SIMSON GARFINKEL

PROOF OF CONCEPT

In the military, most new weapons systems go through an evaluation, or proof-of-concept, phase. These are not full-power tests, but baby steps to show that key technology should work as advertised.

This sort of testing takes place not only with missiles and bombs but also with the cybernetic implements of information warfare. Indeed, any group that is developing tools to disrupt an adversary's information systems would be downright irresponsible if it did not conduct proof-of-concept demonstrations as part of its R&D process. These tests would not cause great harm: instead, they would be designed to whet the appetite of officials higher up the command chain.

And what would a proof-of-concept demonstration for an information warfare weapon look like? Possibly a lot like the computer virus attacks the Internet has experienced in recent years. I suspect that some of these electronic attacks were actually the results of deliberate tests for a future attack that could have truly dire consequences.

To understand my alarm, you need to understand the anatomy of computer viruses and their cousins, worms. Most of these hostile programs have three parts. The first, the "exploit," is the technique the virus or worm uses to break into systems. Most exploits take advantage of a known security flaw—for example, the classic "buffer overflow," in which an excess of incoming data corrupts the information already stored in memory. The second part, the "propagation engine," is the code that targets computers for attack. And the third, the "payload," does the actual damage.

Viewed through this morphology, the major worms that have disabled computers on the Internet—Code Red, Nimda, Klez, and, most recently, Slammer—share a disturbing similarity. Each one employed a novel—and extremely effective—propagation engine. But for exploits, all these worms have used security vulnerabilities that had been previously identified. And as for the payload: all were duds. Even though each gained so-called administrative privileges to alter the systems they infected, none used its privileges to cause mayhem.

Sure, they did some harm. But in nearly every case, the damage was caused by the propagation itself—as if a burglar systematically were to break windows, enter every house on the block, and steal nothing. An actual payload could scramble financial data, erase operating systems, and ruin motherboards by wiping out the contents of their programmable chips.

This pattern has been repeated so many times that I believe at least some of these worms are in fact elaborate proof-of-concept tests, created by a clandestine information-warfare lab. This is more plausible than you might first think. Many of the computer viruses written over the past 20 years have been the work of a small group of teenagers and young adults engaged in competition with their friends and various antivirus companies. The virus creators quickly learned that any bozo could write a program that could erase other people's hard drives. By the mid-1990s the rogue programmers had elevated the game: scoring points among fellow hackers required clever propagation techniques and tricks for outrunning the antivirus shops. Law enforcement agencies know that hackers from the 1990s are now selling their services to organized crime and terrorists; why not the virus writers?

Remember: not a single worm or virus that we have seen in the wild—not one—has employed a novel exploit. That's not surprising. Unknown exploits are far too valuable to reveal in public proof-of-concept testing. Likewise, no worm has deployed a payload that caused significant damage. Why set the death ray to kill, when all you are trying to do is prove that the thing shoots?

Fred Cohen, a computer security researcher who has been studying malicious programs for two decades (and who is credited with coining the term virus), doubts that the worms we have seen are the work of government-run information-warfare labs. But he concedes that smaller labs or solo operators—possibly renegade operations in China or independent outfits looking to sell their services—might have released weakened versions of their worms into the wild as tests. The U.S. military set a sort of precedent in the 1960s when it tested "simulant" germ-warfare agents in the New York City subway and off the coast of San Francisco. And it is widely believed that the original Microsoft Word macro virus, auspiciously named Concept, was written as a proof-of-concept test by a programmer at Microsoft and was inadvertently released on discs at a developers' conference.

No computer virus or worm that has been released on the Internet has deployed a truly damaging payload: why set the death ray to kill, when all you are trying to do is prove that the thing shoots?

Speculation about the origin of these worms has taken on more relevance since last summer, when President Bush reportedly signed an order directing the U.S. government to develop guidelines for launching a cyberattack against enemy computer networks. Before that order, my friends in the military were saying that the United States would never consider conducting an offensive information-warfare campaign: as the nation with the most computers, we have the most to lose. But now our government is in the process of legitimizing cybernetic warfare.

I'm worried. Today's lame computer worms, even with well-known exploits and dummy payloads, have shut down corporate and government networks. A determined enemy would target a new exploit with a really nasty payload. The proof that time could be a hundred billion dollars in damage.