Goals for Computer Security Education

Cynthia E. Irvine
Computer Science Department
Naval Postgraduate School
irvine@cs.nps.navy.mil

Until recently, most of those involved in research, development, and operation of secure computing systems have been either autodidacts or individually mentored by people already working in the field. Today's practitioners learned computer security as it was growing up around them. Security concerns have created an increased demand for computer security professionals. Students want to learn about computer security and potential employers want graduates who can go to work solving their problems.

We, the members of the computer security community, must be responsible for producing the next generation of computer security experts. The objective of this panel is to present and discuss the opinions of people who hire computer science graduates to work on computer security problems. Thus, the panel seeks not to have computer security educators tell the audience what they are teaching, but to have employers tell us what needs to be taught.

Questions to be addressed include:

- What should students be taught to equip them to become productive members of the computer security community?
- Can one emerge from an undergraduate or graduate program ready to go to work as a computer security specialist?
- How up-to-the-minute should graduates be with respect to current technology?
- Is there a fundamental body of knowledge which should be taught universally? Is it based on results from the literature, requirements and criteria, or is it threat driven?
- What suggestions for material to be included in textbooks can industry provide?
- Should students of computer security learn what "hackers" do and try out "controlled hacking" as part of their education?
- How much should students learn about specific systems such as UNIX or commercially available PC operating systems?
- Computer security touches on a broad range of topics in computer science and engineering. Should students know a little about a lot of areas or should their education be focussed in selected areas?
- Should graduates be able to differentiate real security from snake oil? How can this be taught?
- Besides $$$, how can industry help with computer security education?

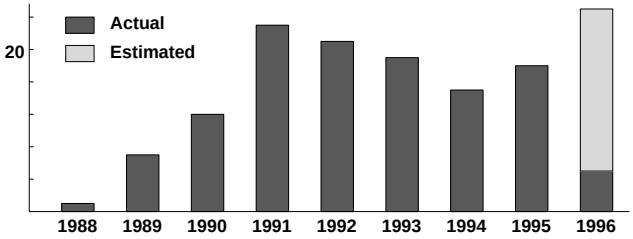

Figure 1: Annual Computer Emergency Response Team Advisories

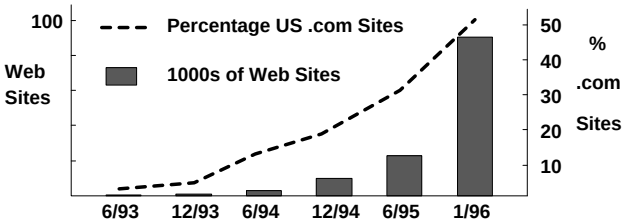

Figure 2: Growth of World Wide Web Sites

Two quite different approaches to computer security education appear emerge from these questions.

First, security in computer systems can be treated as an *ad hoc* set of functions which are modified as vulnerabilities are identified. Examination of Figure 1 shows that, following an initial steep rise, the number of Computer Emergency Response Team (CERT) Advisories per year has remained reasonably constant [3]. Yet as Figure 2 illustrates for the World Wide Web, interconnectivity continues to grow [5]. Consequently we will need a few people to find system flaws and describe the fixes, and a much larger number of people to plug the holes in all of the individual systems. To be proficient at identifying and repairing system vulnerabilities, students will need to become well versed in the intricacies of specific system implementations.

Some classroom exercises can focus on case studies of vulnerabilities and flaws in existing systems that have been exploited by misfeasors and what security experts did to thwart future use of those flaws. Simulations allowing students test penetrations and remedies could be useful. Students could study operating systems and applications to learn how flawed systems have been constructed, choices that lead to the introduction of flaws, and where security blunders often occur. Software engineering could contrast academic concepts with the less than ideal disciplines used in the field. Because the real systems being used in government and commerce are proprietary, schools would need to establish non-disclosure agreements with vendors so that thorough investigations would be possible. Research could result in papers analyzing specific incidents of flaw exploitation such as the Internet Worm of 1988 [10]; categorization of flaws in various systems, e.g. [7]; identification of flaws in specific products, e.g. [4]; and techniques for remedying these flaws.

