Security Cosmology:

Moving from Big Bang to Worlds in Collusion

If there was a "big bang" in the computer security universe, it occurred 15 years ago when the first Internet worm was unleashed (for a perspective on this worm, which some also call the Morris worm, see page 35). Many computer security problems had been recognized and seriously addressed more than 15 years prior to that incident, as documented in the 1970 Ware report (reissued in an unclassified version in 1979)[1] and the 1972 Anderson report.[2] In 1987, the Christma Exec worm caused a less-publicized email storm that denied service to many users of IBM's internal network. But the Internet worm arguably has had more influence on computer and network security than any other single event before or since.

The worm dramatically revealed Internet-connected systems' vulnerability to the several kinds of attacks it incorporated; particularly, how to exploit an unchecked buffer overflow to break into a system. Even today, these kinds of vulnerabilities remain the primary ones that widespread attacks exploit. The worm triggered the creation of the Computer Emergency Response Team (CERT) at the Software Engineering Institute and, subsequently, international incident response teams. It also caused the computer security research and development community to examine how effective (or ineffective) then-current approaches were when dealing with this class of attack.

From a technical standpoint, this is old news. Even in 1988, there were, for example, well-known ways to prevent buffer overflows in new software, to check for them in existing software, and to distribute corrected versions of flawed routines. Why do the dominant software producers continue to distribute software with these vulnerabilities?

"Every day in every way my job gets easier and easier," said a friend of mine recently about his for-hire efforts to penetrate systems as a way to test their security measures. Vendors, he said, tend to bundle software components together, so users' systems often include components they don't know about. Users, on the other hand, display a continuing appetite for "dancing pigs"—demanding animated applications that push vendors to squeeze the last ounce of performance from their systems.

Configuring and squeezing systems in this way often translates to reducing the strength of internal checks and separation mechanisms that could limit the potential damage that attacks inflict against residual flaws. The result? My friend rarely has a problem finding a way into a system he is asked to attack.

It's a matter of trade-offs, costs, and incentives that leaves our society spending pounds on cure rather than ounces on prevention. Too often, the costs of prevention have not benefited the individuals or companies who have initially borne them.

Scapegoating is not the point. Rather, we must recognize that solving the security and privacy problems we face requires considering human behavior, economics, and the influence of laws and regulations as well as technology. Both researchers and developers must take a broad view of the problem and potential solutions, encompassing all of the relevant disciplines.

It's not necessary (or possible!) for each of us to become expert in all of these fields. We must reach out to specialists in human–computer interfaces, economics, and law, be willing to explain technological problems to them in terms they can understand, and be ready to consider their assessments and contributions. This is not easy—colleagues in these fields have their own concerns and are as busy as we are. But the increasing publicity that widespread attacks generate provides us with an opportunity to enlist outside assistance that we must not neglect. We must make these different worlds collaborate, not collide. □

Carl E. Landwehr
Associate Editor in Chief

References

1. W. Ware (ed.), *Security Controls for Computer Systems*, Defense Science Board Task Force on Computer Security, R-609-1, RAND, 1979; www.rand.org/publications/R/R609.1/R609.1.html.
2. J.P. Anderson, *Computer Security Technology Planning Study*, ESD-TR-73-51, ESD/AFSC, Hanscom AFB, Bedford, MA, 1972, NTIS AD-758 206; http://seclab.cs.ucdavis.edu/projects/history/papers/ande72.pdf.