guardian.co.uk

Software makers should take responsibility

Bruce Schneier
The Guardian, Thursday 17 July 2008

A larger | smaller

Cars are well designed because manufacturers face liabilities if they make mistakes (Photograph: David Levene)

A recent study of Internet browsers worldwide discovered that over half – 52% – of Internet Explorer users weren't using the current version of the software. For other browsers the numbers were better, but not much: 17% of Firefox users, 35% of Safari users, and 44% of Opera users were using an old version.

This is particularly important because browsers are an increasingly common vector for internet attacks, and old versions of browsers don't have all their security patches up to date. They're open to attack through vulnerabilities the vendors have already fixed.

Security professionals are quick to blame users who don't use the latest update and install every patch. "Keeping up is critical for security," they say, and "if someone doesn't update their system, it's their own fault that they get hacked." This sounds a lot like blaming the victim: "He should have known not to walk down that deserted street; it's his own fault he was mugged." Of course the victim could have –and quite possibly should have – taken further precautions, but the real blame lies elsewhere.

It's not as if patching is easy. Even in a corporate setting, systems administrators have trouble keeping up with the neverending flow of software patches. There could easily be dozens per week across all operating systems and applications, and far too often they break things. Microsoft's Automatic Update feature has automated the process, but that's the exception. Patching is triage, and administrators are constantly prioritizing it along with everything else they're doing.

It's the system that's broken. There's no other industry where shoddy products are sold to a public that expects regular problems, and where consumers are the ones who have to learn how to fix them. If an automobile manufacturer has a problem with a car and issues a recall notice, it's a rare occurrence and a big deal – and you can take you car in and get it fixed for free. Computers are the only mass-market consumer item that

pushes this burden onto the consumer, requiring him to have a high level of technical sophistication just to survive.

It doesn't have to be this way. It is possible to write quality software. It is possible to sell software products that work properly, and don't need to be constantly patched. The problem is that it's expensive and time consuming. Software vendors won't do it, of course, because the marketplace won't reward it.

The key to fixing this is software liabilities. Computers are also the only mass-market consumer item where the vendors accept no liability for faults. The reason automobiles are so well designed is that manufacturers face liabilities if they screw up. A lack of software liability is effectively a vast government subsidy of the computer industry. It allows them to produce more products faster, with less concern about safety, security, and quality.

Last summer, the House of Lords Science and Technology Committee issued a report on "Personal Internet Security." I was invited to give testimony for that report, and one of my recommendations was that software vendors be held liable when they are at fault. Their final report included that recommendation. The government rejected the recommendations in that report last autumn, and last week the committee issued a report on their follow-up inquiry, which still recommends software liabilities.

Good for them.

I'm not implying that liabilities are easy, or that all the liability for security vulnerabilities should fall on the vendor. But the courts are good at partial liability. Any automobile liability suit has many potential responsible parties: the car, the driver, the road, the weather, possibly another driver and another car, and so on. Similarly, a computer failure has several parties who may be partially responsible: the software vendor, the computer vendor, the network vendor, the user, possibly another hacker, and so on. But we're never going to get there until we start. Software liability is the market force that will incentivise companies to improve their software quality – and everyone's security.

• Bruce Schneier is a security technologist and author
schneier.com/blog