

# The Case of the Crashed Cell Phone Call

Mark Russinovich  
(From Mark Russinovich Blog)

David Solomon, my coauthor for the Windows Internals books, was recently in the middle of an important VOIP call on Skype when the audio suddenly garbled. A second later the system blue screened. He called back after the reboot, but a half hour later the person on the other seemed to stop talking mid-word and the system crashed again. The conversation was essentially over anyway, and since he'd explained the first drop, Dave decided not to call back and formally end the call, but to investigate the cause of the crashes. He launched Windbg from the Debugging Tools for Windows package, selected Open Crash Dump from the File menu, and chose %Systemroot%\Memory.dmp.

He'd previously configured Windbg to use the Microsoft public symbol server by entering "srv\*c:\symbols\*http://msdl.microsoft.com/download/symbols" in the Windbg symbols configuration dialog, so Windbg knew how to interpret the crash dump file. When Windbg loads a crash dump file, it automatically executes a heuristics-based analysis engine that identifies the driver or system component most likely responsible for the crash. The analysis output pointed at the NETw4v64.sys device driver:

```
*****
*                                     *
*                               Bugcheck Analysis                               *
*                                     *
*****
Use !analyze -v to get detailed debugging information.

BugCheck A, {96c000007e, 2, 1, fffff800028a7153}

Unable to load image \SystemRoot\system32\DRIVERS\NETw4v64.sys, Win32 error 0n2
*** WARNING: Unable to verify timestamp for NETw4v64.sys
*** ERROR: Module load completed but symbols could not be loaded for NETw4v64.sys
Probably caused by : NETw4v64.sys ( NETw4v64+66645 )
```

When you click on the "!analyze -v" hyperlink in the output, Windbg prints out some of the data it uses in its analysis. The analysis heuristics aren't perfect, so Dave always clicks the link to look at the additional data, specifically the stack trace at the time of the crash and possibly memory locations associated with the crash. The stack trace records the nesting of function calls on the processor from which the kernel's crash function, KeBugCheckEx, was called. In this case the stack looked like this:

```
nt!KeBugCheckEx
nt!KiBugCheckDispatch+0x6e
nt!KiPageFault+0x20b
nt!KeAcquireSpinLockRaiseToDpc+0x13
NETw4v64+0x66645
0xfffffa60`03f3d060
0xfffffa60`03f3d000
```

You read the stack from bottom to top to follow the chronology of function calls. The trace shows that some code in NETw4v64 called the kernel's ("nt") KeAcquireSpinLockRaiseToDpc function. NETw4v64's stack frame doesn't have a text function name, which is expected for drivers that aren't part of Windows and therefore don't have symbols on the Microsoft symbol server. The next higher frame indicates that KeAcquireSpinLockRaiseToDpc called KiPageFault, most likely not directly, but

# The Case of the Crashed Cell Phone Call

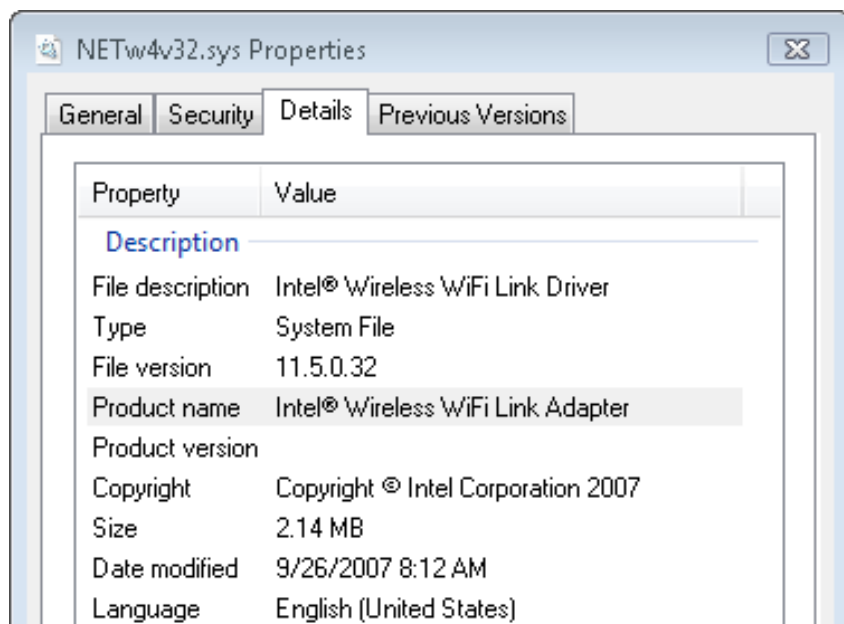
Mark Russinovich  
(From Mark Russinovich Blog)

