Using Secure Data Servers To Maintain Data Integrity

Gary Hull
NASA Automated Systems Incident Response Capability (NASIRC)
Hughes STX Corporation
7701 Greenbelt Road
Greenbelt, MD 20770-2037
Phone +1-301-441-4242
FAX +1-301-441-1853
ghull@flyfish.stx.com

Nicki Fritz
NASA Automated Systems Incident Response Capability (NASIRC)
Hughes STX Corporation
7701 Greenbelt Road
Greenbelt, MD 20770-2037
Phone +1-301-441-4084
FAX +1-301-441-1853
fritz@nasirc.nasa.gov

Abstract

Currently accepted methods for securing and ensuring data integrity primarily focus on those methods implemented on the computer and data storage devices where the data itself is maintained. Typical methodology may include password protection of user accounts and their data files, file owner and user permission flags, and/or assignment of user/group accounts by the system administrator. While those mechanisms proved effective in the past, those techniques are insufficient, if not impractical, now that data accessibility has increased through worldwide use of the Internet.

The growing need to make data available to a diverse group of researchers has lead NASA to use the Network File System (NFS), mail servers, name servers and web servers to facilitate information access and exchange. NFS allows a client workstation to perform transparent file access over the network through the use of remote procedure calls (RPC) that are built on top of external data representation (XDR) protocol. While the RPC protocol provides for exchange of version and authentication or parameters as mechanisms for security, this "trusted system" concept has proven vulnerable by individuals intent on gaining illegal access to the NFS server system. The impact of a system break-in by a "hacker" can sometimes be relatively minor, or it can be devastating. However, in every case, the integrity of data is compromised.

Several tools can protect the integrity of data on publicly accessible data servers. Ideally, a machine that contains valuable or sensitive material should not be connected to the Internet. Since that is not always practical, the installation of firewalls, tcp wrappers and other security tools can enhance data protection. This paper provides an overview of security tools which NASIRC has found to significantly improve data security on systems accessible from the Internet. Security tools which can filter, audit and monitor network traffic and system activity will be discussed. Utilities such as TIGER and COPS will be presented as examples of tools used to detect and clean up security problems.

Computer Security and the NASA Environment

NASA supports and provides eleven high- intensity computing research centers nationwide. The predominant operating system used at each of these sites is UNIX. Interestingly enough, UNIX was not originally designed with any security in mind. It was developed by two computer programmers to satisfy their own need for data sharing with other programmers in an open environment rather than one of privacy.

The bulk of NASA's research efforts are dependent upon this premise of data sharing in a cooperative, open environment, which is facilitated by the UNIX operating system. Since its early days, UNIX has matured considerably and now provides many security mechanisms designed to work within an environment that fosters worldwide sharing of data.

All NASA sites nationwide are connected and have the ability to share data and information via the Internet. UNIX and its inherent networking features have contributed greatly to the successes of NASA, however, the expanding capabilities of this open network environment continue to reveal vulnerabilities to the systems and shared networks.

The sudden popularity of "surfing the net" by casual computer users has added a new dimension to the need for enhanced computer security. Hackers accessing the Internet are able to easily identify NASA's high- performance computing systems and feel challenged to gain illegal access to them. The majority of computer break-ins are to UNIX-based machines, but all operating systems are vulnerable and subject to "cracking."

Contrary to popular belief, hackers do not operate in isolation. The hacker community is well established: hackers use the Internet to share their "cracking" tools and techniques, as well as to publish their successes. They maintain hacker bulletin boards and Mosaic home pages, and effectively monitor accesses to shut out users from outside their closed community. In doing so, these hacker groups often use the same tools the computer security community uses to protect vital systems. In some cases hackers break into a system, create their own "invisible" directories, populate them with illegal copies of licensed software, publish their locations and disseminate that software to other eager hackers, all underneath the noses of site administrators.

