An open system is one which an 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, requires a trustworthy implementation of both the security mechanisms and those mechanisms' protections. Papers in this special issue touch upon these themes.

The paper by Ware discusses 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 non-technical constraints that must be met.

Issues of trust in a network abound; and the paper by Klein, Beth, and Yahalom explore trust-based navigation in distributed systems with inhomogeneous trust relationships. This problem addresses issues raised by the way open systems are used in an internetworked environment, and so is relevant to security and integrity in open systems.

The third paper, by Krajewski, Chipchak, Chodorow, and Trostle, describes work augmenting the Kerberos authentication system to use smart cards; this presents one solution to 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 the solution in the paper is an interesting resolution of this problem.

The next paper, by Radai, spans the boundary between networks and single hosts with a discussion of integrity checking using checksumming. It presents an analysis of the differences between checking the integrity of a file transmitted between hosts, and of a file resident on a single host. Both issues arise in open systems, the former in networked systems and the latter in systems on which files can be (maliciously) altered.

The use of security constraints to scan a system for potential security problems is addressed in the paper by Heydon and Tygar. They describe a system for specifying and checking such constraints, and then apply it to a UNIX system to look for security problems. Their results show that careful specification of security requirements may help in detecting some of the more common security problems.

The issue concludes with a paper by LaPadula which 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
Vinton 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
Ken Van Wyk

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

Matt Bishop