Foreword

Hack Proofing Your Web Applications encourages you to address security issues from the earliest stages of application development onward. Our premise is that there is too much at stake to wait for an audit (or worse, a customer) to find flaws or errors in your code. While we acknowledge that there is no way to *completely* eliminate the risk of a malicious attack on your code, we firmly believe that by following the instructions and recommendations in this book, you will dramatically reduce both the likelihood of an attack as well as mitigate the extent of the damage should an attack occur.

This book covers in detail the following key points to successfully hack proof your Web applications:

- A security process must researched, planned, designed, and written for your organization. The process should include a network security plan, an application security plan, and a desktop security plan. All developer, administrator, and quality assurance teams should participate in creating the plan and ultimately be aware of their role in the security process.
- Testing is a fundamental component to application security. Security tests should be as true to a real attack as possible to establish the success or failure of the security measures chosen. Your defenses should take so much effort to penetrate that hackers will be discouraged by the time and effort required.

- Developers must keep current on changes and/or enhancements to the toolsets that they are using. This is essential in development because of the fast pace at which technology changes. Oftentimes patches or new releases are available and yet are not used because of a lack of awareness or a time-consuming backlog prevents proper installation.
- Developers, Webmasters, and network administrators must keep current on known security threats; this can be easily accomplished by monitoring such Web sites as www.SecurityFocus.com or www.cert.org. These sites offer not only a listing of current issues, but also a forum for developers to seek advice regarding security as well as solutions to registered issues.

Security should be multilayered; it is by necessity complex, at all levels. What may work for one programming language may not work for another. The primary goal of this book is to make developers aware of security issues inherent in each programming platform and to provide sound programming solutions.

Chapter 1, "Hacking Methodology," provides you with a foundation-level understanding of the hacker community and its various motivations. Chapter 2, "How to Avoid Becoming a Code Grinder," discusses the fundamental importance of thinking "creatively" as a programmer and explains the perils of developing code without fully understanding its use, function, and ultimately its security flaws. Obstacles to creative and analytic thought include: An environment controlled by management and business interests that are restricted by physical and intellectual security concerns, industry regulations, dependence on older technology, and cost and deadline constraints; this type of environment does not support open evaluations and testing. Chapter 3, "Understanding the Risks Associated with Mobile Code," explores the dangers associated with the use of VBScript, JavaScript, and ActiveX controls and other forms of mobile code, in the context of user safety and the application's effectiveness. An application's functionality and its real and perceived security are at risk when you use these powerful types of code.