Online 1: Buffer Overflow

Section: A1 Time: 35 minutes

You are given a file A1.c. Replace the <SZ> and <MAX> values according to the given rules. Craft a payload that, when **strcpy**-ed into `buf`,

- first executes `greet()`
- then immediately executes `get_shell()`

Don't change anything except the two macros.
Submit your exploit as `exploit.py`

Expected output:

```
Greetings challenger, 2005123!
CSE406# whoami
root
CSE406
```

```
SZ = 100 + <Last 3 digits of your ID> * 5
MAX = 200 + <Last 3 digits of your ID> * 5
```

Hint:
If function foo's address is `0xDEADBEEF`, then you can call foo using these instructions:

```
mov ebx, 0xDEADBEEF
call ebx
```

Use the following link to create assemble the instructions into raw hex:
https://defuse.ca/online-x86-assembler.htm#disassembly

If a function takes parameters, don't forget to push the values before calling that function.

Online 1: Buffer Overflow

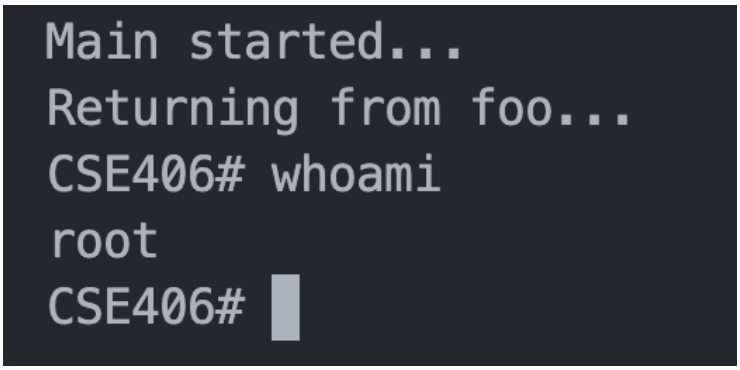
Section: B2 Time: 35 minutes

You are given a file B2.c containing some C code. Your task is to get access to the root shell by exploiting the stack buffer overflow attack.

Change the values **<param_1>** and **<param_2>** according to the given rules:

```
<param_1> = 100 + <Last 3 digits of your ID> * 5
<param_2> = 200 + <Last 3 digits of your ID> * 10
```

Expected output:



Submit your exploit.py file only.