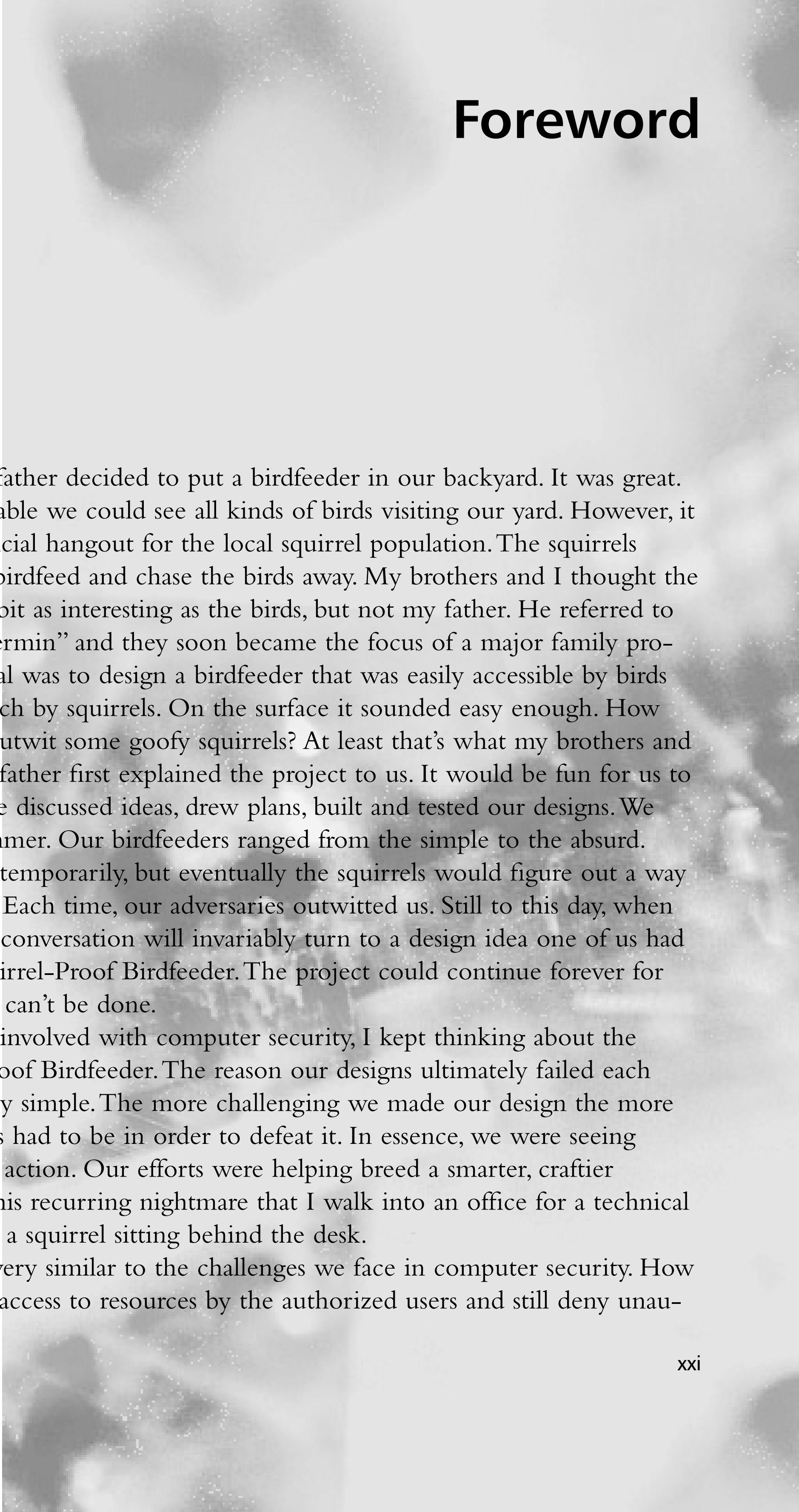

Foreword

Many years ago, my father decided to put a birdfeeder in our backyard. It was great. From our breakfast table we could see all kinds of birds visiting our yard. However, it soon became the official hangout for the local squirrel population. The squirrels would eat all of the birdfeed and chase the birds away. My brothers and I thought the squirrels were every bit as interesting as the birds, but not my father. He referred to them as "acrobatic vermin" and they soon became the focus of a major family project. The project's goal was to design a birdfeeder that was easily accessible by birds but impossible to reach by squirrels. On the surface it sounded easy enough. How hard could it be to outwit some goofy squirrels? At least that's what my brothers and I thought when our father first explained the project to us. It would be fun for us to work on together. We discussed ideas, drew plans, built and tested our designs. We worked on it all Summer. Our birdfeeders ranged from the simple to the absurd. Each design worked temporarily, but eventually the squirrels would figure out a way around our defenses. Each time, our adversaries outwitted us. Still to this day, when we get together, our conversation will invariably turn to a design idea one of us had for the Ultimate Squirrel-Proof Birdfeeder. The project could continue forever for one simple reason: It can't be done.

