

[REVERSE]

ESREV Writeup

1 ère façon PRO

Binaire :

The logo for 'ESREV' is displayed in a bright green, monospace-style font against a dark, rectangular background.

```
file ESREV
```

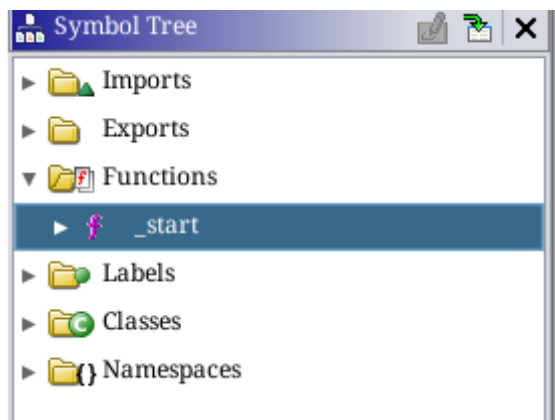
```
[eazy@archlinux Desktop]$ file ESREV
ESREV: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, not stripped
```

```
strings ESREV
```

```
[eazy@archlinux Desktop]$ strings ESREV
ESREV.asm
_Eaaaazy
__bss_start
_edata
_end
.symtab
.strtab
.shstrtab
.text
.data:
```

```
ghidra
```

Selectionner le point de lancement du programme :



Code :

```

                                _start
00401000 48 ba 46 20      MOV      RDX,c1
                                40 00 00 00
                                00 00
0040100a 48 81 fa 4a      CMP      RDX,c2
                                20 40 00
00401011 74 07            JZ       GXM
00401013 ba 00 00 00      MOV      EDX,0x0
                                00
00401018 eb 27            JMP      XGM

                                GXM
0040101a b8 01 00 00      MOV      EAX,0x1
                                00
0040101f bf 01 00 00 00    MOV      EDI,0x1
00401024 48 be 00 20      MOV      RSI,f
                                40 00 00 00
                                00 00
0040102e ba 2c 00 00      MOV      EDX,0x2c
                                00
00401033 0f 05            SYSCALL
00401035 b8 3c 00 00      MOV      EAX,0x3c
                                00
0040103a bf 00 00 00 00    MOV      EDI,0x0
0040103f 0f 05            SYSCALL

                                XGM
00401041 b8 01 00 00      MOV      EAX,0x1
                                00
00401046 bf 01 00 00 00    MOV      EDI,0x1
0040104b 48 be 2c 20      MOV      RSI,X
                                40 00 00 00
                                00 00
00401055 ba 1a 00 00      MOV      EDX,0x1a

```

PATCH l'instruction JZ :

00401000

74 07 40 00 00 00

MOV

RDX,c2

0040100a

48 81 fa 4a 20 40 00

CMP

RDX,c2

= 08h

00401011

74 07

JZ

GXM

00401013

ba 00 00 00 00

MOV

EDX,0x0

00401018

eb 27

JMP

XGM

0040101a

b8 01 00 00 00

MOV

EAX,0x1

0040101f

bf 01 00 00 00

MOV

EDI,0x1

00401024

48 be 00 20 40 00 00 00

MOV

RSI,f

0040102e

ba 2c 00 00 00

MOV

EDX,0x2c

00401033

0f 05

SYSCALL

00401035

b8 3c 00 00 00

MOV

EAX,0x3c

0040103a

bf 00 00 00 00

MOV

EDI,0x0

0040103f

0f 05

SYSCALL

Bookmark...

Ctrl+D

Clear Code Bytes

C

Clear With Options

Clear Flow and Repair

Copy

Ctrl+C

Copy Special...

Paste

Ctrl+V

Comments

Instruction Info...

Modify Instruction Flow...

Patch Instruction

Ctrl+Shift+G

Processor Manual...

Processor Options...

Changer l'instruction JZ (JUMP SI EGALE) par l'instruction JMP (JUMP) :

00 00

0040100a

48 81 fa 4a 20 40 00

CMP

RDX,c2

00401011

74 07

JMP

0x0040101a

00401013

ba 00 00 00 00

JMP

00401018

eb 27

JMPF

Patch sur l'adresse 00401011 avec l'instruction JMP pour accéder à la partie de la mémoire réserver à GXM car par défaut le programme accède

à XGM :

	undefined	undefined __stdcall _start(void)	
		AL:1	<RETURN>
		_start	
00401000	48 ba 46 20 40 00 00 00 00 00	MOV	RDX,c1
0040100a	48 81 fa 4a 20 40 00	CMP	RDX,c2
00401011	eb 07	JMP	GXM
00401013	ba 00 00 00 00	MOV	EDX,0x0
00401018	eb 27	JMP	XGM
		GXM	
0040101a	b8 01 00 00 00	MOV	EAX,0x1
0040101f	bf 01 00 00 00	MOV	EDI,0x1
00401024	48 be 00 20 40 00 00 00 00 00	MOV	RSI,f

- Exporter le fichier patché en format ELF

```
[eazy@archlinux Desktop]$ file * | grep ESREV
ESREV:      ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, not stripped
ESREV-CRACKED: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), statically linked, not stripped
```

- Executer la version crack :

```
[eazy@archlinux Desktop]$ ./ESREV-CRACKED
AS53MBLi3D-PR0GR4M-U_U[eazy@archlinux Desk
OSINT intro
```

flag :

```
AS53MBLi3D-PR0GR4M-U_U
```

2 ème façon

Ghidra :

↑

00402000	41	??	41h	A
00402001	01	??	01h	

|

|

v

f18

00402022	4d	??	4Dh	M
00402023	01	??	01h	

hexa flag = 0x41, 0x53, 0x35, 0x33, 0x4D, 0x42, 0x4C, 0x69, 0x33, 0x44, 0x2D, 0x50, 0x52, 0x30, 0x47, 0x52, 0x34, 0x4D, 0x2D, 0x55, 0x5F, 0x55

0x41 0x53 0x35 0x33 0x4D 0x42 0x4C 0x69 0x33 0x44 0x2D 0x50 0x52 0x30 0x47 0x52 0x34 0x4D 0x2D 0x55 0x5F 0x55

Recipe

From Hex

DelimiterAuto

Input

0x41 0x53 0x35 0x33 0x4D 0x42 0x4C 0x69 0x33 0x44 0x2D 0x50 0x52 0x30 0x47 0x52 0x34 0x4D 0x2D 0x55 0x5F 0x55

Output

AS53MBLi3D-PR0GR4M-U_U

flag :

AS53MBLi3D-PR0GR4M-U_U