# Secure Coding (CSE 2010)

# LAB Experiment: 7

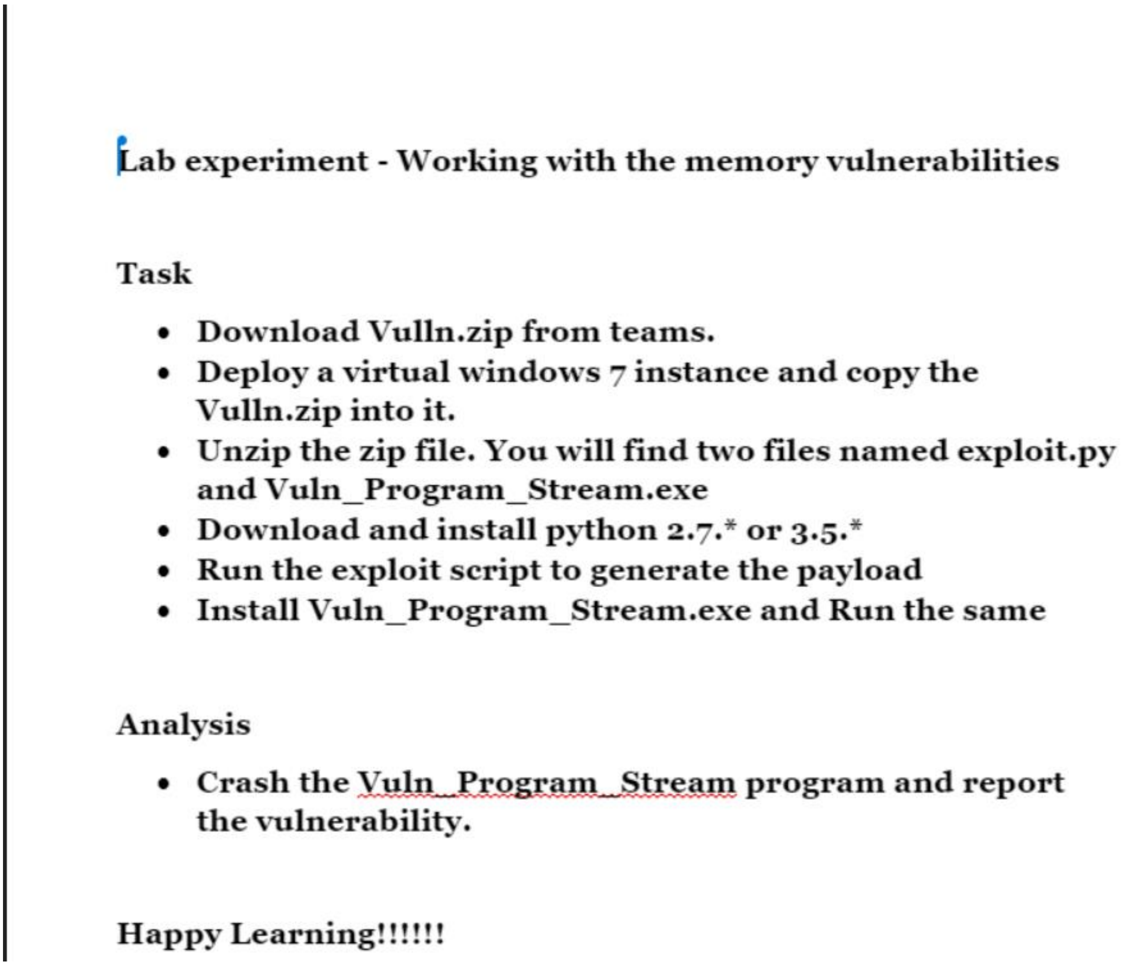
Done by,

Aadil Mohammed

Reg.No:19BCI7052

SLOT: L23+L24

**Question :**



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

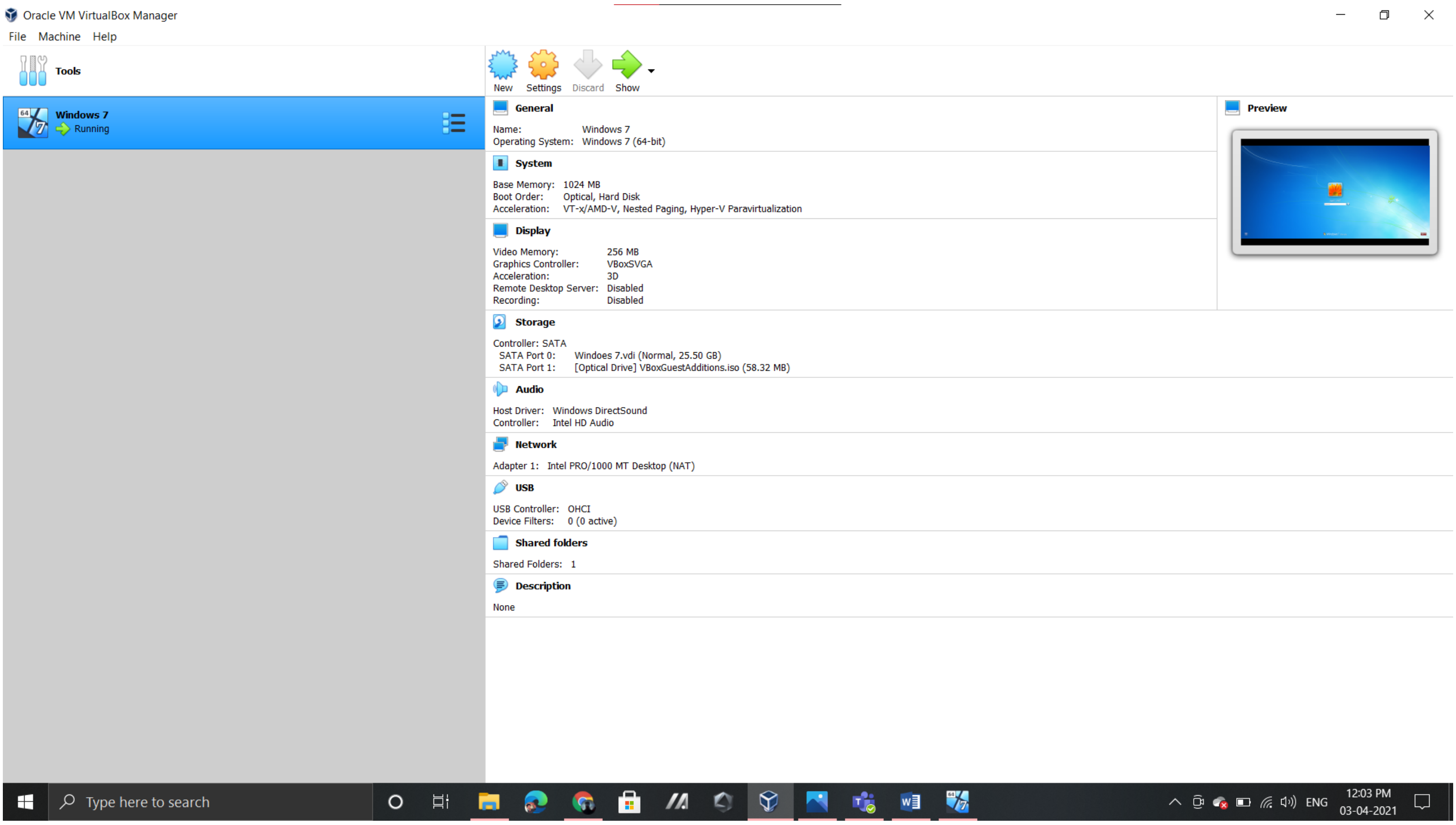
Analysis

- Crash the Vuln_Program_Stream program and report the vulnerability.

Happy Learning!!!!!!

## Solution:

1. Deployment of windows 7 virtual machine was done by using Oracle's VirtualBox, and the windows 7 Ultimate 64-bit ISO file was downloaded from getintopc.com.



**General**
Name: Windows 7
Operating System: Windows 7 (64-bit)

**System**
Base Memory: 1024 MB
Boot Order: Optical, Hard Disk
Acceleration: VT-x/AMD-V, Nested Paging, Hyper-V Paravirtualization

**Display**
Video Memory: 256 MB
Graphics Controller: VBoxSVGA
Acceleration: 3D
Remote Desktop Server: Disabled
Recording: Disabled

**Storage**
Controller: SATA
SATA Port 0: Windoes 7.vdi (Normal, 25.50 GB)
SATA Port 1: [Optical Drive] VBoxGuestAdditions.iso (58.32 MB)

