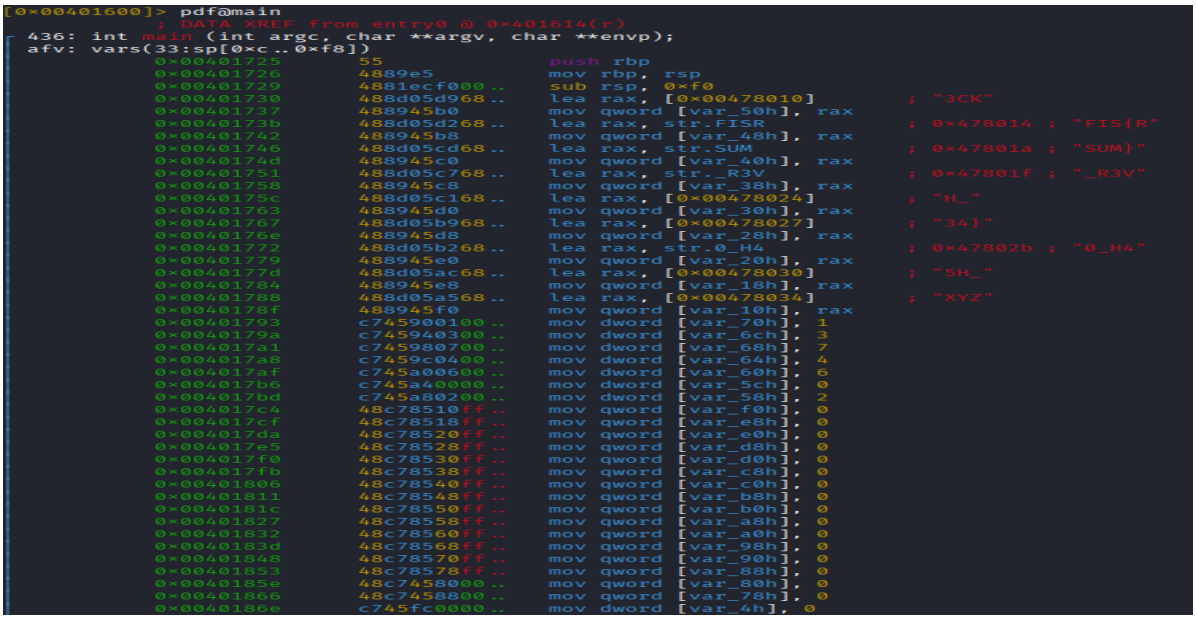
r2 reverse_strings

```
[0x00401600]> pdf@main
            ; DATA XREF from entry0 @ 0x401614(r)
┌ 436: int main (int argc, char **argv, char **envp);
│ afv: vars(33:sp[0xc..0xf8])
│           0x00401725      55              push rbp
│           0x00401726      4889e5          mov rbp, rsp
│           0x00401729      4881ecf000..    sub rsp, 0xf0
│           0x00401730      488d05d968..    lea rax, [0x00478010]       ; "3CK"
│           0x00401737      488945b0        mov qword [var_50h], rax
│           0x0040173b      488d05d268..    lea rax, str.FISR           ; 0x478014 ; "FIS{R"
│           0x00401742      488945b8        mov qword [var_48h], rax
│           0x00401746      488d05cd68..    lea rax, str.SUM            ; 0x47801a ; "SUM}"
│           0x0040174d      488945c0        mov qword [var_40h], rax
│           0x00401751      488d05c768..    lea rax, str._R3V           ; 0x47801f ; "_R3V"
│           0x00401758      488945c8        mov qword [var_38h], rax
│           0x0040175c      488d05c168..    lea rax, [0x00478024]       ; "H_"
│           0x00401763      488945d0        mov qword [var_30h], rax
│           0x00401767      488d05b968..    lea rax, [0x00478027]       ; "34}"
│           0x0040176e      488945d8        mov qword [var_28h], rax
│           0x00401772      488d05b268..    lea rax, str.0_H4           ; 0x47802b ; "0_H4"
│           0x00401779      488945e0        mov qword [var_20h], rax
│           0x0040177d      488d05ac68..    lea rax, [0x00478030]       ; "5H_"
│           0x00401784      488945e8        mov qword [var_18h], rax
│           0x00401788      488d05a568..    lea rax, [0x00478034]       ; "XYZ"
│           0x0040178f      488945f0        mov qword [var_10h], rax
│           0x00401793      c745900100..    mov dword [var_70h], 1
│           0x0040179a      c745940300..    mov dword [var_6ch], 3
│           0x004017a1      c745980700..    mov dword [var_68h], 7
│           0x004017a8      c7459c0400..    mov dword [var_64h], 4
│           0x004017af      c745a00600..    mov dword [var_60h], 6
│           0x004017b6      c745a40000..    mov dword [var_5ch], 0
│           0x004017bd      c745a80200..    mov dword [var_58h], 2
│           0x004017c4      48c78510ff..    mov qword [var_f0h], 0
│           0x004017cf      48c78518ff..    mov qword [var_e8h], 0
│           0x004017da      48c78520ff..    mov qword [var_e0h], 0
│           0x004017e5      48c78528ff..    mov qword [var_d8h], 0
│           0x004017f0      48c78530ff..    mov qword [var_d0h], 0
│           0x004017fb      48c78538ff..    mov qword [var_c8h], 0
│           0x00401806      48c78540ff..    mov qword [var_c0h], 0
│           0x00401811      48c78548ff..    mov qword [var_b8h], 0
│           0x0040181c      48c78550ff..    mov qword [var_b0h], 0
│           0x00401827      48c78558ff..    mov qword [var_a8h], 0
│           0x00401832      48c78560ff..    mov qword [var_a0h], 0
│           0x0040183d      48c78568ff..    mov qword [var_98h], 0
│           0x00401848      48c78570ff..    mov qword [var_90h], 0
│           0x00401853      48c78578ff..    mov qword [var_88h], 0
│           0x0040185e      48c7458000..    mov qword [var_80h], 0
│           0x00401866      48c7458800..    mov qword [var_78h], 0
│           0x0040186e      c745fc0000..    mov dword [var_4h], 0
```

