1

How Did *That* Happen?

Vulnerability Survey

Late one night...

"Look! It's world exportable! All I have to do is get a valid user account," `aBl_tR3kr` (pronounced "able trekker") said aloud.

"What's world exportable?" asked `pl3b` (pronounced "plebe"), between bites of chocolate doughnut.

"This home directory on `windfall.naive.com`."

"Windfall?" asked `pl3b`.

"Yeah, it'll be our windfall in a moment!"

"What're ya gonna do?" queried `pl3b`, trying to learn as much as possible.

"Well," smiled `aBl_tR3kr`, "if they are silly enough to allow NFS over the Internet, they probably aren't using a shadow file either."

"Shadow file?" Now `pl3b` is really perplexed.

"Yeah. See, all I have to do is create a user account on my Linux box that matches one of theirs…yep, there's a user called `joe`." `aBl_tR3kr` typed rapidly away on the keyboard as `pl3b` looked over his shoulder. "Now, I'll log in here as `joe`, and whataya know, I've got Joe's home directory from `windfall.naive.com`!"

"Wow! That was easy!" `pl3b`'s thinking that breaking into UNIX/Linux systems is simple stuff.

"Now, lemme create the `.rhosts` file here, and then we'll login remotely." `aBl_tR3kr` continued with his running commentary as he worked. "OK! We're in

as `joe` over there. Now, let's check the password file…whooooya! No shadow file! I'll just email this password file back to myself, like so. Done!"

"So what good is that file? All the passwords are jumbled up," remarked `pl3b`.

"No problem, my wannabe friend. We'll just use Crack!" `aBl_tR3kr` smiled confidently. And another system fell into his clutches.

NOTE

While this dialog is for illustrative purposes, it does represent a very serious exploit. It does not necessarily reflect a typical attack nor does it reveal all of the potential exploits associated with gaining unauthorized access to a user account. In particular, there are numerous methods available to gain root access once unprivileged access is obtained. We expressly discourage this type of activity; however, it is important to have a sense of the kinds of things that might occur in order to better prevent them. We do encourage the study of the various vulnerabilities in common use for preventative purposes. Further information about general system and network exploits are available in the references in Appendix A.

What Happened?

This dialog poses a number of ill-configured services that allowed for unauthorized access to an Internet system. We will see that this is all too often the case, as illustrated in Chapter 2.

The first problem with the system `windfall.naive.com` is the fact that user home directories are exported via the Network File System (NFS) to the world. This could have been prevented by setting access restrictions on the NFS resources (discussed in Chapter 3), on the `portmap` utility (discussed in Appendix D), through `ipchains` or `iptables` (discussed in Chapter 15 and Chapter 16, respectively), or preferably through a combination thereof.

Our happy cracker, `aBl_tR3kr`, also took advantage of the trusted host file, `.rhosts` in this case, to gain unauthorized access to the system. The use of the `.rhosts` file in the dialog allowed `aBl_tR3kr` to log in to `windfall.naive.com` as `joe` without a password. We discuss such files and recommend against their use in Chapter 3. Then we talk about how to securely replace them in Chapter 11.

The next problem with the system `windfall.naive.com` is that it doesn't use password shadowing (described in Chapter 4). In this case, the lack of password shadowing means that a world-readable file (`/etc/passwd`), including each user's hashed (sometimes referred to as "encrypted") password, is available to anyone who can access a valid user account, which `aBl_tR3kr` was able to do. If a shadow file had been in use, he wouldn't have been able to get a list of hashed passwords as easily because the shadow file is readable only by the root user.

While the hashed password cannot be used to log in, the Crack utility (discussed in detail in Chapter 12) may be used to guess the password based on the hashed password. And Crack is pretty good at what it does. You can't prevent the bad guys from using Crack, but you can make it harder on them by using alternate hashing methods, as we see in Chapter 5.

Other Cracker Activities

Once an intruder has gained access to an account, there are a variety of things the intruder might do. Among the things that such a malicious user will almost certainly do is create a back door to make returning easier. The intruder will also erase any evidence of his or her activity. There are all sorts of freely available utilities that make these tasks simple.