as the result of a reference to a virtual memory address that wasn't currently resident in physical memory. KiPageFault then called KeBugCheckEx with stop code A, which the extended analysis output describes as IRQL\_NOT\_LESS\_OR\_EQUAL:

```
IRQL_NOT_LESS_OR_EQUAL (a)
An attempt was made to access a pageable (or completely invalid) address at an
interrupt request level (IRQL) that is too high. This is usually
caused by drivers using improper addresses.
If a kernel debugger is available get the stack backtrace.
Arguments:
Arg1: 0000096c0000007e, memory referenced
Arg2: 0000000000000002, IRQL
Arg3: 0000000000000001, bitfield :
    bit 0 : value 0 = read operation, 1 = write operation
    bit 3 : value 0 = not an execute operation, 1 = execute operation (only d
Arg4: fffff800028a7153, address which referenced memory
```

Dave hypothesized that the NETw4v64 driver had called the kernel with a corrupted pointer that triggered the invalid memory reference. This particular crash might have been the result of random corruption, even by another driver, so he looked in the %Systemroot%\Minidump directory for the dump file for the first crash. On Windows Vista, the operating system he was running, the system always saves a kernel-memory dump to %Systemroot%\Memory.dmp, overwriting the previous dump, and archives an abbreviated form of the dump, called a minidump, to %Systemroot%\Minidump. He followed the same steps for the second dump and the analysis engine reported the exact same cause for the crash, down to the same corrupted memory pointer value.

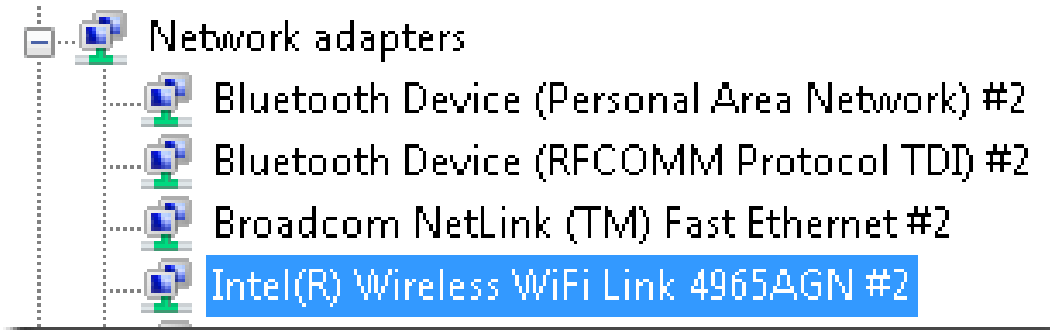
Without performing a meticulous manual analysis of a dump, you can't be certain that the driver the heuristics point at is the culprit, but the first rule of crash mitigation is to make sure you have the latest versions of any implicated drivers. Sometimes Windows Update has optional updates that don't apply automatically, so Dave went to the %Systemroot%\System32\drivers directory to investigate the NETw4v64.sys file for clues as to what device it was for. The file properties dialog showed that it was version 11.5 of the "Intel Wireless WiFi Link Driver":



