On the relationship between specific cases and general cases

devblogs.microsoft.com/oldnewthing/20080425-00

April 25, 2008

Raymond Chen

One of the frustrations I have is people who don't see the relationship between specific cases and general cases.

> **Statement**: In general, you can't do X.
>
> **Refutation**: You're wrong. Y can do X.

Example: The statement "In general, you can't trust the return address" comes with the refutation "You're wrong. .NET trusts the return address." (Sometimes, it's not so much a refutation as just the assumption that the .NET folks are somehow stupid for not understanding this general rule.) If I use this same argument in a different context, the fallacy becomes more clear.

> **Statement**: In general, you can't sell cigarettes to people who appear to be underage without checking their ID.
>
> **Refutation**: You're wrong. A bar can sell cigarettes to anyone.

The flawed refutation introduces new assumptions, namely that you're in the special case of being in a bar. But the bar has its own restrictions, namely that you have to be at least 21 years old to enter. Therefore, they don't need to check your ID because they can assume that everybody in the bar is already at least 21 years old. Similarly, .NET can trust the return address since it doesn't let programs modify arbitrary memory. Memory access must first pass the verifier before the CLR will run it. (Of course, you can access arbitrary memory by using unsafe code or p/invoke, but you have to be granted full trust in order to do it, so you haven't actually accomplished anything.) In the general case, however, a called function does not have control over what other code is running in the same process.

A related source of frustration comes from people who treat all statements as absolute metaphysical truths, even if they are clearly generalizations once you accept that people have free will. I've gotten in the habit of inserting weasel words like *typically* or *generally*, but not

everybody remembers to do this in the comments, exposing themselves to <u>nitpicky responses from other commenters</u>. Welcome to my world.

<u>Raymond Chen</u>

Follow