Abysssec Research

1) Advisory information

Title	: Apple QuickTime FlashPix NumberOfTiles Remote Code Execution Vulnerability
Version	: QuickTime player 7.6.5
Discovery	: http://www.abysssec.com
Vendor	: http://www.apple.com
Impact	: Med/High
Contact	: shahin [at] abysssec.com , info [at] abysssec.com
Twitter	: @abysssec
CVE	: CVE-2010-0519

2) Vulnerable version

Apple QuickTime Player 7.6.5
Apple QuickTime Player 7.6.4
Apple QuickTime Player 7.6.2
Apple QuickTime Player 7.6.1
Apple QuickTime Player 7.6
Apple Mac OS X Server 10.6.2
Apple Mac OS X Server 10.6.1
Apple Mac OS X Server 10.6
Apple Mac OS X 10.6.2
Apple Mac OS X 10.6.1
Apple Mac OS X 10.6

3) Vulnerability information

Class

1- Code execution

Impact

Successful exploits may allow attackers to execute arbitrary code in the context of the currently logged-in user; failed exploit attempts will cause denial-of-service conditions.

Remotely Exploitable

Yes

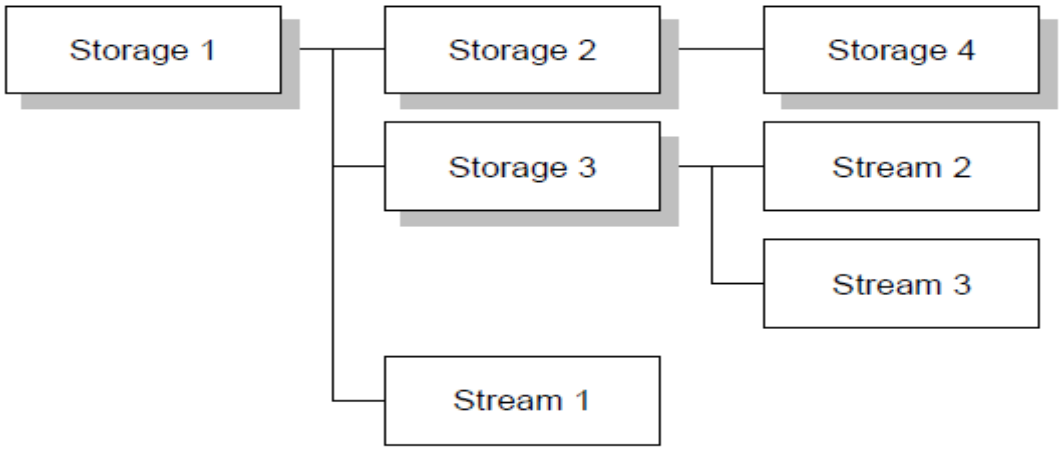
Locally Exploitable

Yes

4) Vulnerabilities detail

Integer overflow:

The FlashPix file format structure is similar to a system file in which the whole file consists of storages and streams. A storage is similar to a folder in a system file and a stream is analogous to a file. Every storage can contain other storages and streams in exactly the same way that every folder can contain folders and files in a system file. The image below shows the concept:



One of the various streams that exist in the file format is SubImage. The SubImage steam consists of a Header and Data where the Header is responsible for Data details and Data contains image information.

In this file format, the image is divided to 64pix*64pix tiles and the number of tiles are stored in the SubImage stream header. The QuickTime Player software reads the number of tiles from the NumberOfTiles field of the header, multiplies it by 16, and allocates the required heap memory based on the result of the multiplication. In the next stage, the app copies the information to the allocated memory based on the number of tiles. In cases where the result of the multiplication is more than 32bits, the allocated memory will be less than the length of the NumberOfTiles in the file and we can write to the heap with the size of the substitution of these two numbers. Now we are going to explain the binary based on the discussed material:

```
.text:67ADB6F0     push    ecx
.text:67ADB6F1     push    esi
.text:67ADB6F2     push    edi
.text:67ADB6F3     xor     edi, edi
.text:67ADB6F5     mov     esi, ecx
.text:67ADB6F7     cmp     [esi+56h], edi
.text:67ADB6FA     mov     [esp+0Ch+var_4], edi
.text:67ADB6FE     jnz     loc_67ADB7DD
.text:67ADB704     mov     eax, [esi+22h]
.text:67ADB707     shl     eax, 4
.text:67ADB70A     push    eax
.text:67ADB70B     call    sub_67B6FDB0
.text:67ADB710     add     esp, 4
.text:67ADB713     cmp     eax, edi
.text:67ADB715     mov     [esi+56h], eax
.text:67ADB718     jnz     short loc_67ADB721
.text:67ADB71A     lea     eax, [edi-6Ch]
.text:67ADB71D     pop     edi
.text:67ADB71E     pop     esi
.text:67ADB71F     pop     ecx
.text:67ADB720     retn
```