The NASAÊAutomated Systems Incident Response Capability (NASIRC) was organized to maintain the highest measure of data and system security, while also insuring the open systems environment NASA has become so dependent upon. Several highly effective security measures identified and recommended by NASIRC for maintaining data integrity and detecting break-ins are discussed in this paper. Many of these measures are currently in use at several NASA sites and have proven effective in deterring unauthorized access to systems where critical data is stored.

The security methods recommended by NASIRC and discussed in this paper can be implemented in any research facility dependent upon open networks to exchange data. Following is an overview of the basic security measures typically used in such research environments, their inherent weaknesses and how they can be augmented and strengthened through newer approaches, utilizing the latest in security techniques, tools and utilities.

Password Protection

The first level of security for all systems in a research environment is password protection. However, the manner in which passwords are used is often self-defeating. Although this is still a highly recommended practice, it alone cannot protect a system from an intruder. Quite often users select passwords that are not difficult to guess or "crack," or they use the same password across multiple accounts or systems. If an intruder wishes to gain access to a particular system, the first step is to run password cracking software against the user accounts. Cracking programs abound on the Internet as shareware, freely available to any would-be hacker. If the user account

password is a common word or name, chances are good it can be guessed. Once the account password is acquired, the intruder is "in" and has access to all data files, unless those files are protected with their own passwords. But, even novice hackers can quickly and easily crack those file passwords.

After gaining access, some intruders may simply "browse" to see what is in each file. For some hackers, the thrill of actually breaking into a system is its own reward and no data is destroyed, copied or altered. The intruder bent on acquiring specific data may copy and export files to another system; this is easily accomplished by use of ftp. In the case of an inept intruder, files may be inadvertently destroyed or irreparably damaged, and a particularly malicious intruder may go on to wreak havoc on an entire system. On some occasions, when intruders suspect their illegal access has been discovered, they will attempt to "cover their tracks" and in the process vindictively destroy critical system software and/or data.

In the NASA environment, as in many others connected to a LAN or WAN, the intruder who manages to break into a user account can then gain access to that user's directories and potentially the entire networked system.

The Internet was initially designed to connect users in research centers worldwide. With the ever-increasing popularity of the Internet today, access is now available to even the most novice home computer user. The far reaching impact is that anyone with a modem and some elementary hacker tools can potentially gain unauthorized access to critical or sensitive data almost anywhere in the worldwide web (WWW). As the number of people "surfing the net" continues to grow exponentially, the computing industry and especially research and development centers like NASA have come to realize the need for more reliable security to protect the integrity of their data.

Network and Data Servers

Network File System (NFS) allows hosts to share files over the network. A common way of using NFS is to install workstations in users' offices with the data on one main disk server. File systems can then be mounted across the network, allowing information to be accessed as though it resided on each individual computer. Fresh out of the package, NFS generally has no security features enabled, which means that any host on the network can access files via NFS regardless of whether they are identified as "trusted hosts" or not. Later versions of NFS have been made more secure (i.e., Secure NFS, available in SunOS version 4.0 or later). Secure NFS makes use of public- and private-key encryption techniques to verify authorized access.

NFS can also be used in conjunction with **Kerberos**, an authentication system developed by M.I.T. It is a third party authentication service that is trusted by other network services. Upon logging in, a user is authenticated by a password and Kerberos provides the user a way to prove his/her identity to other servers and hosts on the network. This authentication is also used by the e-mail system to guarantee that e-mail is delivered to the right person, as well as to guarantee that the sender is who he claims to be. NFS has been modified by M.I.T. to work with Kerberos, which makes it much more secure.

Firewalls

In the computer security arena, the idea of building "firewalls" is a fairly new technique to protect a private network against intrusion. When properly configured, a firewall provides a barrier of protection between an individual network and the outside world by providing strict access control.

