CSE 2010 || Secure Coding

WIN 20-21

Lab: 8

Name: R B Ch S Tarun RegNo: 19BCN7122

Topic: Working with the memory vulnerabilities – Part II

Lab experiment - Working with the memory vulnerabilities – Part II

Task

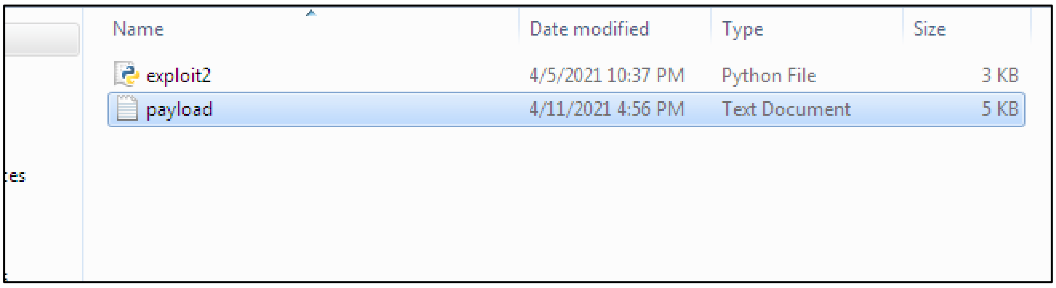
- Download Vulln.zip from teams.
- Deploy a virtual windows 7 instance and copy the Vulln.zip into it.
- Unzip the zip file. You will find two files named exploit.py and Vuln_Program_Stream.exe
- Download and install python 2.7.* or 3.5.*
- Run the exploit script II (exploit2.py- check today's folder)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

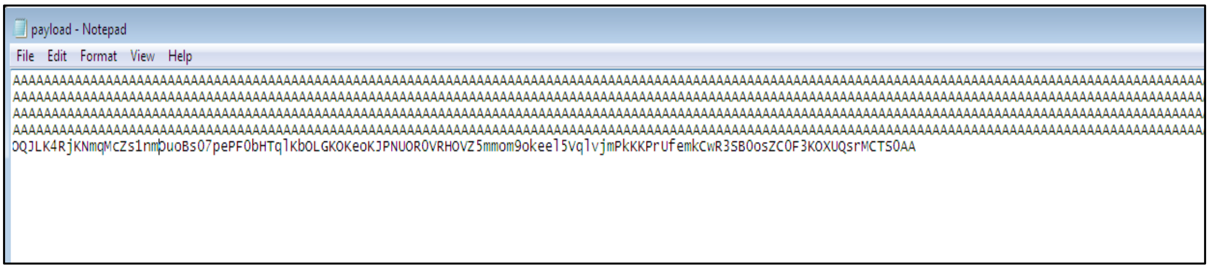
Analysis

- Try to crash the Vuln_Program_Stream program and exploit it.
- Change the default trigger from cmd.exe to calc.exe (Use msfvenom in Kali linux).
  Example:
  msfvenom -a x86 --platform windows -p windows/exec CMD=calc -e x86/alpha_mixed -b "\x00\x14\x09\x0a\x0d" -f python
- Change the default trigger to open control panel.

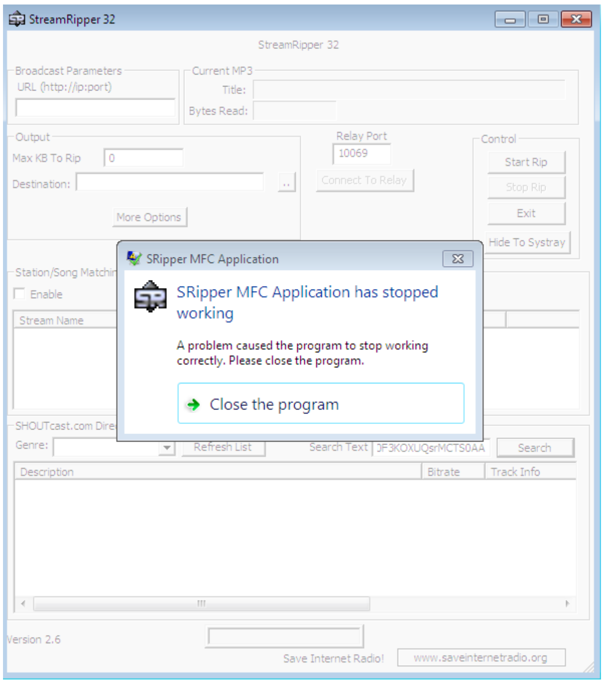
Happy Learning!!!!!!

First we have to generate the payload by running the “exploit2.py” in cmd.





Now after the generation of payload we need to inject the payload into stream ripper, after injecting the payload in to stream ripper its crashed.



This is due to the buffer overflow vulnerability.

A buffer overflow vulnerability occurs when you give a program too much data. The excess data corrupts nearby space in memory and may alter other data. As a result, the program might report an error or behave differently. Such vulnerabilities are also called buffer overrun

And, hence the crashing the application with payload is successful.

Now, PHASE – 2

We have to change the default trigger from cmd.exe to calc.exe.