**Audio**
Host Driver: Windows DirectSound
Controller: Intel HD Audio

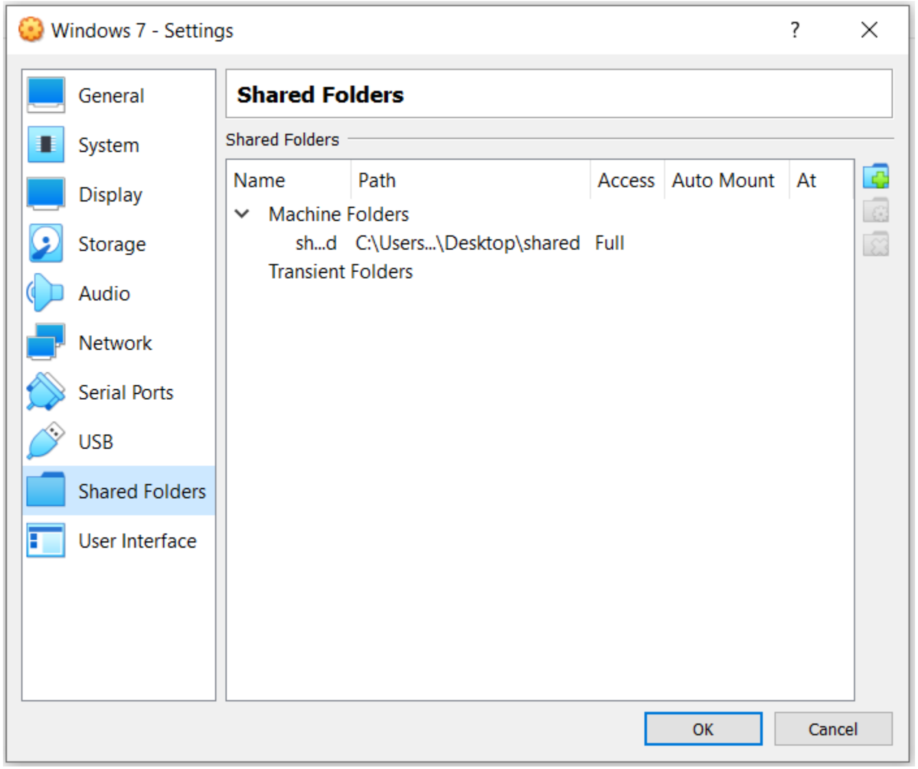
**Network**
Adapter 1: Intel PRO/1000 MT Desktop (NAT)

**USB**
USB Controller: OHCI
Device Filters: 0 (0 active)

**Shared folders**
Shared Folders: 1

**Description**
None

2. The Vullln.zip file was shared to virtual machine from the Host pc by using Oracles VirtualBox Guest additions: Shared folders, by sharing the desktop file through the networks folders in the Machine Folder directory.



