Introduction to Internet Security

John E. Howland
Department of Computer Science
Trinity University
715 Stadium Drive
San Antonio, Texas 78212-7200
Voice: (210) 736-7480
Fax: (210) 736-7477
E-mail: jhowland@ariel.cs.trinity.edu
Web: http://www.cs.trinity.edu/~jhowland

November 26, 2002

Abstract

Internet computer security is a problem of major significance, particularly considering the pervasive impact of the Internet on nearly all segments of the computing industry. Systems used in homes, education, business, and government are all susceptible to invasion. This introduction to computer security is intended as a starting point for further research of computer security issues related to the accounting profession as it deals with firms doing commerce using computer systems which are based on Internet technology. [1]

Subject Areas: Computer Security
Keywords: hacker, cracker, virus, firewall, worm, Trojan horse, time bomb, denial of service, trap door, information warfare

[1] This paper is a tutorial on Internet security which is part of "Opportunities and Risks in CPA Assurances Computer and Networking Security Systems" by Robert E. Jensen, Jesse H. Jones Distinguished Professor of Business Administration, Trinity University, John E. Howland, Professor of Computer Science, Trinity University and Bruce Sidlinger, President, Sidlinger Computer Corporation. This case study was selected as one of eight winners in a national competition held by the American Institute for Certified Public Accountants. This paper appears in the AICPA *1998 AICPA PROFESSOR/PRACTITIONER CASES*, Pages Case No. 98-03:1-33,

1 Introduction

Computer security has been a topic of great importance since the emergence of second generation main frame computers (early 1960's). Some of the earliest applications of these main frame machines involved financial calculations. Floating point arithmetic was often used to represent (inexactly) the values of a penny or dime. More than one dishonest programmer developed software which would accumulate the inaccuracy of each account transaction into a single account. The losses were not noticed on individual accounts, but were later noticed at the institutional level.

In the early 1960's, as the first time-sharing/multi-user systems were being developed, system designers realized that they had to give serious consideration to security designs because these systems allowed remote access to the computer which were not covered by the physical security measures protecting the computer.

Donn Parker, of SRI, emerged as an international computer security expert during the late 1970's and early 1980's. His book, *Fighting Computer Crime* [Park 83], though published in 1983, prior to the Internet as known today, describes a variety of techniques for achieving secure networked systems.

2 Security Policies

It is important to separate the establishment of security policy from the mechanisms one might use to implement or enforce a particular security policy. For example, one might decide that it will be the policy to not allow access to a certain building after 11 p.m. on weekdays. A mechanism for implementing this policy might involve the installation of door locks on all entry points and providing staff which will lock the doors promptly at 11 p.m. on weekdays.

All too often, in the realm of computer security, security policy is decided by available hardware mechanisms rather than first deciding on policy and then designing a mechanism which will be used to implement the policy.

2.1 Need to Know

The military *need to know* policy says, in part, that individuals will not have access to information for which they have no use.

An example of this policy, when applied to a multiuser file system, might be that users have read-write access to their own files but no access to any other user's files. Such a policy may make file sharing cumbersome. An alternate policy might allow read, but not write, access to other user's files. This would go beyond *need to know* because a user may be able to see the contents of files he or she has no need of, even though the user is prohibited from changing such files.

2.2 Policy Authority

Security policy may be established globally by top-level administration or some security policy matters may be delegated to middle level administrators and perhaps even individuals.

3 Security Mechanisms

Computer security mechanisms involve the design of computing hardware, software as well as operational procedures. Some of these ideas are illustrated in the following sections using the Unix operating system. Access to a Unix computer system is controlled through an account password. The passwords must be stored in the system so that the system may authenticate individual users. The password database is encrypted as briefly described in the following Unix manual page. This password encryption scheme [Gram 84, Morr 79] was developed by Robert H. Morris.