En el desensamblado se observa claramente que el programa carga múltiples cadenas constantes en memoria usando instrucciones lea, las cuales corresponden a fragmentos de la flag y a strings falsos.

Esto confirma que la flag está fragmentada en múltiples strings, junto con cadenas que no forman parte de la solución.

Más adelante en main se observa la inicialización de varios enteros consecutivos en memoria.



```
[0x00401600]> pdf@main
            ; DATA XREF from entry0 @ 0x401614(r)
┌ 436: int main (int argc, char **argv, char **envp);
afv: vars(33:sp[0xc..0xf8])
           0x00401725      55              push rbp
           0x00401726      4889e5          mov rbp, rsp
           0x00401729      4881ecf000..    sub rsp, 0xf0
           0x00401730      488d05d968..    lea rax, [0x00478010]       ; "3CK"
           0x00401737      488945b0        mov qword [var_50h], rax
           0x0040173b      488d05d268..    lea rax, str.FISR           ; 0x478014 ; "FIS{R"
           0x00401742      488945b8        mov qword [var_48h], rax
           0x00401746      488d05cd68..    lea rax, str.SUM            ; 0x47801a ; "SUM}"
           0x0040174d      488945c0        mov qword [var_40h], rax
           0x00401751      488d05c768..    lea rax, str._R3V           ; 0x47801f ; "_R3V"
           0x00401758      488945c8        mov qword [var_38h], rax
           0x0040175c      488d05c168..    lea rax, [0x00478024]       ; "H_"
           0x00401763      488945d0        mov qword [var_30h], rax
           0x00401767      488d05b968..    lea rax, [0x00478027]       ; "34}"
           0x0040176e      488945d8        mov qword [var_28h], rax
           0x00401772      488d05b268..    lea rax, str.0_H4           ; 0x47802b ; "0_H4"
           0x00401779      488945e0        mov qword [var_20h], rax
           0x0040177d      488d05ac68..    lea rax, [0x00478030]       ; "5H_"
           0x00401784      488945e8        mov qword [var_18h], rax
           0x00401788      488d05a568..    lea rax, [0x00478034]       ; "XYZ"
           0x0040178f      488945f0        mov qword [var_10h], rax
           0x00401793      c745900100..    mov dword [var_70h], 1
           0x0040179a      c745940300..    mov dword [var_6ch], 3
           0x004017a1      c745980700..    mov dword [var_68h], 7
           0x004017a8      c7459c0400..    mov dword [var_64h], 4
           0x004017af      c745a00600..    mov dword [var_60h], 6
           0x004017b6      c745a40000..    mov dword [var_5ch], 0
           0x004017bd      c745a80200..    mov dword [var_58h], 2
           0x004017c4      48c78510ff..    mov qword [var_f0h], 0
           0x004017cf      48c78518ff..    mov qword [var_e8h], 0
           0x004017da      48c78520ff..    mov qword [var_e0h], 0
           0x004017e5      48c78528ff..    mov qword [var_d8h], 0
           0x004017f0      48c78530ff..    mov qword [var_d0h], 0
           0x004017fb      48c78538ff..    mov qword [var_c8h], 0
           0x00401806      48c78540ff..    mov qword [var_c0h], 0
           0x00401811      48c78548ff..    mov qword [var_b8h], 0
           0x0040181c      48c78550ff..    mov qword [var_b0h], 0
           0x00401827      48c78558ff..    mov qword [var_a8h], 0
           0x00401832      48c78560ff..    mov qword [var_a0h], 0
           0x0040183d      48c78568ff..    mov qword [var_98h], 0
           0x00401848      48c78570ff..    mov qword [var_90h], 0
           0x00401853      48c78578ff..    mov qword [var_88h], 0
           0x0040185e      48c7458000..    mov qword [var_80h], 0
           0x00401866      48c7458800..    mov qword [var_78h], 0
           0x0040186e      c745fc0000..    mov dword [var_4h], 0
```

Estos valores representan el orden de acceso a los fragmentos, es decir:

order = {1, 3, 7, 4, 6, 0, 2};

El programa inicializa además un buffer vacío (varios mov qword ..., 0), lo cual indica que posteriormente se realiza una concatenación progresiva de strings (típicamente con strcat).

Usando el orden identificado y los fragmentos observados, se reconstruye la flag concatenando:

1: FIS{R
3: _R3V
7: 5H_
4: H_
6: 0_H4
0: 3CK
2: SUM}

Concatenación final:

FIS{R_R3V5H_H_0_H43CKSUM}