**Shared Folders**

Shared Folders

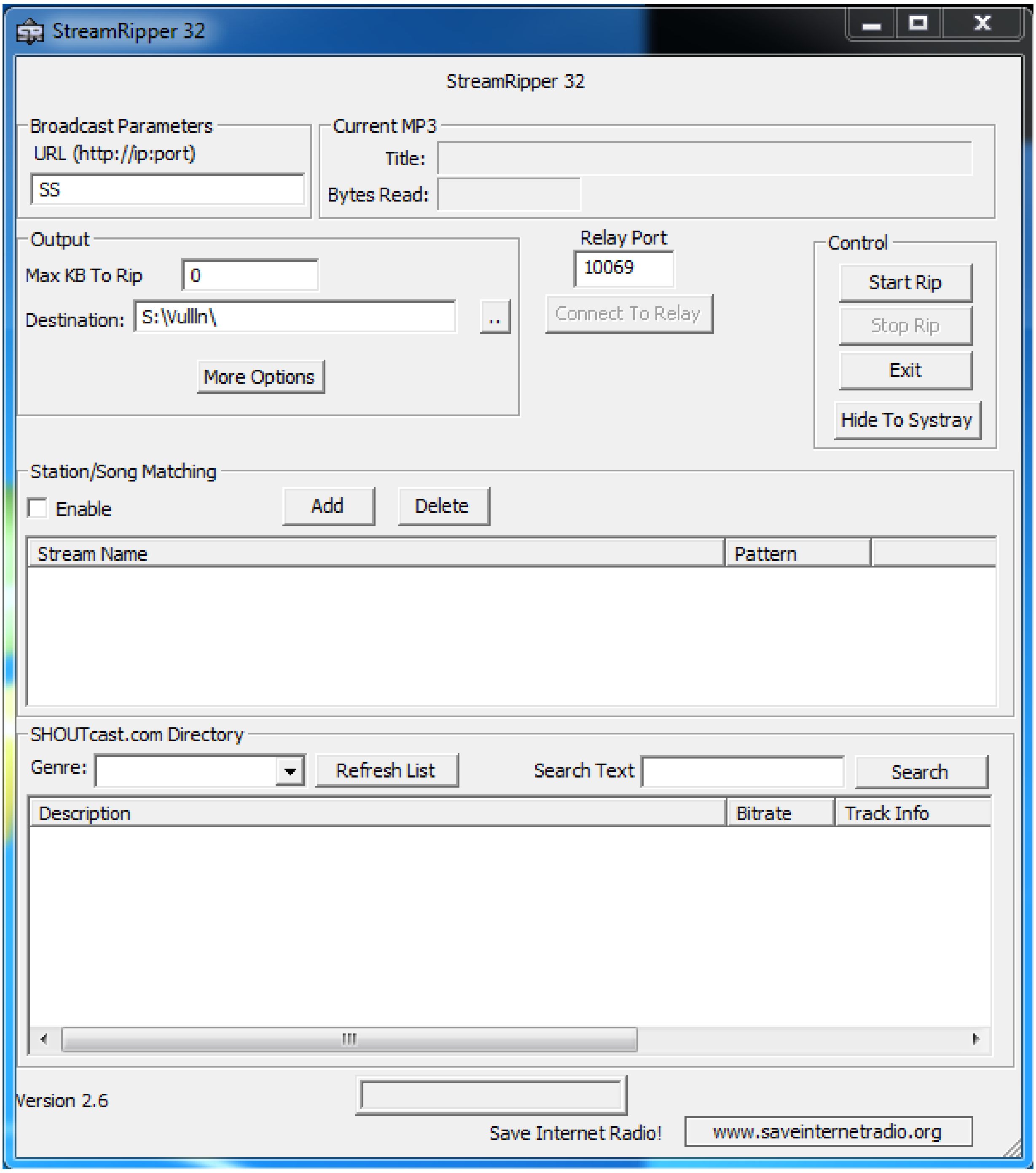
| Name | Path | Access | Auto Mount | At |
|---|---|---|---|---|
| Machine Folders | | | | |
| sh...d | C:\Users...\Desktop\shared | Full | | |
| Transient Folders | | | | |

3. After extracting the Vullln.zip file, compile the exploit python script (in python 2.7). Then in the same directory of the exploit script an exploit payload text file will be created.

<u>Payload:</u>

```
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
ë▯▯ôZ▯▯▯▯▯▯▯▯ÚÇºîPSàÙt$ô]3É±Rƒíü1U————————————————————————————»C±¿Œ·Ö?MØ_Ú|Ø
————————————————————————————————————————————————————————————————————————————————————
¯/èOýÃƒWáŠÐLíáýÍ4aü–XÏW×▯2•…v89òt'²H˜"›°öÂù÷~áºÕšï0äJ>¸K³ŽK•ô)´àJIóË0•vÏ"^ +%²·¸)▯³æ-~•J—
qÛO¼U‡ÝÌmúâÎ£FEã°ú
————————————————————————————————————————————————————————————————————————————————————
lt7¶l@Å^½úAÓ6%–m'ëŽâ(Ú²9™cY¹&¶Îéˆi¯YiÚG³fw¼¬.G''KT6yŽZ9Á¼S%NÌÜËãm
Ž®ªåo`[ƒc«ÞÙ°´ôu^&"…)[Ò~-E¶'"ÿ¤n@Çlµ±Æm8▯ì},,©)XYg‡-3ÉqÉèƒŒÂc'â‹ç³´•o4Íó»
————————————————————————————————————————————————————————————————————————————————————
°0^ŒÍØÇEl…÷°³°{0LGc1I————————————————————————————#ª#ÆÌ•Ã
```

4. From the extracted files install **vuln_Program_Stream,** after opening the StreamRipper32 application from the desktop of the VM, a menu appears:



StreamRipper 32

Broadcast Parameters
URL (http://ip:port): SS

Current MP3
Title:
Bytes Read:

Output
Max KB To Rip: 0
Destination: S:\Vullln\

Relay Port: 10069

Station/Song Matching
□ Enable

| Stream Name | Pattern | |
|---|---|---|

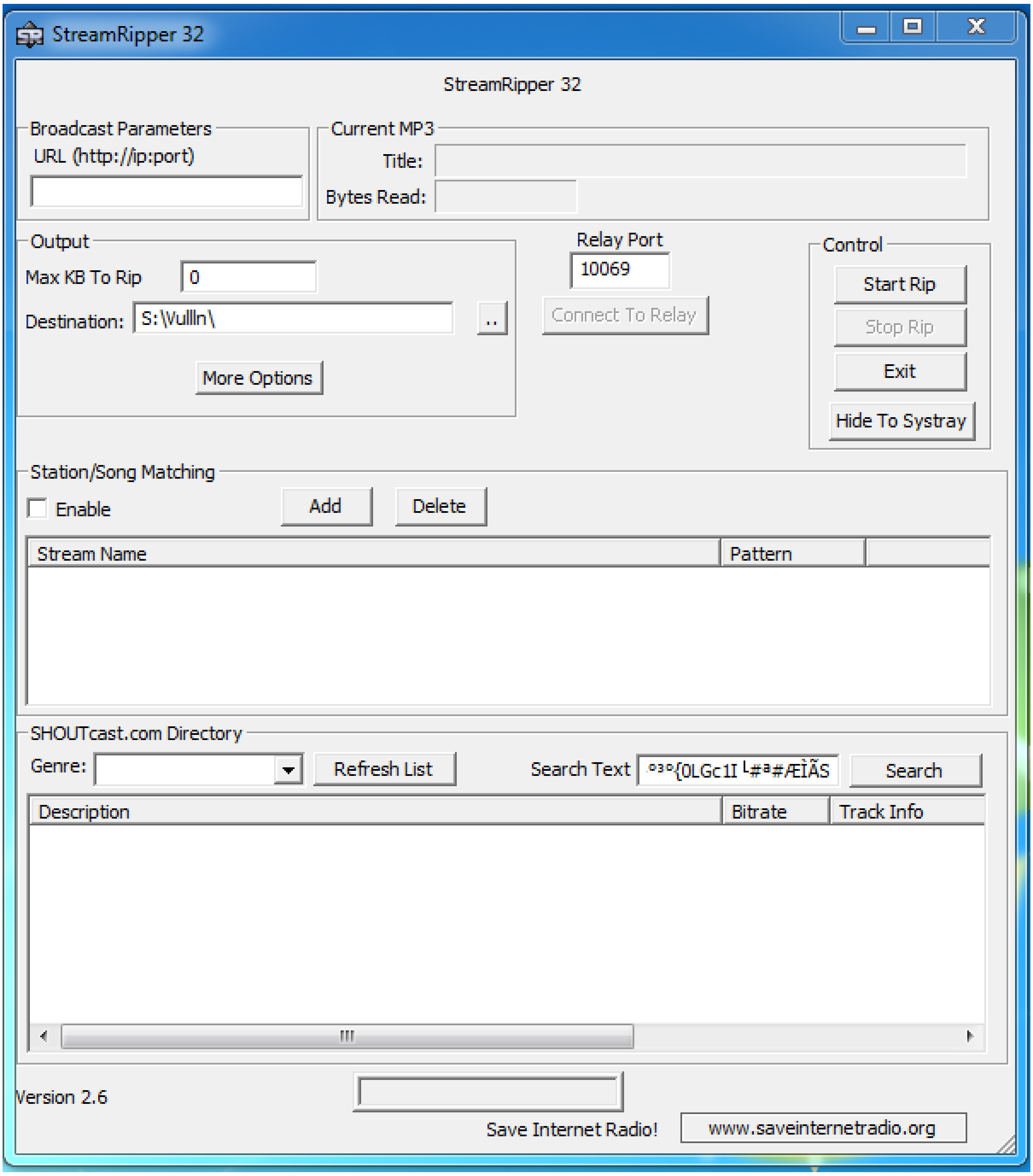
SHOUTcast.com Directory
Genre:
Search Text

| Description | Bitrate | Track Info |
|---|---|---|

Version 2.6

Save Internet Radio!