The purpose of this book is to provide ways to make it difficult to break in initially as well as to detect the evidence of the attack quickly. But you must not use this book, or any other books, as your *only* resource of information for this purpose. New vulnerabilities are identified all the time, and patches and fixes are generated in response. Also, full-disclosure sites (noted in "Full-Disclosure Resources" at the end of this chapter), email lists, and newsgroups will provide additional details regarding the ever changing scene of computer security. Appendix A lists resources that will assist you in staying current.

So, Are You Going to Show Us How to Break into Systems?

No, the purpose of the dialog at the beginning of this chapter is to illustrate the types of activities that occur all too frequently. The intent of this book is to provide you with skills, knowledge, and tools that will allow you to better prepare your systems for use in environments where you do not always know who is accessing what. The techniques and methods discussed are all from the perspec-

tive of restricting use to authorized access and making it as difficult as possible for crackers to gain unauthorized access.

This book is how-to oriented. This means that we will discuss the ways in which various utilities may be used to help protect your environment. What you should do in terms of implementing security at your site is largely left to other texts. While we make recommendations throughout this book, nothing can replace a good organizational security policy (discussed in Chapter 2). Also, we provide references at the end of each chapter if you wish to further investigate each topic we discuss. You will find that many of these references discuss the particulars of what ought to be done to maintain good security generally.

Before we get into the details of using publicly available tools to help secure your Linux[1] system(s), we spend the rest of this chapter discussing the types of vulnerabilities and attacks that are in common use today. We do not detail how they work; rather, we define their characteristics so that you have an understanding of what they are. You can also find additional information through the references cited in "For Further Reading."

A Survey of Vulnerabilities and Attacks

We consider vulnerabilities in three separate categories: technical, social, and physical. Attackers may attempt to exploit vulnerabilities in one or more of these categories.

Technical

There are many types of attacks against the technology we use in computing. Some of these attacks are perpetrated through a modification of a program, script, or data, while others take advantage of the way a particular technology works. The following is a summary of common terms describing attacks that exploit various aspects of computing.

Trojan Horses. Hidden program or script, usually embedded in an authorized program or script, that causes undesirable or unauthorized behavior when the authorized program or script is executed. A particularly nasty example of embedding a Trojan horse or back door in a C

1. Most of what we discuss in this book is relevant to commercial UNIX platforms as well.

compiler is noted in "Trusting Trust," pp. 801–802, in the book *Practical UNIX and Internet Security*. This type of exploit is rather difficult to prevent. Various sections of Chapter 3 describe ways to reduce the likelihood of this type of attack. In particular, verifying checksums and Pretty Good Privacy (PGP) signatures on files that contain them is described in Chapter 3 and revisited throughout this book.

Back Doors. Hidden program, script, or funtionality embedded in a normal program or script, which ultimately allows unauthorized access to a system. Similar to Trojan horses.

Password Cracking. Guessing passwords or utilizing a tool such as Crack to guess passwords. Mitigating password-related problems is discussed throughout this book, and particularly in Chapters 4, 5, 6, and 11.

File Permissions and PATH Settings. Improper settings of either of these can cause systems to be compromised. Further discussion of this topic is found in Chapter 4.

SUID Scripts and Programs. Scripts or programs that run with the real or effective user identifier (UID) set to someone other than the user who invoked the script or program. Normally, the concern is with programs that set UID (SUID) to root. A large number of exploits that allow an unprivileged user to become root take advantage of this mechanism. Similar, though fewer exploits take advantage of the set group identifier (SGID). We take a look at these issues in Chapter 4.

Trusted Hosts. The use of trusted host files often permits an exploit of one system to spread throughout an entire network. (These files are discussed in Chapter 3.) We talk about replacement functionality for these files in Chapter 11.

Buffer Overflows. Failure to bound a read buffer in a program may allow exploitation of the system by writing into the system memory allocated by the read buffer. While writing a program to take advantage of unbounded buffers is difficult and requires special skills, many programs written by such skilled folks are readily available on the Internet. We take a further look at mitigating these types of problems in the section "Software Testing" in Chapter 3.

Scanning and Sniffing. Network scanning may allow an attacker to identify the operating system (OS) running on a particular system as well as the network daemons available. Network sniffing may allow an attacker to obtain confidential information. Network scanning and sniffing is also very useful for security testing and debugging. We discuss some of the tools available for these purposes in Chapters 3 and 17. Ways of reducing the effects of attacks perpetrated with network scanners are discussed in Chapters 10 and 17. Methods to mitigate the effects of attacks using network sniffers are described in Chapters 11 and 15.

