

Cifrado / Descifrado

Análisis de Vulnerabilidades



Pacheco Franco Jesús Enrique

jesus.pacheco@bec.seguridad.unam.mx

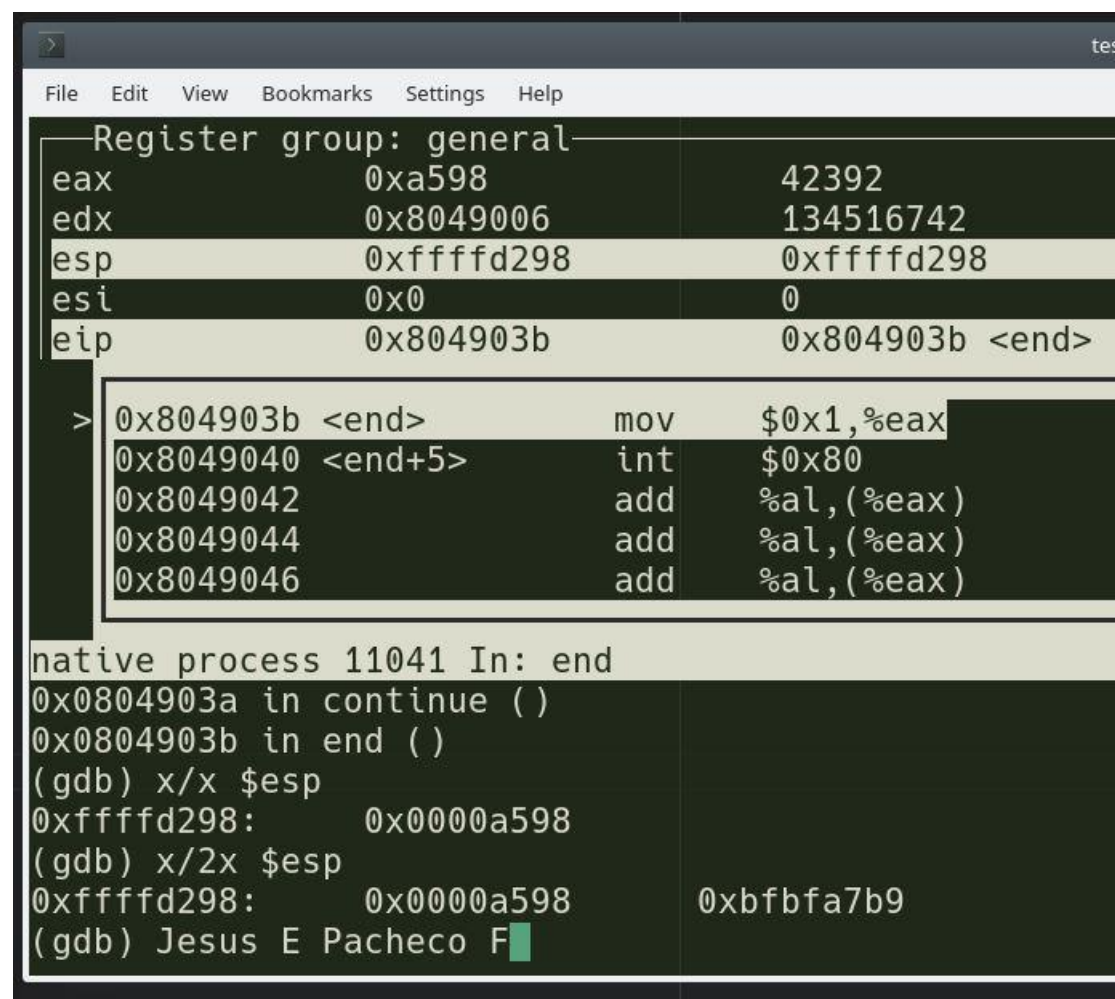
07/Abril/2019

Cadena de Entrada Cifrado

```
.data
string db "GG_Man"
str_len equ $ - string
```

La cadena de entrada de prueba fue “GG_Man” la cual corresponde con los siguientes caracteres ASCII “0x47 0x47 0x5f 0x4d 0x61 0x6e”.

Al terminar el proceso de cifrado en la pila se encuentra lo siguiente:



The screenshot shows a GDB debugger window with the following content:

```
File Edit View Bookmarks Settings Help

Register group: general
eax      0xa598      42392
edx      0x8049006   134516742
esp      0xffffd298 0xffffd298
esi      0x0        0
eip      0x804903b   0x804903b <end>

> 0x804903b <end>      mov    $0x1,%eax
0x8049040 <end+5>      int    $0x80
0x8049042              add    %al,(%eax)
0x8049044              add    %al,(%eax)
0x8049046              add    %al,(%eax)

native process 11041 In: end
0x0804903a in continue ()
0x0804903b in end ()
(gdb) x/x $esp
0xffffd298:      0x0000a598
(gdb) x/2x $esp
0xffffd298:      0x0000a598      0xbfbfa7b9
(gdb) Jesus E Pacheco F
```