3.1 Passwords

```
CRYPT(3)                Library functions                CRYPT(3)

NAME
       crypt - password and data encryption

SYNOPSIS
       #include <unistd.h>

       char *crypt(const char *key, const char *salt);

DESCRIPTION
       crypt  is  the  password  encryption  function.  It is based on the Data Encryption
       Standard algorithm with variations intended (among other things) to discourage  use
       of hardware implementations of a key search.

       key is a user's typed password.

       salt  is  a two-character string chosen from the set [a-zA-Z0-9./]. This string is
       used to perturb the algorithm in one of 4096 different ways.

       By taking the lowest 7 bit of each character of the key, a 56-bit key is  obtained.
       This  56-bit  key is used to encrypt repeatedly a constant string (usually a string
       consisting of all zeros). The returned value points to the encrypted  password,  a
       series  of  13  printable  ASCII characters (the first two characters represent the
       salt itself). The return value points to static data whose content is  overwritten
       by each call.

       Warning:  The key space consists of 2**56 equal 7.2e16 possible values. Exhaustive
       searches of this key space are possible using massively parallel computers.  Soft-
       ware,  such  as  crack(1),  is  available which will search the portion of this key
       space that is generally used by humans for passwords.  Hence,  password  selection
       should,  at  minimum, avoid common words and names. The use of a passwd(1) program
       that checks for crack-able passwords during the selection process is recommended.

       The DES algorithm itself has a few quirks which make the use of the crypt(3) inter-
       face  a  very  poor choice for anything other than password authentication. If you
       are planning on using the crypt(3) interface for a cryptography project,  don't  do
       it: get a good book on encryption and one of the widely available DES libraries.

CONFORMING TO
```

```
       SVID, X/OPEN, BSD 4.3

SEE ALSO
       login(1), passwd(1), encrypt(3), getpass(3), passwd(5)

                        September 3, 1994                       1
```

The above manual page mentions that crypt function is based on the DES algorithm. DES is the government regulated Data Encryption Standard. In July 1998, the Electronic Frontier Foundation announced that it had succeeded in breaking the DES standard in 56 hours using a modified PC computer. The total cost of the modified machine was approximately $200,000.00. Previous successful attempts to break the DES standard took 5 months using a nationwide network of computers and 39 days, again, using a nationwide network of computers. This development means that it is now necessary to improve standard encryption algorithms.

3.2 File Access Mechanisms

Unix file systems support, separately or in combination, read, write, execute and execute, using another user's id, file access modes. These permissions are available for the individual owner of the file, for the group and for the public. The Unix file access mechanism was designed to support a wide variety of security policies, including arbitrary security policies implemented by individual users. For example, individual users may decide what permissions to give themselves (preventing write access would protect a file from accidental deletion), members of their group and general public access. Since file system directories are files, access modes on directories control whether or not the names of files are publicly known and are changeable as well as whether the contents of files are publicly known or changeable. The owner of the file also may change these permissions for members of the group associated with that file as well as the group to which the file belongs.

The system administrator exercises the same control mechanism for files owned by administrative accounts. This flexible security mechanism allows implementation of a variety of different security policies for different groups of users within the same computing system. Other types of computer operating systems (except for most personal computer systems) provide a similar file security mechanism.

4 Internet Security

Ultimately, computer system security may be reduced to the physical security of the components of the computing system. This introduction focuses on the security of systems which use Internet technology. Such systems themselves are often distributed over a geographic area and are accessed by users which are potentially at any geographic location.

How the Internet works is a very large subject and many resources are available for study [Come 95, Come 96, Come 97, Come 98]. A brief description of how the Internet works is given to provide a context for the following sections.

An internet is a collection of interconnected networks. The interesting part of internet technology lies in the methods used to connect the individual networks in an internet. The simplest method of interconnection of two networks is to have a single machine which contains a network interface on each network. Such a machine can function as an information router, forwarding data from one network to another.

It is customary to subdivide files into reasonably small sized blocks, called packets, so that the transmission of a large file from one network station to another does not unfairly monopolize the *bandwidth* (transmission capacity) of the network *media* (cable).

4.1 Identifying Network Stations

In an internet, networks are numbered and, within each network, each network interface is numbered. Hence, it is possible to identify each network station (computer) with at least one number which consists of the network number followed by the interface number. This number is known as the IP (Internet Protocol) number of the station. Stations which have more than one network interface will have more than one IP number and have the potential of forwarding packets of information from one network to another. Messages are sent from one station to another station using message formats consisting of a pair of items; (message-header, message-data).

Each message-header contains, among other items, the IP number of the packet destination as well as the IP number of the packet source. Packets are sent from one station to another along a route which is determined network routers. Each router examines the destination address of each packet it receives and consults a routing table to determine which connected network to which it should be sent.

