Guest Editorial

Matthew A. Bishop University of California at Davis

An open system is one that can be extended or adapted by users writing their own commands or altering parts of programs traditionally seen as part of the operating system, such as command interpreters. The ability of users to modify these systems so extensively creates a tension with the needs of security; specifically, there is an apparent conflict between ease of change and protection boundaries. If a user wants more rights, why not simply write a program that uses those rights and replace the relevant parts of the security mechanism with that program?

As readers know, this rarely works because the enforcement mechanisms are themselves protected from modification. (The major exception to this rule is personal computers.) Determining what the security mechanisms should allow (and prevent) requires a very clear understanding of the security policy desired; protecting those mechanisms adequately and, through them, the system and its users as well, requires a trustworthy implementation of both the security mechanisms and those mechanisms' protections. Articles in this special issue touch upon these themes.

The article by Ware addresses policy concerns in computer networks, as well as the security considerations underlying them. It illustrates the many facets of policy design as well as the nontechnical constraints that must be met.

Issues of trust in a network abound; and the article by Klein, Beth, and Yahalom explores trust-based navigation in distributed systems with inhomogeneous trust relationships. The authors address issues raised by the way open systems are used in an internetworked environment; this issue is relevant to security and integrity in open systems.

The article by Krajewski, Chipchak, Chadorow, and Trostle describes work augmenting the Kerberos authentication system by use of smart cards to circumvent the problem of decrypted Kerberos keys remaining on a workstation. When many users use such a workstation, the existence of such keys raises security threats and this article presents an interesting resolution of this problem.

The use of security constraints to scan a system for potential security problems is addressed by Heydon and Tygar. In their article, they describe a system for specifying and checking such constraints and then apply it to a UNIX system to search for security problems. Their results show that careful specification of security requirements may help detect some of the more common security problems.

This issue concludes with an article by LaPadula that presents a formal model of a trusted computer system and examines UNIX System V in light of that model. Its point, that formal modeling can be used to analyze fairly realistic system representations, is worth considering, given the abstractness of most such models.

The following people served as referees for this special issue, for which they have my deepest thanks: Ed Amoroso, Rebecca Bace, Robert Baldwin, David Balenson, David Bell, Steve Bellovin, Tom Berson, Klaus Brunnstein,Vint Cerf, Bill Cheswick, Tina Darmohray, Jeremy Epstein, Dan Farmer, Deborah Frincke, L. Todd Heberlein, Russ Housley, Kathleen Jackson, Donald Johnson, Sushil Jajodia, Burt Kaliski, Steve Kent, Rob Kolstad, John Linn, Steve Lunt, Teresa Lunt, Doug McIlroy, Michael Merritt, Dan Nessett, B. Clifford Neuman, Peter Neumann, Ron Olsson, Amy Reiss, Michael Roe, Jeff Schiller, Joe Tardo, Chris Wee, Len Wisniewski and Ken Van Wyk.

Finally, and most importantly, I'd like to thank all those who submitted articles for this special issue. Without you, this could never have been done.