virus BULLETIN **COMMENT**

'When will a silver bullet come along that makes computers work as well as toasters?'

John Aycock
University of Calgary

YOUR COMPUTER IS TOAST

Earlier this year, I was asked a question: how do you stop viruses and worms altogether? Completely. Full stop. No more viruses and worms any place. I had to think about this for a moment. It's a very interesting question, and my answer was somewhat surreal: toasters.

I love my toaster. From a user interface point of view, it's brilliant. Even my youngest child can understand how to operate it: it has few controls, and it's easy to form a mental model of how it operates. What's amazing is that – apart from the odd piece of burnt raisin bread – it just works. It's never required an update or a patch. And my toaster has never been hit by a virus or worm, nor has spyware ever absconded with my toast preferences.

The same claims cannot be made for any computer I've connected to a network, no matter what the architecture or operating system. Given how much our society relies upon computers, you would hope that the computers running the power grid were more reliable than the toasters plugged in to it. Yet it's no secret that our computers are breeding grounds for all kinds of malicious software. With mobile phone worms spreading in the wild, virus-like behaviour being exhibited by Sims 2 hacks, and proof-of-concept PDF file worms, is there any logical limit to the places where malware can thrive?

In *Profiles of the Future*, Arthur C. Clarke famously wrote that 'any sufficiently advanced technology is indistinguishable from magic.' I have a corollary to this, which I'll modestly call Aycock's law: any sufficiently advanced technology is susceptible to viruses.

Already we need anti-virus software on our desktops, laptops, and mobile phones; anti-virus for game machines probably isn't far off, either. When will a silver bullet come along that makes computers work as well as toasters?

One of the problems is that computer scientists like to generalize. A general algorithm is cleverer than a less general one; a general design is better than a more specific one. Our computers are general-purpose, and we interconnect them in the hope that they can talk to everything else in some general way. Call me a Luddite, but maybe I don't need my wristwatch chatting with my running shoes via Bluetooth. We don't require generality in every situation, and in some cases we are better off without it. For example, it's hard to verify the security of a web browser that's general enough to be extensible. The plug-ins that extend the browser aren't known until they run, which provides a lot of leeway for malware to exploit through software engineering and social engineering.

Computer memory is generalized, as something which can hold code and data, rather than code *or* data. This fact has been exploited by high-profile worms with buffer overflow attacks for over 16 years now, with the Internet worm in 1988, Slammer in 2003.

Worms, of course, can't spread across communication channels that don't exist. My toaster is not general enough to communicate with the blender beside it. However, the Internet has proven to be a general medium over which disparate devices can talk to one another. You can even buy Internet-enabled refrigerators, presumably to send spam as well as keep it chilled.

At the opposite end of the spectrum lie domain-specific systems. These are tailored to one narrow area, like SQL being used to describe database queries instead of using a general-purpose language like C. Toasters are domain-specific systems too, tailored to the domain of making bread brown. Domain-specific systems have two important properties relating to malware: their functionality is limited, and their normal behaviour is well understood. Suitably limited functionality can deny would-be malware authors from expressing their progeny, and well-understood behaviour allows extremely accurate anti-virus heuristics and emulation to be developed.

That's it. Design computers to do one thing, and only one thing, well. Resist the urge to have them communicate with all their neighbours within earshot. By limiting generality and unnecessary communication channels, hopefully Aycock's law is one that is made to be broken. Toast, anyone?