Spoofing. A user pretending to be another user, a host another host, an Internet Protocol (IP) address another IP address,[2] a domain another domain or address—all are examples of spoofing. These types of attacks and reducing their likelihood are detailed in the books *Hacker Proof* and *Maximum Security*, as cited in "For Further Reading." We discuss ways of alleviating these types of attacks in Chapter 15.

TCP/IP Attacks. Taking advantage of the way that Transmission Control Protocol (TCP) network connections work may lead to a variety of attacks. These types of attacks are quite difficult to prevent and detect. Many of the techniques we discuss in Chapters 10, 11, and 15 temper these attacks. However, you may also wish to consult *Hacker Proof*, as cited in "For Further Reading" for more details on TCP/IP attacks and what you can do to prevent them.

Session Hijacking. Whenever a user assumes control of a network session or connection. A special case of TCP/IP desynchronization attacks. See "TCP/IP Attacks," above.

Denial of Service. Any activity that prevents the normal use of system and network resources is considered a denial of service (DoS). Generally speaking, you can make DoS attacks much more difficult for the perpetrator, but you cannot prevent them. In Chapter 10, we discuss the `xinetd` replacement to `inetd`, which has the capability of preventing certain network-related DoS attacks.

2. Well, it is spoofing if the bad guys do it. We'll have a look at *masquerading* in Chapter 15.

Other Vulnerabilities. There are a variety of vulnerabilities in various network and system applications. Some of them are discussed throughout this book. Your best resource for continuing information about vulnerabilities in many applications is by keeping up to date (see Appendix A). We also note some resources in the section, "For Further Reading."

Social

The weakest link in any Security Policy (see Chapter 2) is the people who are authorized to use the computing environment. Those who deliberately attempt to compromise a computer environment by attacking this weakest link are engaged in what is called *social engineering*. Of all of the aspects of security, this one is the hardest to control. While this aspect of security is outside the scope of this book, we note a few of the common types of social engineering attacks and cite additional general security references in the section "Web Sites."

Shoulder Surfing. As its name implies, this is the process of capturing sensitive information, such as a user's password, by looking over someone's shoulder.

Manipulation. This category of social engineering attacks is often employed to obtain sensitive information. Examples include pretending to be a system administrator or professing to be a high-level official of the organization in order to obtain sensitive information from unwary individuals within the organization.

Physical

It is often said that once physical security is compromised then the game is over. This statement reflects the fact that, if an unauthorized person gains physical access to some part of your computing environment, all the technical security solutions in the world cannot prevent such a person from perpetrating some form of attack. This topic is also not investigated in this book; however, we list a few potential vulnerabilities. A good reference site for this and social engineering security issues may be found at the home page of the International Information Systems Security Certification Consortium, Inc. [(ISC)^2], as noted in the section "Web Sites."

System Access. Once physical access to a computer has been gained, everything from booting the system in single-user mode (see "A Note about LILO" for a discussion of restricting single-user mode access) to simply taking the system becomes possible. Physically restricting access to your most critical systems is highly encouraged.

Networking Issues. Networks are commonly implemented over electrically based media, such as 10/100 base T, radio frequency (RF) communications, microwave technology, and satellite communications. All of these forms of communication may be intercepted through various tapping techniques.[3] The use of fiber-optic media substantially tempers this issue. See the (ISC)^2 home page, as referenced in the section "Web Sites," for further information about this topic.

Other Physical Access Issues. There are a great many vulnerabilities associated with unauthorized physical access, ranging from pulling a fire alarm to looting as a result of a natural disaster. Check out the (ISC)^2 home page, as referenced in "Web Sites," for further information about this topic.

Summary

This chapter provides an overview of a variety of different vulnerabilities and exploits as they relate to computers and networks. While not the focus of this book, this chapter motivates the requirement for securing systems, which *is* the focus of this book. We've provided numerous references for further study.

For Further Reading

Books

Anonymous. *Maximum Security: A Hacker's Guide to Protecting Your Internet Site and Network*. Indianapolis, IN: Sams.net Publishing, 1997.