Occasionally, routers will be so busy they cannot forward a packet to another network, or there will be errors in the routing tables so that no route is correct for a packet. In this case, the packet is simply discarded. This type of delivery of packets is called *best effort* and is unreliable because packets may be lost under certain circumstances. This is not a problem because network reliability depends on higher level conventions, called network protocols, which attempt to retransmit packets when lost.

IP numbers are used to identify internet hosts and must be entered by users when accessing internet services such as Telnet, FTP, WEB URL's, etc. Multi-digit numbers are difficult to enter into user interface programs and recall from memory. Internet developers devised a *Domain Name System* (DNS) which consists of a distributed hierarchical database of names which may be used for most of the IP numbers on the Internet. Internet hosts are grouped together into domains within an organization and networks are grouped together by type, educational (EDU), commercial (COM), etc. Finally, the previously mentioned

types of networks are grouped together by country to form the top level of the DNS database hierarchy. The DNS system automatically converts host names, which are easier to remember, to actual IP numbers. An example of such use occurs when you refer to the Apple Computer WEB site using the URL `http://www.apple.com`.

The important thing to remember about the way the Internet works is that the familiar model of the telephone system (having a dedicated connection from one point in the world to another) does not apply. Information is sent in small packets and the packets may traverse different routes when sent from one location to another. The internet transmission media are constantly being shared by packets from a variety of sources as they make their way (usually) to their destination. Network protocols use the basic internet transmission mechanism to achieve reliable and secure transport of information.

4.2 Network Protocols

Network protocols are standardized communication conventions (implemented in computer software) to provide a variety of internet work services. Among these are protocols for transmission of files (FTP), transmission of WEB documents (HTTP), remote interactive terminal sessions on another computer (Telnet), etc. Protocols exist to provide higher level services, such as reliable point to point connections (Streams) or secure transmission of information. These protocols may cause lost packets to be automatically retransmitted or may encrypt and decrypt data so that it is not easily readable at intermediate routing points during transmission.

5 Internet Security Threats

Internet technology, because it provides a variety of services to clients which are physically located anywhere in the world, poses a challenge to designers of secure systems. The potential exists, anywhere on the route along which packets of a secure transaction are transmitted, for an intruder to access and perhaps even modify the transaction data packets.

It is not necessary to compromise the security of a router to have accesses to packets, one need only jeopardize the security of some host on a destination network on one of the legs along the route. Encryption of packets is a technique which may be used to protect the information in the packet. This removes most of the risk of packet access, however, modification of an encrypted packet will usually destroy the data in the packet and may disrupt the transaction.

The following sections contain a few of the many ways computer system security has been attacked, allowing an intruder to take administrative control of a machine or remove a machine from service. Either action is potentially disastrous for WEB based financial transactions.

6 Unix Security

Internet technology as we know it today evolved from a subset of features which were developed within the Unix operating system. The Internet today supports nearly every type of computer and computer operating system and, recently, Microsoft Windows NT systems have been widely used as Internet hosts, but the majority of Internet hosts for database, WEB servers and transaction processors still use the Unix operating system. Some discussion of a few of the security issues for the Unix operating system is given in the following sections.

6.1 Cracking Passwords

As explained in Section 3.1, a database of encrypted passwords is stored in every Unix machine. A well known Unix program, called `crack`, is available. This program uses information about poor choices of passwords, such as using dictionary words for passwords, to attempt to decrypt the passwords in a password database. Such passwords may then be used to gain unauthorized access to the machine. Intruders often use guest accounts to gain access to the encrypted password database, move it to another machine, and then execute `crack` to gain access to authorized system accounts. System administrators regularly use the `crack` program on their systems to determine which users have crack-able passwords. Passwords which do not use dictionary words and which use a mixture of letters, numbers and special symbols are more difficult to crack.

