Abysssec Research

1) Advisory information

Title	**: Microsoft MPEG Layer-3 Audio Decoder Division By Zero**
Version	**: l3codeca.acm 1-9-0-306 (XP SP2 – XP SP3)**
Discovery	**: http://www.abysssec.com**
Vendor	**: http://www.microsoft.com**
Impact	**: Med/High**
Contact	**: shahin [at] abysssec.com , info [at] abysssec.com**
Twitter	**: @abysssec**

2) Vulnerability Information

Class

1- Division By Zero

Impact

Successfully exploiting this issue allows remote attackers to cause denial-of-service conditions.

Remotely Exploitable

Yes

Locally Exploitable

Yes

3) Vulnerabilities detail

1- Division by Zero:

The vulnerability will occur during paring malicious AVI file contains MPEG Layer-3 stream.

In l3codeca.acm DriverProc (sub_3D522940) routine is responsible to choice true function for parsing various AVI files. This routine takes 5 arguments. Due to third argument is routine it will call true routine for encoded AVI. If third argument is equal to 0x604E, sub_3D522940 will call:

```
.text:3D522C60              mov     eax, [esp+uMsg]
.text:3D522C64              cmp     eax, 600Ah
.text:3D522C69              ja      loc_3D522CF0
.text:3D522C6F              jz      short loc_3D522CDE
.text:3D522C71              lea     ecx, [eax-1]    ; switch 10 cases
.text:3D522C74              cmp     ecx, 9
.text:3D522C77              ja      loc_3D522DA8    ; default
.text:3D522C77                                      ; jumptable 3D522C7D cases 2,5
.text:3D522C77                                      ; jumptable 3D522D07 cases 1-13,17-64
.text:3D522C7D              jmp     ds:off_3D522DD8[ecx*4] ; switch jump
.text:3D522C84
.text:3D522C84 loc_3D522C84:                        ; DATA XREF: .text:off_3D522DD8o
.text:3D522C84              mov     eax, 1          ; jumptable 3D522C7D cases 1,6
.text:3D522C89              retn    14h
.text:3D522C8C ; ---------------------------------------------------------------------------
.text:3D522C8C
.text:3D522C8C loc_3D522C8C:                        ; CODE XREF: DriverProc+1Dj
.text:3D522C8C                                      ; DATA XREF: .text:off_3D522DD8o
.text:3D522C8C              mov     eax, [esp+lParam2] ; jumptable 3D522C7D case 3
.text:3D522C90              mov     ecx, [esp+hDriver]
.text:3D522C94              push    eax             ; int
.text:3D522C95              push    ecx             ; hDriver
.text:3D522C96              call    sub_3D521D00
.text:3D522C9B              retn    14h
.text:3D522C9E ; ---------------------------------------------------------------------------
...
.text:3D522D84
.text:3D522D84 loc_3D522D84:                        ; CODE XREF: DriverProc+A7j
.text:3D522D84                                      ; DATA XREF: .text:off_3D522E00o
.text:3D522D84              mov     edx, [esp+lParam2] ; jumptable 3D522D07 case 67
.text:3D522D88              mov     eax, [esp+hWndParent]
.text:3D522D8C              push    edx
.text:3D522D8D              push    eax
.text:3D522D8E              call    sub_3D522940
.text:3D522D93              retn    14h
```

sub_3D522940 is responsible to parsing MPEGLAYER3WAVEFORMAT (same strf in AVI) , if the value of wFormatTag in WAVEFORMATEX structure (subset of MPEGLAYER3WAVEFORMAT structure) equals with 0x0055 means the type of audio stream is MP3:

```
.text:3D522A9F          mov    [ebp+0Ch], ecx
.text:3D522AA2          cmp    word ptr [edi], 55h
.text:3D522AA6          jnz    loc_3D522B2C
```

If type of stream is MP3, there will be some computational for fields of WAVEFORMATEX structure. First the value of nSamplesPerSec will be read then due to value of this field a variable will give 1152(0x480) or 576 (0x240):

```
.text:3D522AAC          mov    ecx, [edi+4]     ; nSamplesPerSec
.text:3D522AAF          xor    edx, edx
.text:3D522AB1          mov    eax, ecx
.text:3D522AB3          div    dword ptr [esi+4]
.text:3D522AB6          mov    edx, 5DC0h
.text:3D522ABB          cmp    edx, ecx
.text:3D522ABD          sbb    ebx, ebx
.text:3D522ABF          xor    edx, edx
.text:3D522AC1          and    ebx, 240h
.text:3D522AC7          add    ebx, 240h
```

After a bit nAvgBytesPerSec from WAVEFORMATEX structure will multiply with 0x480 or 0x240 and the result will be dived by with value of nSamplesPerSec field.

```
.text:3D522AD1          mov    eax, [edi+8]    ; nAvgBytesPerSec
.text:3D522AD4          imul   eax, ebx
.text:3D522AD7          div    ecx
```

Vulnerable point is here, because no control performed for these fields. If you continue to look at disassembly you will find value of nBlockAlign from WAVEFORMATEX structure will be dividing to result of previously divided value. so if result of previously divided value is 0 with new division we will met an “integer division by zero” error:

```
.text:3D522AD9          mov    edi, [ebp+8]    ; nBlockAlign
.text:3D522ADC          xor    edx, edx
.text:3D522ADE          mov    ecx, eax
.text:3D522AE0          mov    eax, edi
.text:3D522AE2          div    ecx     →  crash point
```

The proof of concept for proving vulnerability Is annexed to this doc in mp3-poc.zip

We contacted Microsoft and they told us this issue is fixed in ms10-052 bulletin but that advisory fixes vulnerability exists in (l3codecx.ax not in l3codeca.acm) and they didn’t fix our vulnerability at all.