This flaw exists in the QuickTimeImage.qtx file. The above code first shows that at address 67ADB704, the value of NumberOfTiles is stored in the EAX register. This value is then multiplied by 16 with a shift left instruction at address 67ADB707 and the result is passed to QuickT_B.67B6FDB0 for allocating memory without bounds checking. For example, if we put 41414141 in this field, the result would be 14141410 after the instruction which is less than 41414141.

In the next section, the values will be copied to memory in a loop that is controlled by NumberOfTiles.

```
.text:67ADB740     mov     ecx, [esi+5Eh]
```

```
.text:67ADB743     mov     edx, [ecx]
.text:67ADB745     mov     eax, [edx+8]
.text:67ADB748     push    0
.text:67ADB74A     push    ebx
.text:67ADB74B     call    eax
.text:67ADB74D     test    al, al
.text:67ADB74F     jz      short loc_67ADB7BF
.text:67ADB751     mov     eax, [esi+56h]
.text:67ADB754     mov     ecx, [esi+5Eh]
.text:67ADB757     mov     eax, [eax]
.text:67ADB759     mov     edx, [ecx]
.text:67ADB75B     mov     edx, [edx+1Ch]
.text:67ADB75E     add     eax, edi
.text:67ADB760     push    eax
.text:67ADB761     call    edx
.text:67ADB763     test    al, al
.text:67ADB765     jz      short loc_67ADB7BF
.text:67ADB767     mov     edx, [esi+56h]
.text:67ADB76A     mov     ecx, [esi+5Eh]
.text:67ADB76D     mov     edx, [edx]
.text:67ADB76F     mov     eax, [ecx]
.text:67ADB771     mov     eax, [eax+1Ch]
.text:67ADB774     lea     edx, [edx+edi+4]
.text:67ADB778     push    edx
.text:67ADB779     call    eax
.text:67ADB77B     test    al, al
.text:67ADB77D     jz      short loc_67ADB7BF
.text:67ADB77F     mov     eax, [esi+56h]
.text:67ADB782     mov     ecx, [esi+5Eh]
.text:67ADB785     mov     eax, [eax]
.text:67ADB787     mov     edx, [ecx]
.text:67ADB789     mov     edx, [edx+1Ch]
.text:67ADB78C     lea     eax, [eax+edi+8]
.text:67ADB790     push    eax
.text:67ADB791     call    edx
.text:67ADB793     test    al, al
.text:67ADB795     jz      short loc_67ADB7BF
.text:67ADB797     mov     edx, [esi+56h]
.text:67ADB79A     mov     ecx, [esi+5Eh]
.text:67ADB79D     mov     edx, [edx]
.text:67ADB79F     mov     eax, [ecx]
.text:67ADB7A1     mov     eax, [eax+1Ch]
```

```
.text:67ADB7A4     lea     edx, [edx+edi+0Ch]
.text:67ADB7A8     push    edx
.text:67ADB7A9     call    eax
.text:67ADB7AB     test    al, al
.text:67ADB7AD     jz      short loc_67ADB7BF
.text:67ADB7AF     add     ebx, [esi+36h]
.text:67ADB7B2     add     ebp, 1
.text:67ADB7B5     add     edi, 10h
.text:67ADB7B8     cmp     ebp, [esi+22h]
.text:67ADB7BB     jb      short loc_67ADB740
.text:67ADB7BD     jmp     short loc_67ADB7C7
```

The value of NumberOfTiles which exists in esi+22h is checked against the EBP register as a counter at address 67ADB7B8 and in if the counter is less than NumberOfTiles, the execution flow will be moved to the beginning of the loop. At the next stage, EBP will be incremented by 1 and 16 will be added to the EDI register where EDI is the index of reading memory.

```
.text:668E27E8     mov     eax, [esi+ecx*4-4] ; Microsoft VisualC
2-9/net runtime
.text:668E27EC     mov     [edi+ecx*4-4], eax
.text:668E27F0     lea     eax, ds:0[ecx*4]
.text:668E27F7     add     esi, eax
.text:668E27F9     add     edi, eax
```

If we change the first NumberOfTiles value to 41414141 at address 668E27EC, an Access violation error occurs.