6.2 sendmail

Unix systems have many special systems programs, called *daemons*, which run continuously in the background, or upon request, to provide services such as sending and receiving mail, remote file service, WEB service, FTP service, etc. Daemons are important extensions to the basic facilities of the operating system, but have proven to be a source of problems from a security point of view. For example, the Internet worm, written by Robert Tappan Morris in November, 1988, exploited a deficiency of the Unix `sendmail` daemon to send executable copies of itself from one machine to another on the Internet [Spaf 88, Spaf 89]. The program did not destroy files on a machine or damage computer hardware, but it did contain a flaw which caused it to recopy itself on a machine, consuming resources and disrupting network services. Morris, the son of Robert H. Morris, who devised the Unix password encryption schemes [Gram 84, Morr 79], was convicted of violation of the Computer Fraud and Abuse Act of 1986, 18 U.S.C. s 1030(A) (5)(A) (1988), appealed unsuccessfully in 1991 and was sentenced to three years probation, 400 hours of community service, a fine of $10,050.00 and the costs of his supervision. This Internet security incident, focused attention on the vulnerability of Internet hosts to attacks. While most of problems which were exploited by the worm have been fixed, the threat still exists today, ten years later, that new problems will be found in Unix daemons and other programs. As recently as July, 1998, new security problems [Walk 98, Gard 98]

have been discovered with programs for sending and receiving e-mail.

6.3 Denial of Service

Denial of service security attacks remove or disable certain system services (or perhaps the entire system). Sometimes a denial of service attack will simply degrade system performance to a level where, though still available, system service is effectively non-existent. Some attacks have caused a crash of the operating system kernel itself. In other cases, the system service daemon has crashed, thereby removing that service without altering other services being offered by the host. An example of a denial of service attack is to send a large number of packets to host in an effort to overwhelm the host and force a crash or reduce service to levels which are unacceptable. Since packets must contain a source address (in this case, the address of the attacker) as well as a destination address, it would be easy to identify the attacking machine. To avoid detection, the attacker machine arranges to send packets having a source address from some other Internet host. This process is known as *IP spoofing*. The so-called *Land* attack involves IP spoofing and usually results in denial of service or a crash. Ferguson and Senie [Ferg 98] have proposed a workaround solution to IP spoofing where each network router to external networks would filter out any spoof-ed packets, i.e., packets being sent to another network whose source address are not equal to the local network. For this workaround solution to be totally effective, filtering would need to be installed on every router on the Internet which routes packets to external networks. This situation highlights one of the difficulties of internet security, namely the requirement that the network administrators of each network on the Internet carefully coordinate security efforts and maintain similar standard levels of security. This is impossible in practice, since many network administrators give little or no attention to network security issues.

6.4 Repeated Attack

Repeated attacks usually involve a systematic probing of certain widely used network ports, looking for responses indicating that the host operating system may be vulnerable to attack. A recent CERT (Computer Emergency Response Team) incident note [CERT 98] alerts Internet network administrators of the existence of new cracker tools to perform repeated systematic attacks.

On July 9, 1998, CERT announced that a new intruder tool had been released by the cracker community which could be used to scan networks on the Internet for many different vulnerabilities. CERT received many reports in early July 1998 that the tool was in widespread use within the intruder community. The tool has the capability to test an Internet host for the following security vulnerabilities:

- statd vulnerability
- imap/pop3 vulnerabilities

- IRIX machines that have accounts with no passwords
- bind vulnerability
- cgi-bin vulnerabilities
- NFS file systems exported to everyone
- X11 (open X servers)

This is an example of the kind of sophisticated tools being developed and used within the cracker community. When such tools become widely available, they may be used by novice crackers to gain access to systems or otherwise disrupt operation of systems, however, they may also be utilized by systems administrators to test the vulnerability of their system and take appropriate security action.

6.5 CGI Scripts

The World Wide Web (WWW) uses a network protocol called Hyper Text Transport Protocol (HTTP) to implement a client-server computing environment over the Internet. HTTP server programs, called Web servers respond to requests, which have been sent over the Internet, from HTTP client programs, called Web browsers. The most widely used server and client programs are produced by the Apache group, Netscape Corporation and Microsoft Corporation.

Common Gateway Interface (CGI) scripts are programs which run on the Web server program when requested by a Web client such as Netscape Communicator. The CGI protocol provides mechanisms for input from a client browser (such as a database query) to be sent to a web server for processing by the CGI program. The CGI standard allows communication between the CGI script and Web server so that the CGI program output may be sent back to the client browser program in the form of a dynamically created Web page.

Web servers may be configured so that programs which have been written by Web client users will be executed on the server. Of course, this is a large potential security threat for the Web server machine in situations where the Web client user cannot be trusted or otherwise certified.

