Shellcode

In this task, the attacker wants to launch a Shellcode attack. The attacker has to completely avoid usage of zero in the script binary.

1) Why it is mandatory to avoid zeros in the attack binary? (1-2 lines max)

The second challenge of the attack is being able to find the address of the paylod used for the attack. This can be solved in two ways: the first uses the stack, while the second leverages the code region, after a `call` instruction. When the `call` instruction is executed, the address of the data is treated as the return address, and is pushed into the stack.

2) Please complete the missing parts (marked as “`***`“) of the code in `mysh.s` that involves the preparation of the code region.

3) What is the meaning of the line ”`xor eax, eax`”? Why is this line necessary?

4) In the line ”`mov [ebx+9], al`”, which other register could be used instead of ”`al`”? Why?

Instructions on basic stuff

1) Compiling to object code:

```
$ nasm –f elf32 mysh.s –o mysh.o
```

2) Generate the binary:

```
$ ld --omagic –m elf_i386 mysh.o –o mysh
```

3) Check current process ID of the shell

```
$ echo $$
```

4) Lunch the attack

```
$ mysh
```

5) Check the attack works well (the process ID must differ from the previous)

```
$ echo $$
```