The second approach is to build security into our systems *ab initio* using an engineering-oriented approach based on fundamental principles. Here it is assumed that one cannot know *a posteriori* whether a system is secure. Students would learn fundamentals such as the Reference Monitor Concept [1] and the safety problem [6]. They would learn how security policies and their mathematical formulations, e.g. [2, 8] provide a blueprint for constructing a system intended to provide policy enforcement. Students could start by seeing how systems with assured security properties were built in the past, e.g., [9] and [11]. Tracing the application and extension of basic security principles through the evolving technology to the hottest developments emerging from industry, the educational program would prepare students to understand how these concepts might be applied in the future. Needless to say hefty doses of software engineering, operating systems, as well as a rigorous set of core computer science courses would be needed.

Many topics are likely to be common to both approaches: an introduction to the concepts underlying modern cryptography; the need for sound cryptographic protocols, and the challenges of key management and key distribution; why auditing as well as identification and authentication are needed and how they are accomplished; the need for physical and personnel security; etc. The use of newly evolving tools and methods from both academic programs and industry can provide interesting challenges to students.

The panelists come from different sections of the computer security community. Leslie Chalmers represents Wells Fargo Bank, an enterprise where lack of security controls could have disastrous consequences in terms of financial impact and loss of consumer confidence. As Deputy Director of the National Computer Security Center, Stephen F. Barnett represents an organization well known for its contributions to computer security education, research and development. Jim Schindler is from Hewlett Packard where he manages the development of new security products and brings to the panel the perspective of an enterprise attempting to meet marketplace demands for infinite MIPS, perfect security, and zero cost. The next panelist is Roger Schell, of Novell, Inc., who has been an educator, worked in the Department of Defense, and now leads a group developing new commercial security technology. Finally, Karl Levitt, Professor of Computer Science at the University of California at Davis, will serve on the panel as a representative of the educational community.

References

[1] James P. Anderson. Computer Security Technology Planning Study. Vol. I, AD-758206. Vol. II, AD-772806 ESD-TR-73-51, Air Force Electronic Systems Division, Hanscom AFB, Bedford, MA, 1972. AD-758 206, ESD/AFSC.

[2] David E. Bell and Leonard LaPadula. Secure Computer Systems: Mathematical Foundations and Model. Technical Report M74-244, MITRE Corp., Bedford, MA, 1973.

[3] CERT-Coordination-Center. Cert advisories. Software Engineering Institute, Carnegie Mellon University,URL ftp://info.cert.org/pub/cert_advisories/. Totals through March 10, 1996.

[4] Drew Dean and Dan S. Wallach. Security Flaws in the HotJava Web Browser. In *Proceedings 1996 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, 1996.

[5] Matthew Gray. Measuring the Growth of the Web, June 1993 to June 1995. net.Genesis Corporation, http://www.netgen.com/info/growth.html, 1996.

[6] M. Harrison, W. Ruzzo, and J. Ullman. Protection in Operating Systems. *Communications of the A.C.M.*, 19(8):461–471, 1976.

[7] Carl E. Landwehr, Alan R. Bull, John P. McDermott, and William S. Choi. A Taxonomy of Computer Program Security Flaws, with Examples. Technical Report NRL/FR/5542-93-9591, Naval Research Laboratory, Washington, DC, November 1993.

[8] T. Levin, S. Padilla, and C. Irvine. A Formal Model for UNIX SETUID. In *Proceedings 1989 IEEE Symposium on Security and Privacy*, pages 73–83, Oakland, 1989. IEEE Computer Society Press.

[9] P. Neumann, R.S. Boyer, R. J. Feiertag, K. N. Levitt, and L. Robinson. A Provably Secure Operating System: The System, Its Applications and Proofs. Technical Report CSL-116, SRI International, Menlo Park, CA, May 1980.

[10] Eugene H. Spafford. Crisis and Aftermath. *Communications of the A.C.M.*, 32(6):678–687, 1989.

[11] Clark Weissman. BLACKER: Security for the DDN Examples of A1 Security Engineering Trades. In *Proceedings 1992 IEEE Symposium on Research in Security and Privacy*, pages 286–291. IEEE Computer Society Press, 1992.