Responses to various ideas on how to get people to stop using that leaked build

devblogs.microsoft.com/oldnewthing/20170323-00

March 23, 2017

Raymond Chen

Some time ago, I told the story of how we used psychology to get people to stop using a leaked build and upgrade to a newer build. People had suggestsions for other ways this could have been done.

One suggestion to avoid the problem in the first place was to have internal builds refuse to activate from an IP address that is not assigned ot Microsoft. This would stop people from installing and activating a build, but of course you can use a build without activating it, and by resetting the activation timer, you can extend the lifetime of an unactivated system to quite a few months. That's quite a few months of potentially messing up the Internet.

So you might say that internal builds should simply refuse to run unless activated. That would prevent these systems from getting off the ground. Unfortunately, it would also prevent our software development partners from installing and activating the builds. Hardware partners wouldn't be able to develop new drivers for the upcoming version of Windows to take advantage of whatever new features are being added. Software partners wouldn't be able to send the Windows team feedback on a proposed new feature, and they wouldn't be able to have an updated version of their software ready to go on release day.

Another suggestion was to have Windows perform an online check at startup to determine whether the build has been declared "fatally broken", and if so, refuse to boot or at least time-limit the build so the user can upgrade to a non-broken version before time runs out.

Well, first of all, that solution would have to be applied retroactively. Somebody ahead of time would have to convince management to devote engineering effort (as well as dedicate bandwidth in our servicing stack) to handle the case that a build at some point in the future needs to be blocked. Seeing as nothing this horrific has happened before in the thirty years of Windows, it might be a hard sell.

Besides, even if such a feature were implemented, you could imagine the PR disaster if it were discovered or deployed: "ZOMG Windows has a remote kill switch!"

And then there were the suggestions to do something that Microsoft was already doing.

"Maybe it's the time to put up somewhere a disclaimer that a build not intended to be public is more likely to corrupt the system, malfunction or not even boot, and it's better to wait for a proper build from Microsoft to the public."

Yup, that's what Microsoft does. You see that statement any time somebody asks for a comment about a leaked build. It goes roughly, "We don't recommend installing unofficial builds because who knows what's in them."

Another superfluous suggestion was to include a warning in all pre-release builds saying that, you know, it's a pre-release build and anything can happen.

Yup, that warning already exists for pre-release builds. But the warning has been around for so long that people have gotten inured to it and don't pay it much heed. It has sort of become the boy who cried wolf: Every single pre-release build comes with this warning, and every single pre-release build manages not to destroy your computer.

Another suggestion was to announce that the build contains a potentially fatal flaw that could cripple your computer and disrupt any networks it is connected to. "We don't care how you acquired this build. We simply do not want you bricking your computer."

People would probably say, "Wow, Microsoft is really worried about this build. There must be some secret feature in that build that they don't want us looking for, and they're using this lame excuse to get us to stop using that build. Hey everybody, make sure you keep your ISOs because there's gold hiding in there somewhere!"

And they'd be right, too. Because the feature that had the fatal flaw was not yet publically announced. If you're installing leaked builds, it's because you're the sort of person who wants to get at non-public information, and unannounced features are the best kind of non-public information. This is basically an announcement that says, "Hey everybody, I strongly recommend that you crawl through every single option dialog you can find, because buried somewhere is going to be a new checkbox that reveals an unannounced feature. Happy hunting!"

The Windows team ultimately decided to solve the underlying problem of people itching for builds in a different way: Make the internal builds public, via the Windows Insiders Program.

Bonus chatter: Ultimately, most of the proposals ended up just over-engineering the solution. If you can solve a problem in ten minutes with psychology, what's the point of spending hundreds of hours trying to solve it with technology?

Raymond Chen

Follow