UTnet Guide to UNIX System Security

Charles Spurgeon
The Network Information Center
of the University of Texas at Austin (UT NIC)

Document version 2.0

Introduction

Every computer user and system administrator needs to be aware of three facts about security for computers attached to the campus network system:

1. —The moment a computer system is attached to the campus network it is vulnerable to system crackers, also called ‘‘hackers’’ in the news media. Call them what you will, the chances are good that a system cracker will try to break into your computer system once it is attached to the network. If they break into your system, they may install special software that will compromise the security of your system.
2. —Vendors ship their computer systems with major security holes. Typically a system comes from the factory configured as though it will never be attached to a public communications system. Until you configure your system to be secure, it will be vulnerable to attack by system crackers armed with lists of well known security holes for each vendor's system.
3. —You are on your own. It is up to you to ensure the security and integrity of your private files, and of the computer systems you may be responsible for.

System administrators and general computer users alike need to know that computer crackers are increasing in number, and that the crackers are equipped with the latest information about security holes in computer systems. The rest of this document is a guide to resources that can help establish and maintain UNIX[1] system security.

---

[1]UNIX is a registered trademark of AT&T.

1. Guide to Resources for System Security

What follows is a list of resources that can help establish system security on host systems. The publicly available resources listed here are slanted toward the UNIX operating system, which reflects the fact that UNIX is widely used on campuses. Several items listed here are available at no charge as files that can be copied from a Computation Center computer. Instructions for retrieving these files are included at the end of this list.

The primary guide to system security for your host is the documentation that came with the system. Each vendor's system is different, and the vendor must be relied upon for the most accurate description of how to establish security for their system.

On the other hand, system security information can be scattered throughout the documentation, and vendors rarely include tutorial information that adequately describes the scope of the security task. The following documents can help make sure that you don't miss any important security issues as you configure your system.

1.1. System Security Documents

- **Improving the Security of Your UNIX System**
  David A. Curry
  April 1990, Technical Report ITSTD-721-FR-90-21, from Information and Telecommunications Sciences and Technology Division, SRI International.
  51 pps, including references and suggested reading list.
  Available on host *emx.utexas.edu* in the directory *˜ftp/pub/netinfo/docs* as *unix-secure.txt*, an ASCII text file, and *unix-secure.ps*, a PostScript file suitable for printing on an Apple LaserWriter or equivalent.

An excellent guide to the subject of UNIX security. The issues are clearly explained, and many examples are included for improving security in UNIX systems in general and Sun systems in particular. Also included is a section on monitoring the security of a system over time, along with a section on publicly available software for improving security.

From the Introduction:

‘‘The UNIX operating system, although now in widespread use in environments concerned about security, was not really designed with security in mind. This does not mean that UNIX does not provide any security mechanisms; indeed, several very good ones are available. However, most ‘out of the box’ installation procedures from companies such as Sun Microsystems still install the operating system in much the same way as it was installed 15 years ago: with little or no security enabled.’’

‘‘This document describes the security features provided by the UNIX operating system, and how they should be used. The discussion centers around version 4.x of SunOS, the version of UNIX sold by Sun Microsystems. Most of the information presented applies equally well to other UNIX systems. Although there is no way to make a computer completely secure against unauthorized use (other than to lock it in a room and turn it off), by following the instructions in this document you can make your system impregnable to the ‘casual’ system cracker, and make it more difficult for the sophisticated cracker to penetrate.’’

- **UNIX System Administration Handbook**
  Evi Nemeth, Garth Snyder, and Scott Seebass.
  Prentice Hall, Englewood Cliffs, NJ. 1989. 593 pps. with index and bibliography.
  ISBN 0-13-933441-6.
  $32.00

Your vendor’s documentation is the final word for system administration on your system, but the material can be scattered throughout the documentation, and there is usually little explanation as to why certain system administration tasks need to be done, or how the entire set of system administration tasks fit together. Adequate system administration is essential to preserving the integrity of data and programs, which is why this excellent guide to the entire set of UNIX system administration tasks is listed.

From the Preface:

‘‘Without guidance, it’s easy for a new administrator to lose sight of ultimate goals in a sea of documentation or to inadvertently skip over one or more critical areas through failure to realize that they are important. That is where this book fits in. While we can’t claim that *everything* you need to know is included, we can guarantee that after reading it you will know pretty much what UNIX administration involves and how to go about it.’’