J.P. Wack and L.J. Carnahan define a firewall system as "a router, a personal computer, a host, or a collection of hosts, set up specifically to shield a site or subnet from protocols and services that can be abused from hosts outside the subnet." [1]

In a typical firewall system, the IP router (or gateway) is replaced by a multi-homed host that does not forward packets. Because the IP packets are not forwarded, the connection between the protected network and the outside world is effectively severed by the firewall. A firewall located at a higher-level gateway (i.e., Internet connection) isolates an entire network. This level of security is not always necessary since individual workstations and desktop computers may not be used to store critical or sensitive data and applications. As is the case with NASA, many organizations may find that only a few specific computer systems require this high degree of protection. In those instances, a firewall can be placed at lower-level gateways to isolate and protect specific subnets (i.e., high performance computers, NFS).

A firewall is designed to filter and limit entry and exit of specific Internet protocols, such as e-mail and ftp, to only approved IP addresses. The firewall software does not screen or filter individual users, but rather Internet addresses outside the range of configured IP addresses. In setting up a firewall, parameters can be set to allow or disallow Internet protocols by name. Some services can be restricted to outgoing connections only; others to be received only by machines on the local network; further limitations can be made to specific machines on the local network.

Since address ranges and protocol types can be identified, firewalls can also log and report unauthorized attempts to breach the system. "A firewall with appropriate *alarms* that sound when suspicious activity occurs can also provide details on whether the firewall and network are being probed or attacked." [2]

In one sense, firewalls are the sacrificial lamb for the network. Firewalls are not foolproof, but by utilizing strict security measures on the firewall machine(s) the path to the larger network is guarded. Determined intruders are first blocked by the firewall and cannot directly attack any systems behind it. What a firewall cannot do, however, is prevent a break-in from within the local network. A user who is normally authorized access via a named IP address is behind the firewall, and thus his activities cannot be screened by the firewall should he attempt any unauthorized accesses elsewhere in the network.

The drawback to firewalls is that by restricting outside access into the local network, they also restrict access from the local network to the outside world. When security is more important than service, a firewall is appropriate.

Firewalls can be configured with many different variables, but should be designed with an eye to providing the needed services users require while not compromising the level of security needed. Many vendors are now offering firewall software to protect their systems; other vendors have begun to offer machines specifically for building a firewall system. One of the better offerings comes from Trusted Systems Inc., a free software product called **FWTK** (Firewalls Tool Kit), which is available from NASIRC.

Another drawback of firewalls is that once installed, they are not self-maintaining. Usually a skilled system administrator is required to initially configure the firewall and maintain the system. Depending on how the system is designed and configured, firewalls can be relatively inexpensive or they can be quite costly. However, they are worth the investment because the financial impact of a single break-in can be considerably more than the initial cost of establishing a firewall. A recent study by NASA's Jet Propulsion Laboratory found that the annual financial impact due to break-ins for each NASA center could approach $2.4 million, yet the installation of adequate security measures (such as firewalls) to prevent those break-ins would have cost approximately $500,000 annually.

TCP Wrappers

Like firewalls, tcp wrappers can control access to network services. However, unlike firewalls, tcp wrappers also offer the unique ability to log all requests for Internet services. This latter capability adds a real "monitoring dimension" to the system and lets the administrator "see" who is using the network. The greatest advantage of tcp wrappers is that there is virtually no impact on authorized computer users.

Installing a tcp wrapper is simple and does not require any change to existing system software. All that is required is a simple change to the /etc/inetd.conf file.

A tcp wrapper runs a daemon service called *tcpd*, which is designed to provide access control of other named Internet daemon or services. Typical services that can be monitored by tcp wrappers are those normally started by the *inetd* system daemon including: *fingerd, ftpd, tftpd, telnetd, rshd, rlogind* and *rexecd* daemons. Monitoring of these services is accomplished by placing *tcpd* in the same directory path as the other Internet service daemon, then editing the /etc/inetd.conf file and replacing the path to each network service daemon with *tcpd.* For example, to monitor the finger command, change the following entry in the /etc/inetd.conf file.

Default Entry

finger stream tcp nowait nobody /usr/etc/in.fingerd in.fingerd

tcp wrapper Entry

finger stream tcp nowait nobody /usr/etc/tcpd in.fingerd

When the *inetd* receives a request for the *fingerd*, it will instead start *tcpd*, which in turn writes the request to the log file before starting the real finger daemon to complete the request.