When I first got involved with computer security, I kept thinking about the Ultimate Squirrel-Proof Birdfeeder. The reason our designs ultimately failed each time was actually very simple. The more challenging we made our design the more cunning our squirrels had to be in order to defeat it. In essence, we were seeing Darwinian theory in action. Our efforts were helping breed a smarter, craftier squirrel. I still have this recurring nightmare that I walk into an office for a technical interview and there's a squirrel sitting behind the desk.

This scenario is very similar to the challenges we face in computer security. How can we provide easy access to resources by the authorized users and still deny unauthorized access?

Luckily, as Solaris System Administrators, we have some excellent tools available to us. Sun Microsystems has spent a great deal of effort in designing Solaris to be both stable and secure. This book is your reference guide for not only securing your Solaris systems, but also for securing the environment in which they operate. It is not designed to be an introduction to UNIX or a primer on Solaris System Adminstration, but rather a reference guide for experienced Solaris sysadmins who need to make sure their systems are secure.

Starting with Chapter 1, we attempt to level the playing field between you and your systems. It begins by discussing how to evaluate your current security scheme. One thing a hacker will always take advantage of is a sysadmin's complaceny. We start by going over the default settings you will find on a newly installed Solaris 8 system. We also go over the basics of testing, monitoring, and documenting security procedures.

Next, in Chapter 2, we cover the standard security tools available from Sun Microsystems. This includes an overview of Sun's BSM product and a look at the features of Sun's Trusted Solaris 8.

In Chapter 3, we introduce third-party security tools which are commonly used to secure and monitor Solaris systems. This chapter not only recommends some valuable tools to have on hand but where to get them and how to configure them for maximum effectiveness.

We begin discussing how to protect our resources in Chapters 4 and 5. First, by covering how users are authenticated on a Solaris system. Then by discussing how to configure file permissions and commonly used protocols such as FTP and NFS to transfer information safely among our authenticated users.

Once we have our systems secure, we need to explore our options for providing secure network services. Network users today need access to resources both on your local network and on the Internet. Opening this door can be a tremendous headache for a sysadmin. A major portion of this book is devoted to providing secure access on both sides of your router. Chapter 6 expands our focus to how Solaris 8 operates securely in a networked environment by providing DNS and DHCP services to network clients. In Chapter 7, we learn how to configure a secure Web and e-mail server. In Chapter 8, we narrow our networking focus by concentrating on how to configure Solaris to be a router and provide firewalling services. Chapter 9 is totally devoted to providing information on the configuration of the security features of Squid, one of the most popular apps for providing Web access to users.

Knowing your opponent's methods and tools is the first step in defeating their efforts. Now that we've learned what tools we have available, in Chapter 10 we learn what tools hackers commonly use to circumvent our security. We cover the most popular methods of attack, such as Distributed Denial of Service, Ping of Death, and the much-hated buffer overflow exploit. We discuss how they are used, what to be on the lookout for and how to configure our Solaris systems to prevent their use against us.

Finally, in Chapter 11 we cover what we can do to prepare for that day when hackers make it passed our main defenses. This chapter covers the configuration of a Solaris Honeypot system using freeware or commercial products. With a well-designed Honeypot system and some luck, we can lure our intruders away from our real systems. If designed correctly, it can tie up an intruder while collecting information on them. We can use this data later to plug the gaps they used to get in. Our final chapter also covers the use of a popular file monitoring tool called Tripwire which takes a snapshot of our systems and alerts us when key files have been altered.

This book comes full circle. From describing the need for improved and consistent security to learning what to do when our efforts fail.

Our Ultimate Squirrel-Proof Birdfeeder Project failed for the same reason that many security plans fail. Squirrels, like many hackers, are very curious, very single-minded, and have a lot of time on their hands. They also tend to work together. Eventually we figured out how to defeat them. We found that by monitoring their efforts and changing our designs in response we were able to build our Ultimate Squirrel-Proof Bird Feeder. The key is that's it's not one design, but an ever-changing design. The same holds true for designing your Ultimate Hack-Proofing Solaris Plan. It's not something you do once and ignore. It takes constant reviewing, monitoring, and improving. Using the information in this book you will be able to keep your resources secure provided you understand the importance of one simple truth: The hackers are out there and they want your sunflower seeds.

—Randy Cook, SCSA
Technical Editor