

# Windows Update Kills VB6 Apps

By [John Browne](#)

Not going to bury this lead<sup>1</sup>: Microsoft's latest update has stopped some VB6 apps from working.

Yeah. That day we always knew was going to come *some day*? Well, turns out *some day* was last Tuesday: August 13, 2019.

The day that VB6 died.

Not to mention VBA and VBScript.



## What happened, exactly?

Not sure who to point at here, really. Microsoft, in looking for a solution to the [BlueKeep vulnerability](#) (itself a significant attack vector on older Windows OS instances), found at least [two RDP vulnerabilities](#) going back to Windows 7--and all three are worms, which is extra scary. The patch for those--released on Tuesday--kills some RPC code in all flavors of classic Visual Basic, including VBA macros, VBScript, and, of course, good old still-wildly-popular Visual Basic 6.0 (and previous versions). Affected applications, according to Microsoft, may just quit with an error message. [You can read about it here on ZDNet](#).

So you, or your IT staff, are faced with the dilemma: protect the entire network (by updating Windows) and kill critical apps, or keep apps alive and risk a complete network infection.

Risk the network or kill the apps?



## What's the deal with Visual Basic?

Like Cream of Wheat and '57 Chevys, VB comes from a more innocent time. A time when we pictured desktop computers as a way to revolutionize business, not crime. VB revolutionized a lot of organizations in a good way, But that was a long time ago, and in the meantime cyber crime and maliciousness have become high art. Nation states are engaged in a new kind of warfare: cyber war. And the side effect of this is the release of those cyber weapons into the hands of criminals. And one of the worst actors in this new kind of war are some old friends from Europe:



VB isn't bad, per se. It's just old. The world of IT has moved forward while VB remained static, stuck in time. And like that '57 Chevy with its dated drum brakes, zero seat belts, bias-ply tires, and steel dashboard, it's inherently more dangerous than any contemporary model.

## Ain't going to happen here

In [The Fifth Risk](#), author Michael Lewis explains how people who live in "Tornado Alley" believe, in spite of all the evidence to the contrary, that they will never be hit by a tornado. As a result of this denial-ism, they don't take cover when a tornado is coming. Then they are somewhat surprised when their house blows away, or they learn the secret of the afterlife. Unfortunately, too many IT professionals have the same attitude with respect to application health and security. No one plans to get hacked; no one plans on their mission-critical application just stopping after a critical OS update.

Until, of course, it does.

## What can be done?

Things change fast. Just two weeks ago we were alerted that [Microsoft is sunsetting support for VBScript in IE11](#). And now the question is whether you want to [risk a BlueKeep attack or break your line of business apps](#). Then again, just today (Aug 19), came word that Microsoft has already begun addressing the problem that VB-related apps might experience with RPC failures after the latest update.

The writing is on the wall. Will anyone read it?

VB's time is over. VB6, VBScript, VBA. Sure, Microsoft will "support" vbrun for the foreseeable future<sup>2</sup>, but that's no guarantee that apps will continue to work, or that they won't be a vector of attack. It's clear that if Microsoft has to choose between security and legacy app support, they are going with security. Their customers may get pretty steamed if VB-enabled apps quit working because of an update, but not nearly as steamed as they will if an un-blocked worm infects their entire network.

Truly your only real solution is to [eliminate VB](#). Some apps you can just sunset, some you can replace with an off-the-shelf commercial app, and some you can migrate or rewrite. Obviously we're big fans of the migration option, [when it makes sense](#). Surprisingly, it often does, especially for larger, complex, apps that continue to be updated and that are mission-critical to the organization.

There are a lot of bad actors out there. Don't let Visual Basic be the unlocked door that let's them in.

<sup>1</sup> I know, everybody is spelling "lede" these days, but as an ex-newspaperman (briefly, ok?) I refuse to go along [bites cigar stub angrily].

<sup>2</sup> Microsoft has committed to support the VB runtime for some unknown time in the future. [Read more here](#). Note the careful use of the word "core" and that the VB6 IDE hasn't been supported since 2008. Third party controls and non-core stuff like RPC may stop working at any time without warning.