Access control is initiated in the same way but includes one additional step. Data in the /etc/hosts.allow and the /etc/hosts.deny files are referenced prior to *tcpd* passing the request to the service daemon. As their names imply, these two files are used to control access to the network services. Access is determined by specifically naming the network addresses granted access in the allow file and the network addresses not granted access in the deny file. When *tcpd* is started by *inetd*, it not only logs the request but also checks the access control information in these files and permits completion of the request only if the originating address is named in the allow file. In the absence of allow and deny files everyone is granted access to the service on that network.

How monitoring information obtained from a tcp wrapper is processed is up to the creativity of the system administrator. Simple, locally written UNIX scripts can be designed to look for unauthorized log file entries and report them via e-mail messages to the appropriate person. Sophisticated shareware is also available. For example, a tool called **swatch** can be used to activate an "alarm" and dial a phone number when unauthorized access is detected.

A tcp wrapper in combination with a tool like swatch will give the system administrator immediate notice when an intrusion occurs. The unauthorized access can then be monitored while the attack is in progress to either gather data and information on the attack scenario, or to implement immediate protective measures. This is critical because some protective measures can only be implemented while an attack is in progress.

Installation of tcp wrappers is easy and well worth the time and effort to implement. Tcp wrapper software is available to NASA sites via the NASIRC WWW homepage and the NASIRC anonymous ftp, located in the path:

/toolkits/UNIX/TCP_Wrappers/tcp_
wrappers.tar.Z

File System Security

UNIX system security can be divided into three main areas of concern. This paper has already addressed two areas: account security and network security. The third area, file system security, is concerned with preventing unauthorized access by legitimate users or hackers to the data stored in the system.

Routine checking for security holes in the file system is another important part of making any system secure. Files that can be modified or inadvertently grant too many permissions to unauthorized users can in turn open a system up for penetration by hackers. NASIRC recommends using the UNIX *find* command to search for four file system characteristics: setuid and setgid files, world-writable files, unowned files and .rhosts files.

• setuid and setgid files

One common hacker trick is to break into a system, gain root access and leave a setuid program secretly hidden somewhere on the system. That way, even if the originally exploited vulnerability is later secured, the hacker is still able to regain root access. The command to search for setuid and setgid files is find / -type f -a \ (-perm -4000 -o -perm -2000 \) -print. This command can take as little as fifteen minutes or as long as two hours to run, depending on the characteristics of the system. The generated list of files should be closely examined to determine if they need these permissions.

A good example of a file that does not require setuid permissions is /usr/etc/restore, which is released in the SunOS environment with the setuid permissions turned on. However, since there is a security hole associated with this command, the setuid permissions should be turned off. Doing so will have no adverse impact on the use of the restore command. Use the command chmod u-s /usr/etc/restore to remove the setuid bit. When looking at the output of the *find* command, be especially suspicious of files in the following system directories: /bin, /etc, /usr/bin, /usr/ucb and /usr/etc.

• world-writable files

World-writable files give all users write access to those files; anyone who gains illegal access to the system can then modify those files. In addition, a world-writable directory means anyone can add or delete a file from that directory. To locate world-writable files and directories use the command find / -perm -2 -print. Any printer error log files, linked files or /dev files found by use of this command are of no concern.

• unowned files

Files not owned by existing users can be found by running the command find / -nouser -print. Any files located with this command might be a clue that a hacker has gained access to the system.

Even if that is not the case, this gives the system administrator an opportunity to remove old files that should have been cleaned up when the original user was removed from the system.

• .rhosts files

In nearly all systems, no one should have .rhosts files in their accounts because in a trusted system environment they allow a user to gain access to another system on the same network without entering a password. In special circumstances, system administrators may have a need for them to do remote maintenance. In these cases, additional security measures must be taken such as keeping the local host in a secure, locked room. If .rhosts files are used in a trusted system environment this creates a vulnerability which can be exploited by hackers using IP spoofing. Since the .rhosts file exists in the user's home directory, the command to locate these files is find /home -name .rhosts -print.