Another CGI security concern involves screening or filtering of input data which is sent to the server program for processing. In general, it is a difficult task to filter out all inputs which may lead to error situations or pose security threats. One obvious problem is the verification of the identity of client users by password. If the clear text of the password is sent over the Internet, it can be visible by any intruder on a machine connected to the source or destination networks. One solution for this problem is to use encryption/decryption protocols on the client and server programs so that the password, which is sent over the Internet, is not easily visible.

6.6 Windows NT Security

In Section 6, we covered a few areas of security concern for Unix based systems. The security issues seem to be similar for other widely used operating systems. In this section we address some security problems in the Microsoft Windows NT operating system.

Denial of service attacks have been discovered. Microsoft has provided security patches for most of these problems. Microsoft distributes software updates, known as *Service Packs*. Currently, SP3 fixes more than thirty known security related problems and should be applied to any Windows NT workstation or server which is attached to the Internet. In addition, Windows NT has proven to be susceptible to the following types of attacks:

- Weak Passwords, Authentication Attacks
- Privilege Escalation
- Non-captive Environments

7 Improving Internet Security

There is much interest in improving Internet security. One of the most widely known efforts is that of Computer Emergency Response Team [CERT 98] Coordination Center which is part of the Network Systems Survivability program in the Software Engineering Institute, a federally funded research and development center at Carnegie Mellon University.

Among the materials available are a collection of documents called Security Improvement Modules. These include:

- Detecting Signs of Intrusion [CERT 97]
- Security for a Public Web Site [CERT 97]
- Security for Information Technology Service Contracts [CERT 98]
- Preparing to Detect Signs of Intrusion [CERT 98]

These papers provide information on software risk evaluation and practical guidance for organizations interested in improving security of Internet based systems.

Security consulting firms provide a variety of useful services for improving the security of Internet based computing systems. For example, Ernst & Young provides an educational program which teaches in-house systems staff some the art of cracking into their local site. The class teaches techniques on how to break into Internet, intranet, extranet and dial-in systems as well as exploitation, reconnaissance and host vulnerability evaluation. In an evaluation of the Ernst & Young course, Inforworld [INFO 98] found that not only did the course cover the standard well known methods to break into systems, but it also covered

some novel hacks such as breaking into an Windows NT machine with Virtual Network Computer remotely by using a 3Com PalmPilot. The course provides three days of intensive training on network and net host security.

7.1 Open Software

There is a development within the computer software community which is known as Open Software. Open Software is usually distributed, at no cost (or at the cost of the distribution media) under a license which provides not only an executable version of the program, but also provides the original source programming language which may be used to generate executable versions of the software. The licensee is free to modify the program in any way and make any use of the software. Such licenses usually permit redistribution of the software and any changes the licensee has made to the program, provided such distribution is done at no cost and includes all software sources. This radical approach to software development and distribution denies the concept of intellectual property in software was originally developed by Richard Stallman, founder of the Free Software Foundation (http://www.fsf.org/fsf/) in 1984.

Stallman's idea was that users and programmers would be able to receive software at no (or nominal) cost and be free to use and improve the software and provide the software plus improvements to others who would, in turn, use and improve the software. A user would be prohibited from selling software under this licensing agreement, but one could sell services which are related to using, supporting and documenting the software. The license agreement, known as the GNU [GNU 91] license agreement, provides not only free software, but also freedom for the licensee to make any changes to the software and even redistribute the software, provided the software and changes are distributed at no charge and are distributed, including changes, in source form.

Stallman's original free software project was to attempt to interest others in joining his effort to produce a free version of the Unix operating system. The name, GNU, associated with Free Software Foundation projects derives its name from the recursive acronym, Gnu's Not Unix. It is Stallman's vision that eventually free software would lead to a plethora of very high quality programs which would be widely (and freely) available. It has taken a long time, but recently free software is beginning to be a significant force within the software industry. An example of this is the widespread use of the Linux [Linu 98] operating system. A Finnish student, Linus Torvalds, began the development of the kernel (heart) of a Unix compatible operating system. He used GNU license compilers and other free software programs and released his kernel under the GNU license. In a relatively short time hundreds, then thousands of programmers were contributing to this effort. The GNU software development efforts were quite fruitful as well, producing GNU versions of the hundreds of programs which make up a Unix system. Today, current versions of Linux, such as RedHat 5.1, rival commercial versions of Unix as well as operating systems from Microsoft, in performance, features and support.