3. Newt Gingrich found this out the hard way!

Atkins, Derek, et al. *Internet Security: A Professional Reference*, Indianapolis, IN: New Riders Publishing, 1996.

Barret, Daniel J. *Bandits on the Information Superhighway*. Sebastopol, CA: O'Reilly & Associates, 1996.

Chapman, D. Brent, and Elizabeth D. Zwicky. *Building Internet Firewalls*. Sebastopol, CA: O'Reilly & Associates, 1995.

Cheswick, William R., and Steven M. Bellovin. *Firewalls and Internet Security: Repelling the Wily Hacker*. Reading, MA.: Addison-Wesley, 1994.

Cooper, Frederic J., et al. *Implementing Internet Security*. Indianapolis, IN: New Riders Publishing, 1995.

Denning, Dorothy E. *Information Warfare and Security*. New York, NY: Addison-Wesley, 1998.

Garfinkel, Simson, and Gene Spafford. *Practical UNIX and Internet Security*, 2d ed. Sebastopol, CA: O'Reilly & Associates, 1996.

Garfinkel, Simson, and Gene Spafford. *Web Security & Commerce*. Sebastopol, CA: O'Reilly & Associates, 1997.

Hughes, Larry Jr. *Actually Useful Internet Security Techniques*. Indianapolis, IN: New Riders Publishing, 1995.

Icove, David, et al. *Computer Crime: A Crimefighter's Handbook*. Sebastopol, CA: O'Reilly & Associates, 1995.

Klander, Lars. *Hacker Proof: The Ultimate Guide to Network Security*. Las Vegas, NV: Jamsa Press, 1997.

Kyas, Othmar. *Internet Security Risk Analysis, Strategies, and Firewalls*. London: International Thomson Computer Press, 1997.

Pabrai, Uday O., and Vijay K. Gurbani. *Internet and TCP/IP Network Security Securing Protocols and Applications*. New York, NY: McGraw-Hill, 1996.

Siyan, Karanjit, and Chris Hare. *Internet Firewalls and Network Security*. Indianapolis, IN: New Riders Publishing, 1995.

Interesting Cracker Tales

These references do not provide any technical details but do relate stories about (in)famous attacks.

Dreyfus, Suelette. *Underground Tales of Hacking, Madness, and Obsession on the Electronic Frontier*. Kew, Australia: Mandarin, 1997.

Littman, Jonathan. *The Watchman*. Boston, MA: Little, Brown, 1997.

Shimomura, Tsutomu, with John Markoff. *Takedown*. New York, NY: Hyperion, 1996.

Stoll, Cliff. *The Cuckoo's Egg*. New York, NY: Pocket Books, 1990.

Web Sites

For information related to Domain Name Service (DNS), InterNetNews (INN), and Dynamic Host Configuration Protocol (DHCP), check out the home of the Internet Software Consortium (ISC),

```
http://www.isc.org/
```

The home of `sendmail`, a commonly used mail transfer agent (MTA), has lots of good information about `sendmail`-related vulnerabilities:

```
http://www.sendmail.org/
```

For good general security information, links, and references, visit the Information Systems Security Association site,

```
http://www.issa-intl.org/
```

Professional certification in the field of security is available from (ISC)². This site also has information about technical, physical, personnel, and many other areas of security. Their home page is

```
http://www.isc2.org/
```

For an excellent collection of security tools and resources, visit the home of the Computer Operations, Audit, and Security Technology (COAST) site at

```
http://www.cerias.purdue.edu/coast/
```

which also contains a great many links. Check them out!

Other good security sites include:

```
http://www.fish.com/security/
http://www.security-focus.com/
http://www.cerias.purdue.edu/
ftp://ftp.porcupine.org/pub/security/index.html
```

Full-Disclosure Resources

The following Web sites are considered full-disclosure sites because they publish details of various vulnerabilities and often offer code samples that exploit some of the vulnerabilities. The purpose of listing these sites here is to inform you about what the bad guys already know.

```
http://www.insecure.org/
http://www.l0pht.com/
http://www.rootshell.com/
```

You will also find a list of underground sites at

```
http://www.cerias.purdue.edu/coast/hotlist/
```