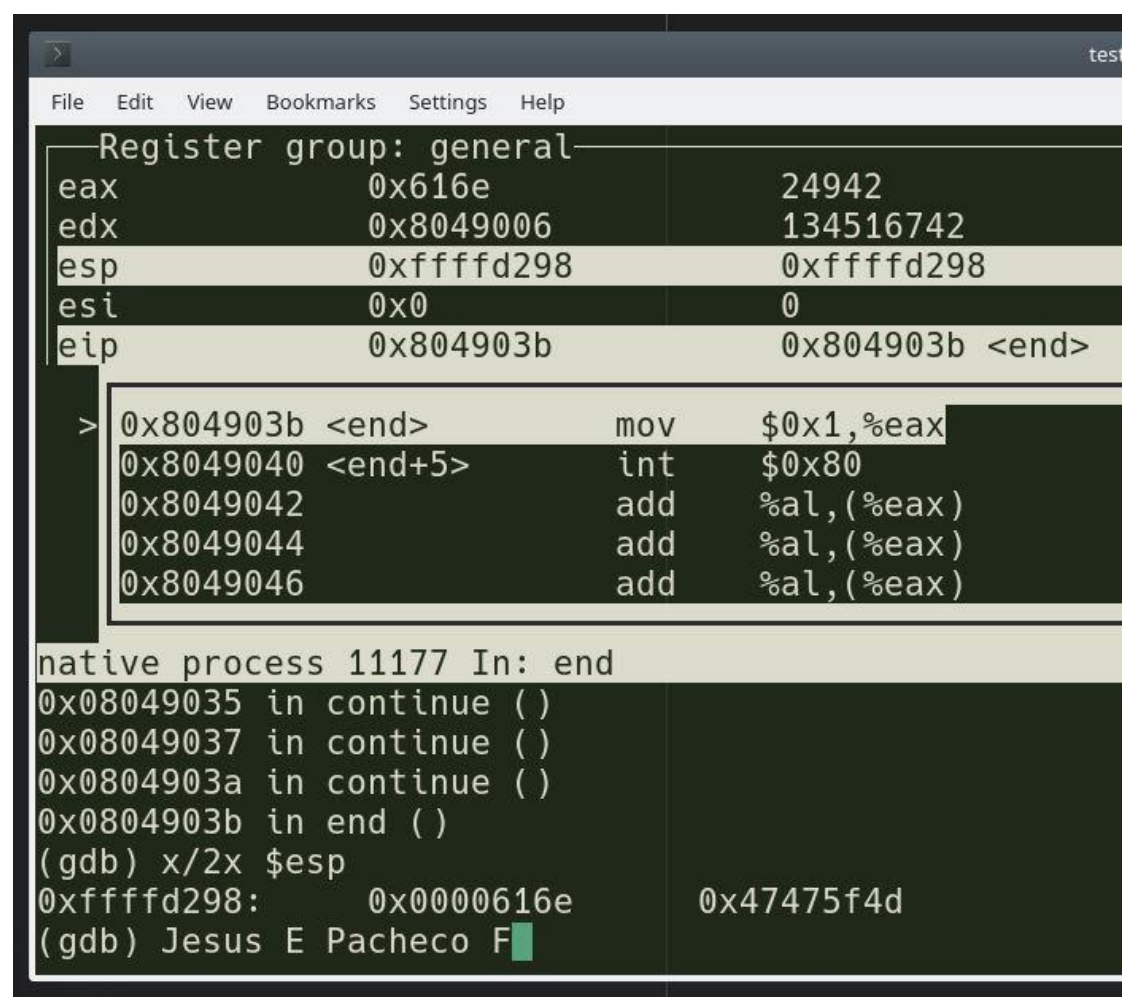
Observamos que la cadena resultante fue “0xbf 0xbf 0xa7 0xb9 0xa5 0x98”, este sería nuestro criptograma.

Cadena de Entrada Descifrado

```
.data
string db 0xbf, 0xbf, 0xa7, 0xb9, 0xa5, 0x98 ; Cadena a descifrar.
str_len equ $ - string ; Longitud cadena a cifrar.
```

Para descifrar colocamos la cadena que nos regresó el programa anterior.

Texto descifrado en la Pila



The screenshot shows a debugger window with the following content:

Register group: general

Register	Value	Comment
eax	0x616e	24942
edx	0x8049006	134516742
esp	0xffffd298	0xffffd298
esi	0x0	0
eip	0x804903b	0x804903b <end>

> 0x804903b <end> mov \$0x1,%eax
0x8049040 <end+5> int \$0x80
0x8049042 add %al,(%eax)
0x8049044 add %al,(%eax)
0x8049046 add %al,(%eax)

native process 11177 In: end

Address	Disassembly	Comment
0x08049035	in continue ()	
0x08049037	in continue ()	
0x0804903a	in continue ()	
0x0804903b	in end ()	
(gdb) x/2x \$esp		
0xffffd298:	0x0000616e	0x47475f4d
(gdb) Jesus E Pacheco F		

Como podemos observar en la pila tenemos la misma secuencia de caracteres hexadecimales que se ingreso en el programa de cifrado, entonces podemos concluir que el programa funciona correctamente.