File system integrity can be periodically checked and monitored by using available shareware utilities. NASIRC recommends the use of **Tripwire** to establish a "baseline database" for a designated set of files. Subsequent checks by Tripwire then compare the same files to the baseline database and report any changes found. Variations found could indicate penetration by an unauthorized user (i.e., if a hacker has gained root access to the system and has installed his own system utility with the same name as an original operating system command).

It's not uncommon for hackers to break into a system and employ a "Trojan'd Telnet" command (known as a sniffer) that then captures users' login and password information. In this scenario, the only recourse is to shut down the network, completely restore the system from the last known secure full backup, and instruct all users to change their passwords. Tripwire is available to all NASA sites from the NASIRC toolkit and can be found in the path:

/toolkits/UNIX/Tripwire/tripwire-1.1.tar.Z

General Security Practices

NASIRC recommends the following steps to establish basic or general computer and network security:

• Install all vendor security patches as soon as possible on all host computers. Vendor patches are located in the NASIRC archive and are accessible from the MOSAIC homepage (located in the path http://nasirc.nasa.gov) or the NASIRC anonymous ftp (nasirc.nasa.gov).

• Implement "one-time passwords" when and where practical. **S/KEY** is a utility available for this purpose and is located in the path ftp://crimelab.com/pub/security/skey/.

• Use "shadow passwords" to effectively hide the password file somewhere on the system.

• Use a "smart passwd command," one that evaluates a password at the time it is selected to determine if it can be easily cracked. Typically users pick passwords they can remember, like their name or phone number; by doing so they also make it easy for a hacker. When a user changes his password, the "smart passwd command" evaluates the new password based on user account information. If it might be too easily guessed, the user is informed and required to select a different password.

• Disable or delete unused and dormant system and user accounts.

• Implement a trusted system environment only when and where absolutely necessary. When used, NASIRC highly recommends all network traffic be filtered and monitored by tcp wrappers.

• Where use of the finger service is required, use a modified finger daemon that does not display the user's home directory and the source of the last login.

• Use NFS sparingly. Export file systems read-only and not unrestricted to the world when NFS is used.

• Protect all servers with tcp wrappers and firewalls.

• Use Network Information Service (NIS) only where absolutely necessary.

• Use only the *inetd* and *portmapper* services required for normal day-to-day operation of each computer system on the network. Monitor and audit the associated network traffic with tcp wrappers.

• Keep abreast of current computer security information, available tools and trends. Share information by communicating with NASIRC regarding security problems and incidents.

Security Utilities

NASIRC recognizes three categories of tools to enhance basic site security. Each category is defined by the function it serves.

Basic Security Implementation

Tools in this group help a system administrator establish a basic level of computer and network security by identifying needed operating system patches and existing vulnerabilities in critical system files. Included in this category are tools like COPS, TIGER and secure_sun.

Monitoring Tools

These tools allow system filtering, auditing and monitoring of network traffic and user activity. Examples are tcp wrappers, Tripwire and SPI.

Intrusion/Detection Tools

Tools in this group are used primarily for penetration testing by system administrators who want to aggressively identify vulnerabilities in a system. In penetration testing, established criteria may require the system be compromised to validate the existence of any vulnerabilities detected. Under those circumstances the tester would then use sophisticated hacker tools against the vulnerabilities to gain system access. Tools such as COPS and TIGER can detect and identify holes in a system an intruder might exploit.