- **UNIX System Security**
  Patrick H. Wood and Stephen G. Kochan
  Hayden Books, Indianapolis, Indiana, 1985. 299 pps. with index.
  ISBN 08104-6267-2
  $34.95.

A comprehensive look at UNIX system security from the point of view of AT&T UNIX System V. The book is divided into three main sections, Security for Users, Security for Programmers, and Security for Administrators. Several appendices contain example security programs and even include a copy of the SUID patent.

From the Introduction:

‘‘The key word when it comes to computer security is *awareness.* Awareness is needed by administrators in understanding how to make and keep their system secure, how to educate users on security, how to perform periodic security audits, and how to set security standards and enforce them. Awareness is also needed by users in understanding how file permissions work so that other users can’t read or destroy their files, how to send files through the network so that they can’t be read by others, how to write secure programs, how to choose good passwords and keep them secret, and the importance of not leaving terminals unattended. Awareness is needed by the users’ and administrators’ management in understanding the need for a company-wide security policy and in allowing their employees to spend time implementing this policy.’’

- **Computer Viruses and Related Threats: A Management Guide**
  John P. Wack, Lisa J. Carnahan.
  NIST Special Publication 500-166
  Available on host *emx.utexas.edu* in directory ˜ftp/pub/netinfo/docs as *nist-secure-guide.txt*, an ASCII text file.

A guide to computer viruses prevention from a system management point of view. The guide contains suggestions for formulating policies for software management and user education to minimize the risk of computer viruses in both multi-user and personal computers.

From the Introduction:

‘‘This document provides guidance for technical managers for the reduction of risk to their computer systems and networks from attack by computer viruses, unauthorized users, and related threats. The guidance discusses the combined use of policies, procedures, and controls to address security vulnerabilities that can leave systems open to attack. The aim of this document is not to provide solutions to the wide range of specific problems or vulnerabilities, rather it is to help technical managers administer their systems and networks such that manifestations of viruses and related threats can be initially prevented, detected, and contained.’’

2. System Security Software

Bug fixes to close security holes in system software are handled by the vendor for your system. Sun Microsystems has made some bug fixes available on the Internet, and this effort is described in the document listed above entitled, ‘‘Improving the Security of Your UNIX System,’’ by David Curry.

There are also several packages for improving system software described in the Curry document. Two of these packages are available as source code on host *emx.utexas.edu.* This software is made available locally as a convenience to system administrators with the necessary skills. No support is available for this software, and no guarantees are made as to its suitability for any purpose. Use at your own risk.

- **npasswd**

  The *npasswd* program is available via anonymous FTP as described below, from host *emx.utexas.edu.* The program is stored as a set of shell archive (shar) files in directory *˜ftp/pub/npasswd.* The program may also be retrieved as a compressed tape archive (tar) file.

The *npasswd* program, developed by Clyde Hoover at the University of Texas Computation Center, is designed to be a replacement for the standard UNIX *passwd* program. The *npasswd* program makes passwords more secure by refusing to allow users to select passwords that can be too easily breached by a system cracker.

From the source code:

‘‘Npasswd is a pretty-much-plug-compatible replacement for passwd(1). This version incorporates a password checking system that disallows simple-minded passwords.''

- **COPS**

  The COPS package is available as a set of shell archive (shar) files in UNIX ``compress'' format, on host *emx.utexas.edu*. The files may be retrieved from the *~ftp/pub/netinfo/src/cops* directory via anonymous FTP as described below. The COPS package is also available from host *ftp.uu.net* as described in the David Curry document, above.

The COPS package is a collection of shell scripts and C programs that will run on most UNIX systems. The package is a tool for system administrators that checks for many common security problems in UNIX systems.

From the documentation:

``COPS is a security tool that is useful to system administrators, system programmers, or for anyone who would like to learn about UNIX security. It does not restrict a system's environment by placing constraints on activity; it is a purely diagnostic tool that checks and reports on the current status of a given UNIX machine.