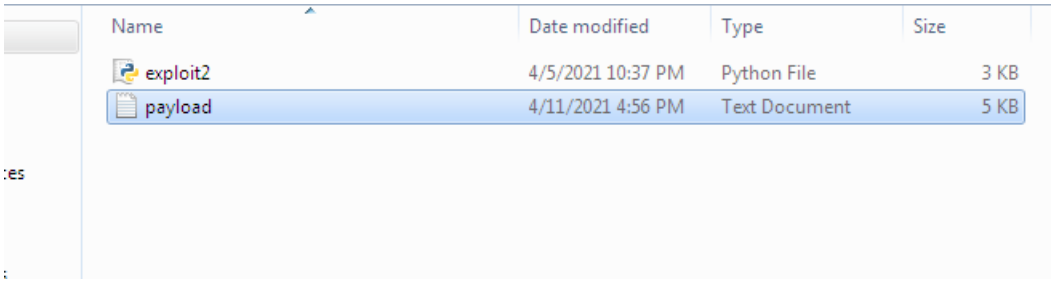
And to do that we have to generate the shell code using msf venom in kali linux.

The command used to generate the shell code is:

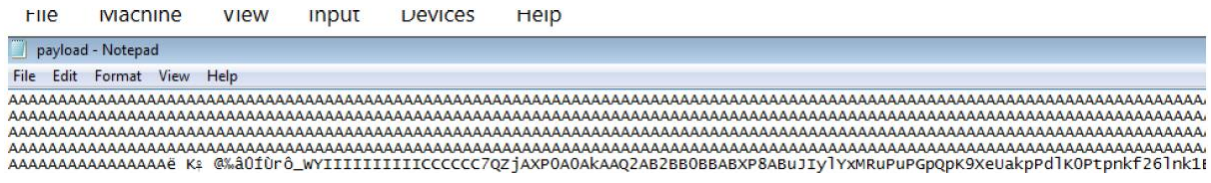
```
msfvenom -a x86 --platform windows -p windows/exec
CMD=calc -e x86/alpha_mixed -b "\x00\x14\x09\x0a\x0d"  -f
python
```

Now we have to change the shell code in exploit2.py and then we have to run the python script in command prompt.

After that we'll get the payload as before.



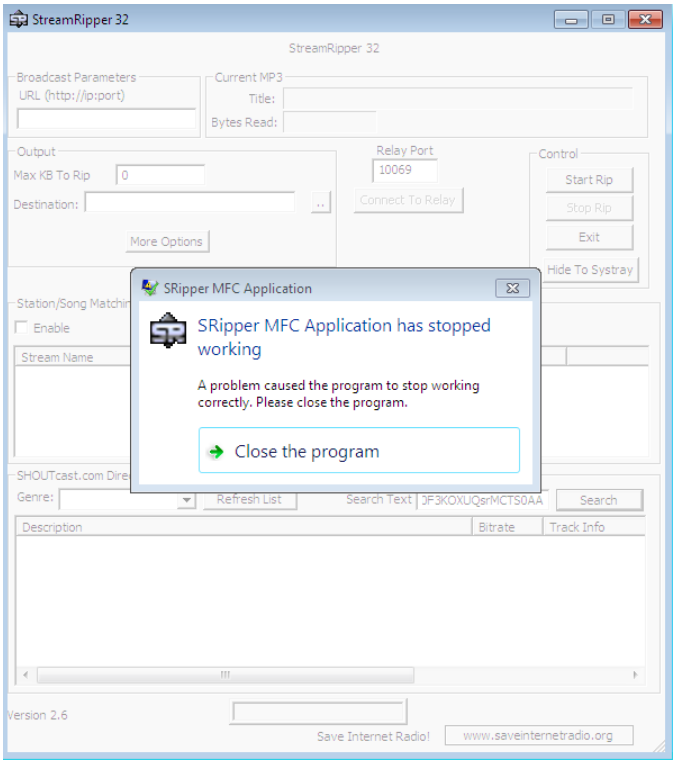
Payload after executing exploit2.py script:(for calculator)



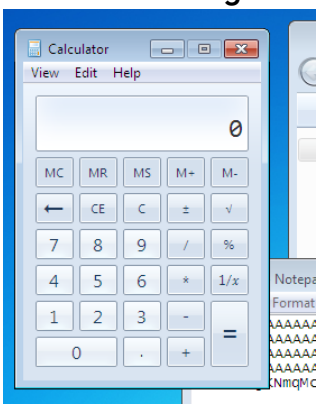
After running the payload in the vuln :

Analysis:

1. The vulln program crossed asusual.



2. After crashing the vulln program calculator opened..



This happened due to buffer overflow vulnerability,
A buffer overflow vulnerability occurs when you give a program too much data. The excess data corrupts nearby space in memory and may alter other data. As a result, the program might report

an error or behave differently. Such vulnerabilities are also called buffer overrun.

The same goes to the control panel, after changing the default trigger and generate the shell code and and changing the shell code in python script and generating the payload.

The application crashes and control panel opens.

This also due to the buffer overflow vulnerability.