# Raport pentru lucrarea 1: Scrierea shellcode

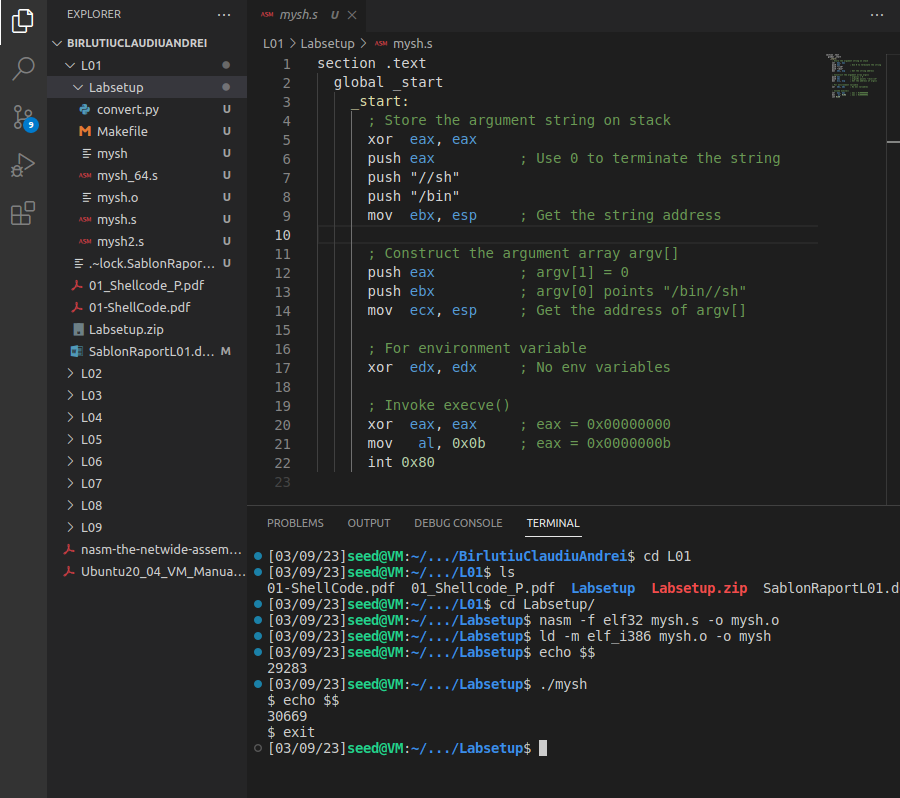
Autor: Birlutiu Claudiu-Andrei, gr 30643

1. ***Sarcina 1: Scrierea shellcode***

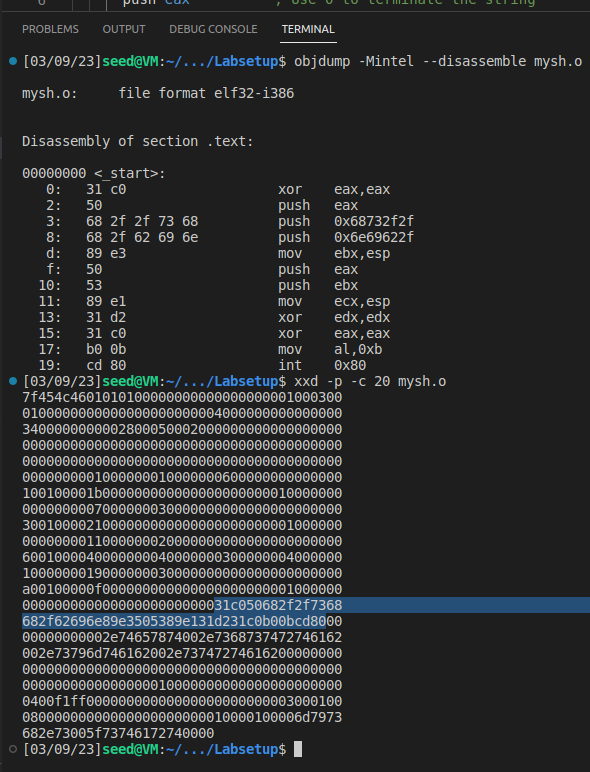
### Sarcina 1a. Procesul integral

Pentru realizarea acestei sarcini s-a făcut următorii pași:

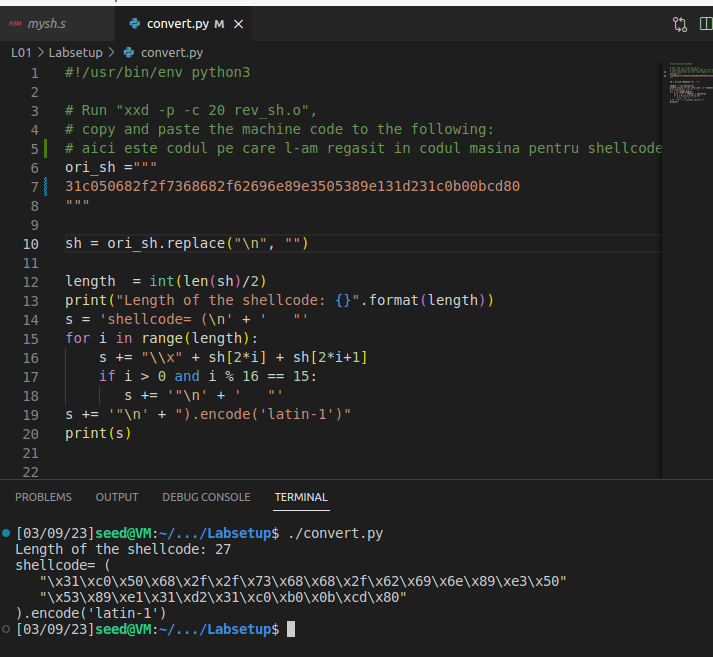
* s-a compilat codul assembly x86 pentru a obtine fișierul binar cu extensia .o ( fisier cu codul obiect, dar care nu are editate toate legăturile ) folosindu-ne de utilitarul **nasm**.
  + nasm -f elf32 mysh.s -o mysh.o
* se observa argumentele cu care s-a compilat codul: **elf32** ("Elf" este prescurtarea pentru Executable and Linkable Format (Format Executabil și Linkabil) și face specificare ca formatul binar așteptat este pe 32 de biti, **mysh.s** este numele fisierului cu codul de compilat iar codul obiect (binar) obținut are denumirea de **mysh.o.**
* pentru obtinerea **binarului executabil final** se folosește editorul de legături (**ld**), cu argumentele necesare, în special cu optiunea de a se lucra pe 32 de biti (x86). Se poate observa succesiunea comenzilor în imaginea de mai jos și de asemenea se observa executia programului prin lansarea unui shell nou (procese diferite la rularea comenzii ‘*echo $$*’) comenzile executate pentru a obtine fișierul executabil. (*ld -m elf\_i386 mysh.o -o mysh*)



* în continuare am dezasamblat codul obiect pentru a observa structura codului mașina ce reprezintă **shellcode**-ul; m-am folosit de utilitarul **objdumb,** cu optiunea pentru Intel (*objdump -Mintel --disassemble mysh.o*). De asemnea am rulat și comanda xxd pe codul obiect astfel încât sa observam continultu fisierului binar mysh.o generat în pasii anteriori (*xxd -p -c 20 mysh.o*)



* am copiat codul mașina generat pentru **shellcode** în fișierul convert.py pentru tiparirea codului python cu stringul ce reprezintă codul mașina pentru shellcode pentru a fi copiat în programe cu scop de atac. Se observa în următoarea imaginea rularea fisierului convert.py



* **Observație**: Am observat ca în urma generarii codului mașina pentru shellcode nu exista 0-uri ceea ce arata ca inserarea unui astfel de string în cadrul unui program este destul de puternica, deoarece nu are terminator ( \0 marcheaza finalul unui sir de caractere);

### Sarcina 1b. Eliminarea zerourilor din cod

### Sarcina 1c. Furnizarea argumentelor pentru apeluri sistem

### Sarcina 1d. Furnizarea variabilelor de mediu pentru execve()

## Sarcina 2: Folosirea segmentului de cod

### Explicația detaliată pentru fiecare rând al codului mysh2.sh

### Implementarea unui nou shellcode care tipărește cele două variabile de mediu

## Sarcina 3: Scrierea de shellcode pe 64 biți

### Repetarea sarcinii 1b pe shellcode pe 64 biți