Written in Bourne shell, generic commands (awk, sed, etc.) and some C, the system is basically a shell script that runs several small security programs. Theoretically (at least), it attempts to find the following problems (among others) on a generic UNIX system, and then mails or saves the results, if indeed any problems do exist:

```
 --Checks /dev/*mem and all devs listed in "/etc/fstab" for world
   read/writability.
 --Checks special/important directories and files for "bad" (world
   writable, whatever) modes.  (/etc/passwd, /bin, etc.)
 --Checks against /etc/passwd for crummy passwords (user selectable, it
   can be as vigorous or as lax as you wish.)
 --Checks /etc/passwd for non-unique uids, invalid fields, non-numeric
   user ids, etc.  Also includes a password checker.
 --Checks /etc/group for non-unique groups, invalid fields, non-numeric
   group ids, etc.
 --Checks all users' home directories and their
   .login/.cshrc/.rhosts/.profile/etc. files
 --Checks all commands and paths listed in /etc/rc* and crontabs for
   world writability.
 --Checks for bad root paths, world exportable NFS systems, some other
   misc stuff.
 --Includes the Kuang expert system.  Written by Robert Baldwin, this
   basically checks to see if a given user (by default root) is
   compromisible, given that certain rules are true.  Kind of hard to
   explain in a sentence, but worth the price of admission.
 --Checks the system for _changes_ in SUID status.  This is the one (the
   only) program that should be run as superuser, because it runs a
   "find" on all SUID programs from the / directory, and then uses that as
   a reference file for future runs.''
```

3. Retrieving Files From emx.utexas.edu

Several of the items listed above are available as files that can be retrieved from the Computation Center host system *emx.utexas.edu.*

3.1. Copying Files From An emx Account

If you have an account on emx, then you can login to *emx* and simply copy the files from the publicly available directories listed above, using the UNIX copy command ‘‘cp.’’

3.2. Electronic Mail

If you don’t have an account on *emx,* but have access to electronic mail on a network connected system, you may retrieve netinfo documents from *emx.* This is done by sending electronic mail to the mail-based server located on *emx.* The electronic mail-based netinfo server will send you copies of the documents available in the netinfo files on *emx.* The source code is not available through the netinfo server.

The mail-based server is a simple program that reads commands from messages sent to it, and responds by sending back the information requested. For instance, the fastest way to get more information about the server would be to send electronic mail to *netinfo@emx.utexas.edu,* with the word

help

in the subject line or the body of the message. The netinfo server will respond with a list of the commands that it supports.

You can obtain a list of all the documents available in the *docs* directory on emx.utexas.edu by sending the command

send index docs

in the subject line or the body of an electronic mail message. Remember that the netinfo server is a simple program. Do not send any other verbiage in the message and do not use quotation marks around the commands you send. Finally, don’t repeat the same command in a single message. In other words, if you put the command, ‘‘send index docs’’ in the subject line of your message (which is fine as far as the netinfo mailer is concerned) don’t put the same command in the body of the same message.

3.3. Anonymous FTP

You can retrieve all of the documents in the *netinfo* directories through a mechanism known as ‘‘anonymous FTP.’’ FTP stands for *file transfer program,* which is an application program based on TCP/IP protocols. The FTP program makes it possible to retrieve files from any host that has a network connection and is running the FTP software. The anonymous FTP mechanism makes it possible for anyone to retrieve files from certain special directories on a remote system, whether or not they have a login account on that system.

To open an anonymous connection simply type,

ftp emx.utexas.edu

and hit the return key. The FTP program will connect to the emx system and issue a login prompt. At the login prompt type,

anonymous

and hit the return key. The FTP program will prompt for a password. At the password prompt type your username and machine name. Actually any password may be used, but typing your username and machine name is a courtesy to the administrators of the remote host.

After opening an anonymous FTP connection to emx, you can ‘‘cd’’ to the directory containing the files you want. At that point, you can simply retrieve the specific files you are interested in.

Example:

Here’s an example of using the FTP program to log into host *emx.utexas.edu:*

```
% ftp emx
Connected to emx.utexas.edu.
220 emx FTP server (Version 4.161 Wed Apr 5 12:57:18 CDT 1989) ready.
Name (your-host:your-name): anonymous <-type "anonymous" here
331 Guest login ok, send ident as password.
Password: <-type name and host, no echo from emx
230 Guest login ok, access restrictions apply.
ftp> cd pub <-change directory to "pub"
250 CWD command successful.
ftp> cd netinfo
250 CWD command successful.
ftp> cd docs
250 CWD command successful.
ftp> get unix-secure.ps <-request PostScript version of unix-secure doc
200 PORT command successful.
150 Opening data connection for unix-secure.ps (ascii mode) (274267 bytes).
226 Transfer complete.
local: unix-secure.ps remote: unix-secure.ps
295115 bytes received in 8.1 seconds (36 Kbytes/s)
ftp> quit
221 Goodbye.
```