

### Pergunta P1.1 - Buffer overflow em várias linguagens

```
Insira número: 1
Insira número: 1
Insira número: 1
Insira número: ^C
codigofonte$ 
CSI: ~/Aulas/1718-EngSeg/TPraticas/Aula10/codigofonte
Introduza número: 1
Introduza número: 1
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 10
    at L0verflow2.main(L0verflow2.java:18)
user@CSI:~/Aulas/1718-EngSeg/TPraticas/Aula10/codigofonte$ 1
bash: 1: command not found
user@CSI:~/Aulas/1718-EngSeg/TPraticas/Aula10/codigofonte$ 
Traceback (most recent call last):
  File "L0verflow2.py", line 5, in <module>
    tests[i]=test
IndexError: list assignment index out of range
user@CSI:~/Aulas/1718-EngSeg/TPraticas/Aula10/codigofonte$
```

Python e Java quebram a execução assim que detectam o acesso a memória não alocada pela variável, dando o python 1 célula de segurança.

O c apenas devolve segfault quando se acede a células de memória não mapeadas ou apenas de read only.

Neste caso como a variável i é reescrita, o ciclo torna-se infinito e as células de memória escritas são restritas ao buffer e à própria variável i.

Nota: ver 12 1,1,1,....

### Pergunta P1.2 - Buffer overflow em várias linguagens



### Pergunta P1.4 - Read overflow

```

...../a.out.LS
_COLORS=rs=0;di=01;34:ln=01;36:mh=00;pi=40;33:so=01;35:do=01;35:bd=40;33;01:cd=4
0;33;01:or=40;31;01:mi=00:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:
ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31
:*.lzh=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=
01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;
31:*.lzo=01;31:*.xz=01;31:*.zst=01;31:*.tzst=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=
01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.
ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;
31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.mjpg=01;35:*.mjp
eg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35
:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz
=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:
*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob
=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.f
lc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:
*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=00;

```

Pode ser feito o acesso a células de memória fora do scope do programa.

### Pergunta P1.5

```
, instead of 0x61626364user@CSI:~/Aulas/1718-EngSeg/TPraticas/Aula10/codigofonte
999999999999999999999999999999999999999999999999999999999999999999999999dcb
You win this game if you can change variable control to the value 0x61626364'
Congratulations, you win!!! You correctly got the variable to the right value
user@CSI:~/Aulas/1718-EngSeg/TPraticas/Aula10/codigofonte$
```

- override do buffer todo
- tabela ascii
- inserir em ordem inversa pois estamos a inserir de baixo para cima no inteiro e ele é lido em big endian.

### Pergunta P1.6

Endereço pode ser obtido através do uso do gdb disassemble win ou através de objdump -d a.out.

End:begin + 00000000000000740;

Para facilitar imprimiu-se o endereço.

[illegible]

### Pergunta P1.7

todo.