

# EEYE: Windows MIDI Decoder (QUARTZ.DLL) Heap Corruption

**Source:** <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2003-07/0301.html>

---

**From:** Derek Soeder (*dsoeder\_at\_eeeye.com*)

**Date:** 07/23/03

To: <bugtraq@securityfocus.com>

Date: Wed, 23 Jul 2003 14:49:42 -0700

Windows MIDI Decoder (QUARTZ.DLL) Heap Corruption

Release Date:

July 23, 2003

Severity:

High (Remote Code Execution)

Systems Affected:

Windows 98

Windows 98 SE

Windows Millennium Edition

Windows NT 4.0

Windows NT 4.0, Terminal Server Edition

Windows 2000

Windows XP

Windows Server 2003

Description:

A little over six hundred years ago, in a quaint German hamlet called Hamelin, the Pied Piper proved to the townsfolk that he could take control of their rodents and children with just a song. Turns out the same thing works on Windows.

Microsoft provides a component called QUARTZ.DLL that allows Windows applications to play MIDI music through a common interface. Windows Media Player and Internet Explorer, for example, both use QUARTZ.DLL to play MIDI music files (.mid extension); in the case of Internet Explorer, MIDI files can be played automatically when a web page is visited through the use of a specific HTML tag.

eEye Digital Security has discovered a pair of flaws in all versions of QUARTZ.DLL that would allow a specially-crafted MIDI file to cause the execution of arbitrary code when played. In the worst case, an attacker could construct a malicious .mid file and have it play automatically

## SecurityFocus Bugtraq: EEYE: Windows MIDI Decoder (QUARTZ.DLL) Heap Corruption

whenever a victim attempts to view certain HTML, such as an attacker-controlled website, resulting in the compromise of the victim's machine.

### Technical Description:

Modern folklore contends that some bands used to inject subliminal messages into their music by recording spoken commands or phrases and dubbing them backwards into the track. Although these allegations and the effectiveness of the technique were never proven conclusively, it is known that computers running a vulnerable version of QUARTZ.DLL will happily do whatever they're instructed to do without litigation, as long as the commands in the MIDI music are in machine language.

The QUARTZ.DLL vulnerability discussed in this advisory is a heap buffer overrun resulting from an integer overflow. If a Text or Copyright string with a specified length of FFFFFFFFh is included in the MIDI file, QUARTZ will attempt to allocate a zero-byte heap block, then copy the text string -- and any data following it -- to the newly-allocated location in the heap. As a result, all contiguous pages of heap memory following the zero-byte block are overwritten until either the source pointer reaches an invalid page boundary, the destination pointer reaches the end of heap memory, or another thread is dispatched and faults out trying to use corrupted heap memory.

The reason this vulnerability exists is because QUARTZ increments the specified string length (in order to make room for a null terminator) without checking for a potential overflow condition. The incremented value (now 0) is passed to LocalAlloc(), which succeeds, while the original value (FFFFFFFh) is given to memcpy() to copy the string data from the file image into the heap buffer.

For the sake of brevity, we have unfortunately omitted the details of the MIDI file format from this advisory, and will instead skip straight to the following example of a malicious MIDI:

```
4D 54 68 64 ; 'MThd' header chunk tag
00 00 00 06 ; size of header chunk data (6)
00 01 ; MIDI file version (1)
00 01 ; number of tracks (1)
65 49 ; pulses per quarter note (PPQN)

4D 54 72 6B ; 'MTrk' track chunk tag
00 00 00 10 ; size of track chunk data (16)
00 ; delta-time for event (0)
FF 02 ; non-MIDI event (Copyright)
8F FF FF FF 7F ; VLQ for text length (FFFFFFFh)
65 45 79 65 32 30 30 33 ; (start of malicious data)
```

There are many possible ways to exploit this overflow; the following is a sampling of instructions at which exceptions were observed in the aftermath of loading a malicious MIDI in Internet Explorer:

## SecurityFocus Bugtraq: EEYE: Windows MIDI Decoder (QUARTZ.DLL) Heap Corruption

```
CALL [EAX] ; we control EAX
CALL [EAX+C4h] ; we control EAX
CALL [ECX+0Ch] ; we control ECX
JMP [EAX+28h] ; we control EAX
MOV [ECX], EAX ; we control EAX, ECX
MOV [ESI], ECX ; we control ECX, ESI
```

Of particular interest are "unlink" sequences such as "MOV [ECX], EAX / MOV [EAX+4], ECX", which could be used to overwrite the unhandled exception filter in KERNEL32 during the first instruction, then cause an exception with the second (for instance, if EAX pointed somewhere into read-only memory, or if EAX was near a page boundary such that EAX+4..7 landed in an invalid memory region).

A second heap buffer overrun involving a 16-bit integer overflow and subsequent memory allocation was also discovered, but to save space we will only briefly mention it here. The number of tracks in the MThd chunk, a 16-bit field, is subjected to some arithmetic in order to determine the necessary size for an array of track data structures. In particular, the size of the block is calculated as:

$$(\text{number\_of\_tracks} * 24\text{h}) + 9\text{E}0\text{h}$$

However, the arithmetic is performed entirely in 16 bits, and as a result, setting the number of tracks to 1751 (6D7h) or greater will cause an insufficiently small heap block to be allocated. This vulnerability can be leveraged to overwrite DWORDs in the heap at specific intervals with arbitrary data. Note that Windows 2003 is not susceptible to this vulnerability, as it contained a check to ensure that the number of tracks is never greater than the exact highest value safe for the 16-bit arithmetic.

### Vendor Status:

Microsoft was contacted on April 16, 2003, and has released a patch for this vulnerability. The patch is available at:

<http://www.microsoft.com/technet/security/bulletin/MS03-030.asp>

This vulnerability has been assigned the CVE identifier CAN-2003-0346.

### Credit:

Derek Soeder – eEye Digital Security

### Greetin's:

6Ds; TJB, JC, MC, JAG, AH, JRJ, SMJ, JM, KP; Uma; and finally, Trust, when it's not spelled with a \$.

### Copyright (c) 1998–2003 eEye Digital Security

Permission is hereby granted for the redistribution of this alert electronically. It is not to be edited in any way without express consent of eEye. If you wish to reprint the whole or any part of this alert in any other medium excluding electronic medium, please email [alert@eEye.com](mailto:alert@eEye.com) for

## SecurityFocus Bugtraq: EEYE: Windows MIDI Decoder (QUARTZ.DLL) Heap Corruption

permission.

### Disclaimer

The information within this paper may change without notice. Use of this information constitutes acceptance for use in an AS IS condition. There are NO warranties with regard to this information. In no event shall the author be liable for any damages whatsoever arising out of or in connection with the use or spread of this information. Any use of this information is at the user's own risk.

### Feedback

Please send suggestions, updates, and comments to:

eEye Digital Security  
<http://www.eEye.com>  
[info@eEye.com](mailto:info@eEye.com)