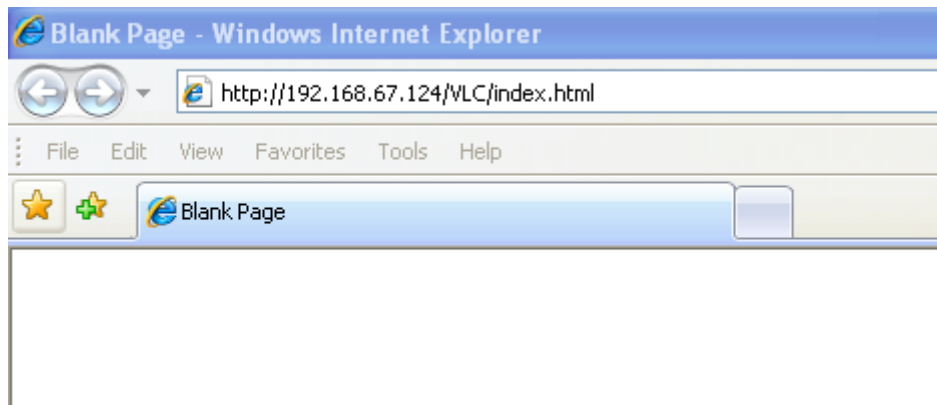


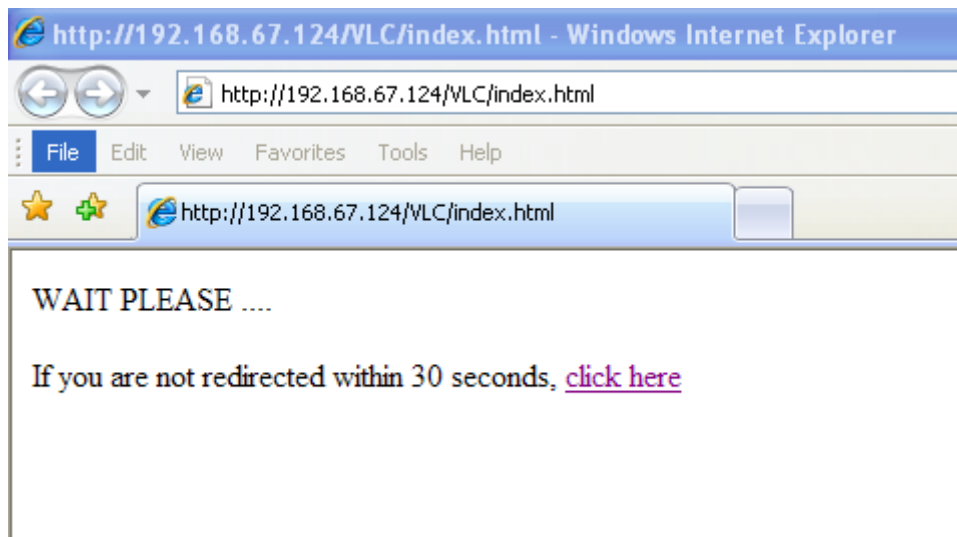
Practical demonstration of VLC ActiveX vulnerability

For this demonstration I will use a web server, the link can be sent to the victim by mail, in this case we send him a link to a webserver, which is:

<http://192.168.67.124/VLC/index.html>



The victim should click on the link included in the mail, or we should suggest him somehow to visit that link.



When executed, the javascript code crashed here:

| Address | Hex dump | ASCII |
|----------|---|-------------------|
| 04791F20 | 00 73 00 00 05 00 02 00 A2 01 08 02 E0 2F 79 04 | .s..+.0.660x/y+ |
| 04791F30 | FF FF FF FF 61 70 61 72 61 74 6F 5D 5B 40 5B 70 | aparato][@[p |
| 04791F40 | 69 73 74 61 5D 5D 00 04 C0 1B 79 04 06 00 05 00 | ista]].+y+.+ |
| 04791F50 | AF 01 08 02 F8 E6 78 04 F4 05 4F 80 A8 C5 78 04 | *60°px+r+0C2+X+ |
| 04791F60 | F4 05 4F 80 60 C6 78 04 F4 05 4F 80 08 D7 78 04 | r+0C° Fx+r+0C°Hr+ |
| 04791F70 | F4 05 4F 80 30 C5 78 04 F4 05 4F 80 02 00 06 00 | r+0C°+X+r+0C°+.+ |
| 04791F80 | B5 01 08 02 30 15 74 6D 00 00 00 00 02 00 02 00 | r600Stm....0.0. |
| 04791F90 | B7 01 08 02 30 15 74 6D 00 00 00 00 02 00 02 00 | r600Stm....0.0. |
| 04791FAC | B1 01 08 02 60 61 69 6E 00 01 37 00 02 00 02 00 | 070main.07.0.0. |
| 04791FBC | B3 01 08 02 47 65 6E 65 72 61 6C 00 02 00 02 00 | 160General.0.0. |
| 04791FCC | BD 01 08 02 00 C9 78 04 C8 C7 78 04 11 00 02 00 | 4600.Fx+H x+.0. |
| 04791FDC | BF 01 08 02 00 4A D8 65 A8 E5 78 04 69 74 69 6F | r600.Jf+ed0x+itio |
| 04791FEC | 6E 00 00 00 03 00 03 00 BA 01 08 02 74 69 6D 65 | n...+.+. 00Stime |

We can see that this zone is full of garbage, and in this case that the execution will jump to 7300.

In the following third example, it will jump to 370200, which is part of the garbage that was left there:

| | | |
|----------|-----------------|-------------------------------|
| 04431FC2 | 5B | MOV EBP,0 |
| 04431FC3 | E9 18C30500 | POP EBX |
| 04431FC8 | 8B02 | JMP axvlc.0448E2E0 |
| 04431FCA | 891424 | MOV EAX,DWORD PTR DS:[EDX] |
| 04431FCD | FF50 14 | MOV DWORD PTR SS:[ESP],EDX |
| 04431FD0 | 8B53 08 | CALL DWORD PTR DS:[EAX+14] |
| 04431FD3 | 85D2 | MOV EDX,DWORD PTR DS:[EBX+8] |
| 04431FD5 | 0F84 19FFFFFF | TEST EDX,EDX |
| 04431FDB | 90 | JE axvlc.04431EF4 |
| 04431FDC | 8D7426 00 | NOOP |
| 04431FE0 | 8B0A | LEA ESI,DWORD PTR DS:[ESI] |
| 04431FE2 | 891424 | MOV ECX,DWORD PTR DS:[EDX] |
| 04431FE5 | FF51 64 | MOV DWORD PTR SS:[ESP],EDX |
| 04431FE8 | 8B53 3C | CALL DWORD PTR DS:[ECX+64] |
| 04431FEB | 85D2 | MOV EDX,DWORD PTR DS:[EBX+3C] |
| 04431FED | 0F84 0CFFFFFF | TEST EDX,EDX |
| 04431FF3 | 8DB6 00000000 | JE axvlc.04431EFF |
| 04431FF9 | 8DBC27 00000000 | LEA ESI,DWORD PTR DS:[ESI] |
| 04431FFB | 8DB6 00000000 | LEA EDI,DWORD PTR DS:[EDI] |
| 04431FFD | 8DB6 00000000 | MOV ECX,DWORD PTR DS:[ECX] |

DS:[0456B4E4]=00370200

| Address | Hex dump | ASCII |
|----------|---|------------------|
| 0456B4E4 | 00 02 37 00 02 00 02 00 F3 01 08 02 6D 61 69 6E | .07.0.0.3070main |
| 0456B4F4 | 00 02 37 00 08 00 02 00 F1 01 0A 02 00 00 00 00 | .07.0.0.30.0.... |
| 0456B504 | 30 30 33 32 30 5D 20 6D 61 69 6E 20 61 63 63 65 | 00320] main acce |
| 0456B514 | 73 73 20 64 65 62 75 67 3A 20 63 6F 6E 6E 65 63 | ss debug: connec |
| 0456B524 | 74 75 7F 7F 32 32 75 30 72 73 7F 7F 75 7F 75 73 | |

This way, approximately 1 in 20 executions will jump to our own code. This needs more research to make it more reliable, but it can be seen that the bad initialized pointer makes the program crash, and jump to any direction left in memory by previous files, or by previously visited pages.

Ricnar