Recently, a commercial software developer, Netscape Corporation, released

the source program for Version 5 of Netscape Communicator, free of charge, using a software license which is very similar to the GNU license. Netscape introduced the term Open Software to describe this software license. IBM Corporation recently announced a licensing agreement with Apache.Org concerning inclusion of its GNU license Apache Web server software with Internet server computer systems it markets. Apache is the most widely used Web server program on the Internet. It remains to be seen whether or not the Open Software licensing approach will be used by other major software vendors, but there are important implications for Internet security when GNU (or similar) license software is used.

Since the source code for open software products is freely available it may be inspected and analyzed by the entire software community. Such public scrutiny has the potential of providing a more complete review of algorithms used in the software and a more complete assessment of the security risks in the software than commercial software products receive. Finally, recent experience with the Linux operating system indicates that security problems, when detected, are fixed more rapidly (often in a matter of hours) than security problems with commercial software products. Although Linux is free, it is possible to buy commercial service contracts which support the Linux operating system. This development is beginning to be noticed by corporate information technology departments.

8 Physical Security

A time-honored maximum of computer security has been that you don't have a secure system unless you can provide a level of physical security for all of the hardware in the computer system. Some secure military computing sites go as far as not only providing as secure building to house the computing hardware, but also to provide electronic shielding to contain any electro magnetic radiation which may be generated by the computing system.

Unfortunately, when the Internet is used in computing applications, it is not practical to secure, physically, all elements of the computing system, nor is it always possible to provide physical verification of the identity of a remote computer system user. Since physical security is not possible on Internet based systems, the inherent risks must be assessed and be compared with the benefits to be derived from using Internet technology in the system design.

9 Certification of Internet Financial Transactions

A few years ago, the National Science Foundation withdrew its financial support for (and control of access to) the Internet. Immediately, business and industry began to explore the possibility of Internet based business transactions. These explorations resulted in astonishing growth of the Internet and business activity which, in turn, generated the need for Internet based financial transactions.

While some security issues were addressed in the design of the networking technology used in the original Internet, its designers did not anticipate the potential for Internet based commerce. Consequently, because of the success of these initial experiments conducting a variety of Internet based business activities, it is now necessary to address many security issues as an addition to current Internet technology.

Internet based business activity is now being promoted in virtually all segments of the economy and a part of this effort involves gaining customer acceptance of and confidence in Internet based transactions.

The American Institute of Certified Public Accountants has begin a certification effort which would provide an easily identifiable seal of approval of business practice which uses Internet technology. This certification provides a criteria for business practice, transaction integrity and information protection. Internet based business sites which have achieved certification may bear the *WebTrust* seal which includes verification of Internet based business practices by an independent CPA firm. WebTrust systems use VeriSign [Veri 98] public key encryption technology to transmit financial transaction data over the Internet.

10 Summary

This paper discusses some of the technical aspects of computer system security. While it is true that one does not have secure systems without physical security at each of the access points, machines, network routers, network cable, etc., there is yet another important factor of security; the human element of cracking.

Consultant Ira Winkler, in a presentation at the Black Hat Briefings '98 Conference, held in Las Vegas, Nevada, [Kers 98] described how, in four days of telephone work and internet research, he was able to obtain bank login id's and passwords for seventy three newly hired bank employees. This breach of security was performed by Winker who was working as a security consultant for the unnamed bank. Security experts estimated that it might have been easy to use the information gained in the security evaluation to transfer more than $2 million dollars from banking accounts.

The bank had rather good computer based security measures which included unbreakable encryption, firewalls and sophisticated public-key infrastructures. However, certain security policies were flawed together with a breakdown of certain office procedures which resulted in employees releasing login id's and passwords over the telephone.

Computer system security necessarily must involve policy, enforcement mechanisms, technology, physical security and human factors.

References

[Brow 95] Brown, Carol E., and Sangster, Alan, "Electronic Sabotage", http://www.bus.orst.edu/faculty/brownc/lectures/virus/virus.htm, January 8, 1995.

[CERT 97] Firth, Robert, et al, Detecting Signs of Intrusion, http://www.sei.cmu.edu/pub/documents/sims/ps/sim001.ps, Security Improvement Module CMU/SEI-SIM-001, August 1997.

[CERT 97] Firth, Robert, et al, Security for a Public Web Site, http://www.sei.cmu.edu/pub/documents/sims/ps/sim002.ps, Security Improvement Module CMU/SEI-SIM-002, August 1997.

