Not every first-chance exception is a security vulnerability

devblogs.microsoft.com/oldnewthing/20071218-00

December 18, 2007

Raymond Chen

In the category of dubious vulnerability, I submit the following (paraphrased) report:

> If I call the `FormatMessage` function, I can cause a buffer overflow exception if I provide an insertion that is more than 2000 characters long.

The `FormatMessage` function in Windows NT, 2000 and XP used the dynamically expanding buffer technique to allocate memory for the resulting message. If the resulting string was more than one page in length (4KB on an x86 system), there was an exception thrown when the `FormatMessage` function tried to write to the 4096th byte of the buffer. This looks like a buffer overflow, and in a sense it is, but it's a controlled overflow (the bytes beyond the end of the buffer are under the program's control), the exception is entirely expected, and it is correctly handled. Using intentionally invalid pages to trigger just-in-time memory commit is a rare technique, so it's not surprising that people aren't familiar with it. In fact, to avoid these sorts of false alarm security vulnerability reports, the kernel folks rewrote the `FormatMessage` function in Windows Vista so it doesn't use this technique any more. It's an odd Catch-22. You remove something that is frequently mistaken for a security vulnerability so that people stop mistakenly reporting it, but the fact that you remove it only confirms in the mind of the people who filed the false alarms that they found something for real!

(For further reading, may I recommend this blog entry from Larry Osterman.)

Raymond Chen

Follow