Sometimes the bug isn't apparent until late in the game

devblogs.microsoft.com/oldnewthing/20040910-00

September 10, 2004

Raymond Chen

I didn't debug it personally, but I know the people who did. During Windows XP development, a bug arrived on a computer game that crashed only after you got to one of the higher levels.

After many saved and restored games, the problem was finally identified.

The program does its video work in an offscreen buffer and transfers it to the screen when it's done. When it draws text with a shadow, it first draws the text in black, offset down one and right one pixel, then draws it again in the foreground color.

So far so good.

Except that it didn't check whether moving down and right one pixel was going to go beyond the end of the screen buffer.

That's why it took until one of the higher levels before the bug manifested itself. Not until then did you accomplish a mission whose name contained a lowercase letter with a descender! Shifting the descender down one pixel caused the bottom row of pixels in the character to extend past the video buffer and start corrupting memory.

Once the problem was identified, fixing it was comparatively easy. The application compatibility team has a bag of tricks, and one of them is called "HeapPadAllocation". This particular compatibility fix adds padding to every heap allocation so that when a program overruns a heap buffer, all that gets corrupted is the padding. Enable that fix for the bad program (specifying the amount of padding necessary, in this case, one row's worth of pixels), and run through the game again. No crash this time.

What made this interesting to me was that you had to play the game for **hours** before the bug finally surfaced.

Raymond Chen

Follow