www.saveinternetradio.org



Relay Port: 10069

Max KB To Rip: 0

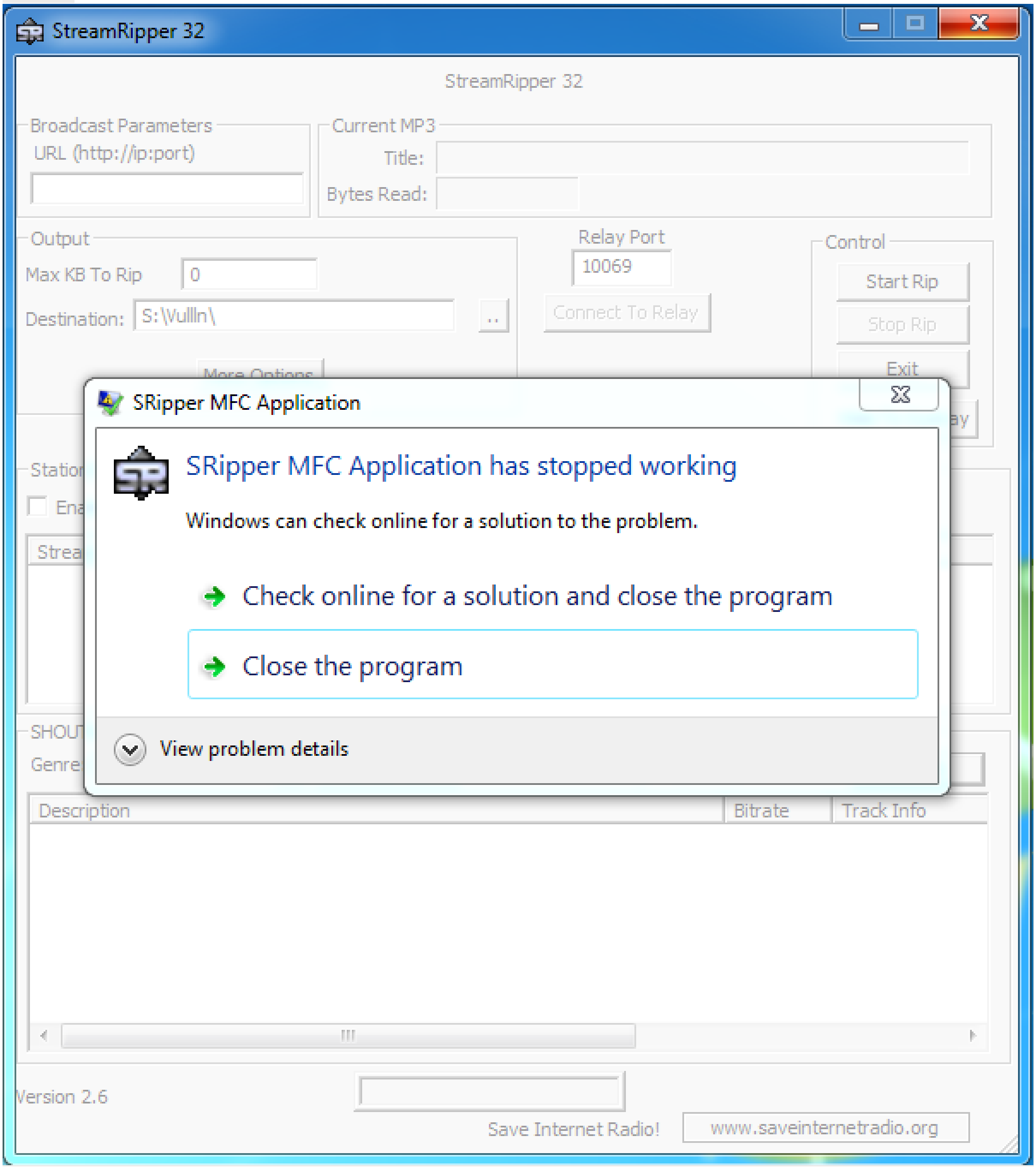
Destination: S:\Vullln\

Search Text: ᵒ³ᵒ{0LGc1I └#ª#ÆÌÃS

Version 2.6

5. Now paste the exploit payload in the Search text section as follows:

6. Click the search button, which will cause the application to crash and resulting in the error message as follows:



SRipper MFC Application

SRipper MFC Application has stopped working

Windows can check online for a solution to the problem.

Analysis: The above application StreamRipper32 was crashed was built around the 32 bit architecture and can only handle 32 bit queries and files, when the exploit payload is placed in the search text textfield query which was a 64 bit query and search button is pressed, it causes the application to crash. Perhaps the textfield input which was entered exceeded the 32 bits of stack memory allotted to that search query field during compile time.