[CERT 98] Allen, Julia, et al, Security for Information Technology Serviec Contracts, http://www.sei.cmu.edu/pub/documents/sims/ps/sim003.ps, Security Improvement Module CMU/SEI-SIM-003, January 1998.

[CERT 98] Kochmar, John, et al, Preparing to Detect Signs of Intrusion, http://www.sei.cmu.edu/pub/documents/sims/ps/sim005.ps, Security Improvement Module CMU/SEI-SIM-005, August 1998.

[CERT 98] The Official CERT Web Site, http://www.cert.org, 1998.

[CERT 98] CERT Incident Note IN-98.02, http://www.cert.org/incident_notes/IN-98.02.html, July 2, 1998.

[Come 95] Comer, Douglas E., *Internetworking with TCP/IP Volume I: Principles, Protocols, and Architecture*, 3rd edition, Prentice Hall, 1995.

[Come 96] Comer, Douglas E., *Internetworking with TCP/IP Volume III, Client-Server Programming and Applications for the BSD Socket Version*, 2nd edition, Prentice Hall, 1996.

[Come 97] Comer, Douglas E., *The Internet Book: Everything You Need to Know About Computer Networking and How the Internet Works*, 2nd edition, Prentice Hall, 1997.

[Come 98] Comer, Douglas E., *Internetworking with TCP/IP Volume II, ANSI C Version: Design, Implementation, and Internals*, 3rd edition, Prentice Hall, 1998.

[Ferg 98] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks Which Employ IP Source Address Spoofing", Network Working Group, RFC 2267, January 1998.

[Gard 98] Gardner, Dana, "E-mail Bug Stirs Up a Scare", Infoworld, pp 20, August 3, 1998.

[GNU 91] The Official Linux Web site, http://www.fsf.org/copyleft/COPYING, 1991.

[Gram 84] Grampp, Fred T. and Morris, Robert H. Morris, "Unix Operating System Security", *AT&T Bell Laboratories Technical Journal*, Volume 63, Number 8, Part 2, pp 1649-1672, October, 1984.

[Haen 96] Haeni, Reto, "An Introduction to Information Warfare", http://www.seas.gwu.edu/student/reto/infowar/info-war.html, August 23, 1996.

[INFO 98] McClure, Sturat, "E & Y Teaches the Fine Art of Hacking at your Site", Inforworld, pp 88, July 27, 1998.

[Kers 98] Kerstetter, Jim, "Human Side of Hacking", PC WEEK, pp 6, August 3, 1998.

[Linu 98] The Official Web Site for the Linux Operating System, http://www.linux.org 1998.

[Morr 79] Morris, Robert H., and Thompson, Ken, "Unix Password Security", *Communications of the ACM*, Volume 22, Number 11, pp 594-597, November 1979.

[Mudg 97] Mudge, Peter and Benjamin, Yobie, "Deja Vu All Over Again", Byte Small Systems Journal, Vol. 22, No. 11, pp 81-86, November 1997.

[Negr 95] Negroponte, Nicholas, *Being Digital*, Alfred A. Knopf, 1995.

[Park 83] Parker, Donn B., *Fighting Computer Crime*, Charles Scribners' Sons, 1983.

[Shim 96] Shimomura, Tsutomu with Markoff, John, *Takedown*, Hyperion, 1996.

[Spaf 88] Spafford, Eugene H., "The Internet Worm Program: An Analysis", Purdue University Computer Science Department Technical Report CSD-TR-823, 1988.

[Spaf 89] Spafford, Eugene H., "The Internet Worm Incident", Proceedings of the 1989 European Software Engineering Conference (ESEC 89), Springer-Verlag, Number 87, in *Lecture Notes In Computer Science* series, 1989.

[Stol 89] Stoll, Clifford, *The Cuckoo's Egg*, Doubleday, 1989.

[Stol 95] Stoll, Clifford, *Silicon Snake Oil*, Doubleday, 1995.

[Udel 97] Udell, Jon, "The Value of Free Software", Byte Small Systems Journal, Vol. 22, No. 12, pp 109-112, December 1997.

[Veri 98] The Official VeriSign Web site, http://www.verisign.com/, 1998.

[Walk 98] Walker, Christy and Moeller, Michael, "E-Mail Full of Holes", PC Week, pp 8, August 3, 1998.