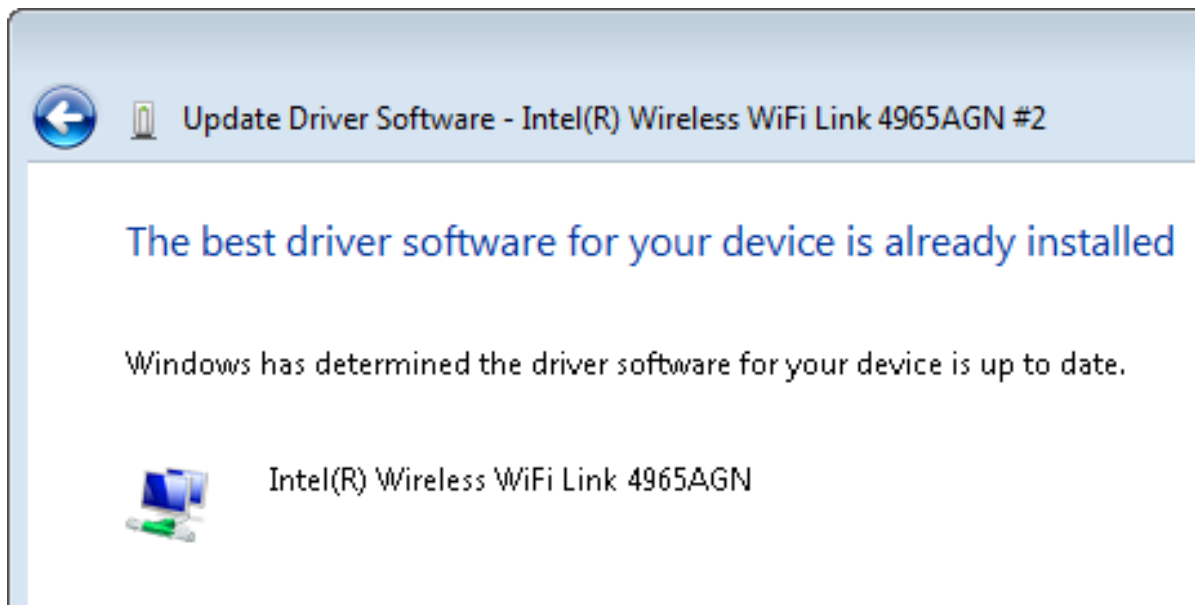
# The Case of the Crashed Cell Phone Call

Mark Russinovich  
(From Mark Russinovich Blog)

Armed with the knowledge that it was an Intel wireless network driver, he opened Device Manager, expanded the Network Adapters node and found a device with a similar name:



He right-clicked and chose “Update Driver Software...” from the context menu to launch the driver update wizard, and told it to check Windows Update for a newer version. Unfortunately, it reported that he had the most current version installed:



Sometimes OEMs have drivers posted on their Web sites that they haven't yet been made available to Windows Update, so Dave next went to Dell, the brand of his laptop, to check the version there. Again, the version he found posted seemed to match the one he had:

# The Case of the Crashed Cell Phone Call

Mark Russinovich  
(From Mark Russinovich Blog)

## Intel Intel(R) PRO/Wireless 3945ABG Network Connection [<< return to results](#)

---

**Release Date:** 3/6/2008

**Version:** 11.5.0.0 (TIC148234), A06 [▶ Other Versions](#)

**Download Type:** Driver

**File Format:** Hard-Drive

**File Size:** 75 MB

[Download Now](#)

[▶ Add to My Downloads](#)

[▶ Sign In to View My Saved Downloads](#)

OEMs often get hardware vendors to create custom versions of hardware tuned for specific cost, power, capability or size requirements. The original hardware vendor will therefore not post drivers for an OEM-only device or post drivers that are generic and might not take advantage of OEM-specific features. It's always worth checking, though, so Dave went to Intel's site. To his chagrin, not only was there a newer version that installed and worked as expected, but the Intel driver was version 12.1, a major release number higher than the one Dell was hosting:

## Downloads

---

### Intel® PRO/Wireless 3945ABG Network Connection

3 downloads on 1 page supported for **Windows Vista® Ultimate, 64-bit version**

---

**Download Types:**

- Drivers
- Software Applications

Drivers			
Title	Ver.#	Date	Download
1.  Intel® PRO/Wireless and WiFi Link Drivers-Only for Windows Vista* (5612KB) Microsoft Windows Vista* Drivers for Intel® WiMAX/WiFi and WiFi Links, Intel® Wireless WiFi Link 4965AGN and Intel® PRO/Wireless Network Connections.	12.1.2.1	11/24/2008	<a href="#">Download</a>

# The Case of the Crashed Cell Phone Call

Mark Russinovich  
(From Mark Russinovich Blog)

Intel also conveniently offered the driver in a “Drivers-Only” download that was a mere 7MB, one tenth the size of the package on Dell’s site that also includes value-add management software.

Dave couldn’t conclusively close the case because he couldn’t be sure that the Intel driver was the actual cause of the crashes, but the crashes haven’t reoccurred. Even if the Intel driver wasn’t the root cause, Dave was happy that he picked up a newer version that most likely had performance, reliability and maybe even power-management improvements. The case is a great example of simple dump analysis and the lesson that Windows Update and even an OEM’s site might not have the most up-to-date drivers. Hopefully, Dell will start leveraging Windows Update to provide its customers the latest drivers.