FIREWALL CONFIGURATION

The firewall configuration errors study published in Computer's June issue (A. Lampson, “A Quantitative Study of Firewall Configuration Errors,” pp. 62-67) was a nice surprise. This is an important subject, but not much is written about it.

As the author points out, “a systems administrator must configure [the firewall] according to a security policy that meets the company’s needs.” It seems that the first question an auditor would ask is, “Does the firewall configuration correspond to the company’s firewall security policy?” Or perhaps the question is, “Does the company even have a policy?” Verification against a security policy removes subjectivity from the error classification.

Unfortunately, this article starts below this level and is more concerned with pure CheckPoint firewall configuration errors, which in many respects are due to subtle design issues that are not relevant to more recent firewall designs. The lack of a formal top-down approach to security is the reason for the misconfiguration of any brand of firewall and the subsequent large-scale security failures.

In my professional practice, I use a well-known access control model (B. Lampson, Protection, ISS Princeton, 1971) to formalize firewall security policy implementations (policy level) into firewall access matrices (functional specification level). Firewall configurations (detailed design level) are just translations of this matrix into configuration commands.

In its simplest form, such a matrix has rows listing all subjects accessing objects and columns representing the objects the subjects access. Intersections document the access rights. Incoming access subjects are remote systems and users, and objects are local systems. The access rights list permit-using protocols like HTTP or “none.”

It probably isn’t possible to control a complex firewall without such a matrix. For multiple security domains or inter-facel firewalls, setting up access matrixes between the relevant interface pairs makes the complexity manageable.

Wlodek Stankiewicz
Nice, France
wlodek.s@free.fr

The author responds:
I agree that a disciplined top-down methodology for writing a firewall policy is the right way to go. Even better than a methodology would be software that provides a higher level of abstraction to the administrator. In fact, several years ago I participated in the design and implementation of such a system at Bell Labs (Y. Bartal et al., “Firmato: A Novel Firewall Management Toolkit,” Proc. IEEE Symp. Security and Privacy, IEEE Press, 1999, pp. 17-31.)

However, as my article shows, reality is much uglier than we would hope for. Well-meaning designs and methodologies break down in the face of an overworked, high-turnover staff that is under time pressure and facing business demands that conflict with good security practices. I mainly intended to highlight what most security experts know from their own personal experience: that lots of firewalls are poorly configured. And, in my opinion, this situation is not likely to change any time soon unless we see much better tools appearing in the market.

Avisahi Wool
Somerset, N.J.
yash@acm.org

OPEN SOURCE PERSPECTIVE

In “A Critical Look at Open Source” (IT Systems Perspectives, July, pp. 92-94), Brian Fitzgerald presents an illuminating perspective of the relations between the software industry and collaborative programming.

That the author should find it necessary to state that “most [open source] developers are in stable cohabitating relationships, often with children, and are experienced IT professionals, paid for their work on F/OSS projects” is very revealing of the ethical chasm that exists between industry and the open source community.

I wonder how many of the great names in the history of computing (from the homosexual, idiosyncratic Alan Turing on) would have fit today’s ideal picture of the software industry. Industry, by and large, doesn’t want highly original, creative people: Companies prefer predictability and conformity. For all I know, these qualities might even be a sine qua non for the management of a software project in an industrial environment.

The concern that errors could “irretrievably harm F/OSS projects, which lack the marketing muscle to help undo the damage to volunteers’ reputations” is amusing. Industry thinks developing error-ridden, poor quality software is acceptable as long as there is a marketing department that can fool customers into accepting it as a fact of life.

Most industrial software is flawed, delivered late, over budget, and, in general, a resource hog. Much of it is driven by marketing needs more than technical excellence. The relative lack of these defects in open source products is due not just to the quality of open source programmers but also to the industry’s values and practices.

Trying to introduce open source principles into these failing practices would be the equivalent of the computing profession condemning itself to technical quackery.
Simone Santini
La Jolla, Calif.
ssantini@sandiego.edu

We welcome your letters. Send them to computer@computer.org.