Robert M. Koch
Final Project
Operating Systems
Dr. Jeonghwa Lee
Spring 2007

Sometimes, “Cursor” Refers to the User

The lure of pretty moving mouse cursors is too hard for some people to resist; at the least, this is the principle assumption behind one recent attack on machines running Microsoft Windows. The stack-overflow based attack exploits a total of three weaknesses – two in the Microsoft libraries and one in the MS compiler – to deliver its payload of arbitrary (usually malicious) code execution. In a statement released 2007-03-29 by the United States Computer Emergency Readiness Team, the threat is classified as “publicly available[6], and [...] actively exploited.”[1] The attack is potent on all Windows machines, including Vista, that have not applied the 2007-04-03 MS07-017 out-of-sequence patch. Unpatched machines are vulnerable in more ways than simply installing a crafted animated cursor. In fact, “embedding a malicious .ANI file within an HTML web page[...] allows the vulnerability to be exploited with minimal user interaction by simply coaxing a user to follow a hyperlink.”[2] This unauthorized loading of externally-defined content from sources of indeterminate trust is a “flaw” present in Explorer's browser code (and consequently having far-ranging effects throughout the Microsoft operating systems).

The attack gains control of the processor by exploiting flaws in Microsoft's processing of the RIFF file format – the file format of “.ANI” cursors. RIFF is an ASCII-tagged multimedia file format that has been the subject of security attacks before; in this case, an almost identical form of attack was discovered in January of 2005, exploiting the same section of faulty code.[3] Researchers are duly worried that Microsoft's inability to properly patch the flaw the first time around are an indication of future security concerns, despite Microsoft's “Security Development Life-cycle”.[4]

The structure of a RIFF file consists of a sequence of blocks of the following form: a four-byte ASCII tag, followed by the length of the block and the block data. Animated cursors contain a block tagged “anih”, containing the animated cursor's header. The size of this structure is typically 28 bytes; in

1 “http://www.kb.cert.org/vuls/id/191609”,
Vulnerability Note VU#191609: Microsoft Windows animated cursor stack buffer overflow,
US-CERT, 2007-04-23

2 “http://research.eeye.com/html/alerts/zeroday/20070328.html”,
eEye Digital Research Zero-Day Tracker #EEYEZD-20070328,
eEye Research, eEye Digital Security, 2007-03-28

3 “http://www.microsoft.com/technet/security/Bulletin/MS05-002.mspx”
Microsoft Security Bulletin MS05-002:
Vulnerability in Cursor and Icon Format Handling Could Allow Remote Code Execution (891711)
Microsoft Corporation, 2005-04-12

4 “http://www.computerworld.com/action/article.do?command=viewArticleBasic&articleId=9015899”
Researchers question Vista security after ANI exploit
Gregg Keiser, ComputerWorld Security, 2007-04-06

Microsoft's original code, this block size was assumed and never validated. The header structure was allocated on the stack, filled with the contents of the “anih” block *as specified in the file*. If the file contents indicate a block size longer than 28 bytes, the stack-stored header structure will be overflowed, overwriting other data on the stack – such as the caller's return address. Crafting a header block to overwrite the return address with a pointer into its data area allows the contents of the header block to be executed as code with the security permissions of the exploited process. The original discovery of this attack led to the patch mentioned above.[5]

Microsoft's changes to the affected code were only a partial fix, however, as is evidenced by the new attack. Examination of the exploit revealed its defense: “If at first you don't succeed, try; try again.” While the original patch prevents the normal “anih” header block from overflowing the buffer (by checking that the block is of the expected size), no check is made that the first header is the only such header. After validating the first header encountered, the originally vulnerable code is invoked to process all subsequent blocks – including the extra “anih” headers embedded in recently crafted exploits. By placing a second malformed header block in the file, the crafted file sidesteps the simple-minded security of Microsoft's original patch, and invokes the original, vulnerable code, with precisely the same effect.[6]

The third and final flaw is in Microsoft's compiler. However, upon examination of the flaw, it cannot be condemend. The Microsoft compiler usually emits a stack canary to prevent just the sort of buffer overflow this attack exploits. However, the RIFF-handling code does not have a buffer on the stack. Instead, a structure is stored on the stack, and later *used* as a buffer. Under default settings, the MS compiler will not recognize this misuse of a non-buffer memory area, and will not place a canary on the stack. I am under the (unverifiable) impression that few other compilers will perform this check, either. As a result, the fault lies not so much with the compiler as with the offending code. Because the exploit can affect Windows Vista machines that are not patched up to 2007-04-03, this exploit will currently affect any new computer purchased as of the date of writing; in order to obtain the patch mentioned above, you must (through some means) use Windows Update.

5 “http://research.eeye.com/html/advisories/published/AD20050111.html”
Published Advisory #AD20050111: Windows ANI File Parsing Buffer Overflow
eEye Research, eEye Digital Security, 2005-01-11

6 “http://milw0rm.com/exploits/3634”
“http://www.milw0rm.com/sploits/04012007-Animated_Cursor_Exploit.zip”
Windows Animated Cursor Handling Exploit (0day)
“Credit: milw0rm,metasploit, SkyLined, http://doctus.net/”