COPS

This security tool for system administrators that checks for numerous common security problems on UNIX systems. Consisting of shell scripts and C programs, COPS can be run on almost any UNIX variant. Some of the things COPS checks:

• /dev/kmem and other devices for world read/writability

• special files and directories for bad modes

• easily guessed passwords

• duplicate user IDs, invalid fields in the password files

• duplicate group IDs, invalid fields in the group file

• all users' home directories and their .cshrc, .login, .profile and .rhosts files

• bad root paths, NFS file systems exported to the world

• whether a given user can be compromised given certain conditions

• changes in the setuid status of programs on the system

TIGER

A set of Bourne shell scripts, C programs and data files used to perform security audits of UNIX systems, TIGER mainly reports on ways systems can be compromised to the root level. In some cases TIGER can correct vulnerabilities found. TIGER is an ideal tool for newly installed UNIX environments and when updated security measures have not been maintained.

secure_sun

Similar to COPS, this tool developed by TAMU is a set of Bourne shell scripts designed to perform a security audit of the SunOS 4.1.x environment. It finds and fixes fourteen of the most common SunOS security holes. This tool could be best applied in a newly installed SunOS environment.

Tripwire

Tripwire was designed to aid system administrators and users in monitoring a designated set of files for any changes. It checks and records file integrity information, then compares future results against this "baseline database" and flags any changes. NASIRC cautions system administrators to be sure their system is "clean" before running Tripwire the first time. If not, the baseline database may reflect system files that have already been compromised, which will then be effectively protected by Tripwire.

SPI

Security Profile Inspector was developed by CIAC, with funding from the U.S. Air Force and Department of Energy, to perform security audits of many UNIX platforms. SPI is similar to Tripwire in that it checks file system integrity, establishes a "baseline database" and monitors system and network traffic. It is also similar to TIGER in that it can detect and fix vulnerabilities. Although it can be configured to run via CRON, the unique advantage of SPI is that it also can run interactively through a built-in user interface to provide "on demand" system security verification. NASIRC recommends this tool for all UNIX environments as part of an overall security verification program.

References

1. Wack, John R. and Lisa J. Carnahan, "Keeping Your Site Comfortably Secure: An Introduction to Internet Firewalls," *NIST Special Publication* 800-10, 1994, pp. 16.

2. Wack, John R. and Lisa J. Carnahan, "Keeping Your Site Comfortably Secure: An Introduction to Internet Firewalls," *NIST Special Publication* 800-10, 1994, pp. 18.

Bibliography

Bellovin, S. M., "Security Problems In The TCP/IP Protocol Suite,"*Computer Communication Review*, Vol. 19, No. 2, April 1989, pp. 32-48.

Chapman, D. Brent, "Network (In)Security Through IP Packet Filtering," *The Third USENIX UNIX Security Symposium*, Baltimore, MD, September 1992.

Curry, David A., "Improving The Security Of Your UNIX System," ITSTD-721-FR-90-21, 1990.

Farmer, Dan and Wietse Venema, "Improving the Security of Your Site by Breaking Into It," *Computer Security Newsgroup*, 1993.

Hunt, Craig,*TCP/IP Network Administration*, O'Reilly & Associates, Inc., Sebastopol, CA, 1992.

Kroeger, Thomas M., "How To Improve Security On A Newly Installed SunOS 4.1.3 System," University of Hawaii Computing Center, July 1994.

Morris, Robert T., "A Weakness In The 4.2BSD UNIX TCP/IP Software," AT&T Bell Laboratories, February 25, 1985.

NASA Automated Systems Incident Response Capability (NASIRC), "Security Tools Catalog," Version 1.0, July 1994.

Nemeth, Evi, Garth Snyder and Scott Seebass, *UNIX System Administration Handbook*, Prentice Hall, Englewood Cliffs, NJ, 1989.
Ranum, Marcus J., "Thinking About Firewalls," Trusted Information Systems, Inc. publication.

Venema, Wietse, "TCP Wrapper Network Monitoring, Access Control, and Booby Traps," Mathematics and Computing Science, Eindhoven University of Technology, The Netherlands, 1993.

CAUTION: NASIRC does not encourage or recommend penetration testing. In cases where it is deemed critically necessary, penetration testing should not be initiated without a thorough investigation and understanding of the practical, ethical and legal ramifications. It is also imperative that written approval by all appropriate levels of NASA management be obtained prior to any penetration testing.