Guarding The Fortress

Efficient Methods To Monitor Security on 300 Systems

Michele D. Crabb (*crabb@nas.nasa.gov*)
- Sterling Software / NASA Ames Research Center

RNS-94-001 April 1994

ABSTRACT

With the increased proliferation of desktop UNIX workstations, computer security has become an increasingly larger headache for sites which have several hundred to over 1000 systems. Most people would agree that keeping a watchful eye on a handful of workstations or mainframes is a simple task. However, keeping a watchful eye on several hundred workstations of differing architectures can be a security administrator's never-ending nightmare, if not done efficiently. Unlike many other types of UNIX system administration tasks, which can be placed on the queue and done at a later date, delaying the completion of a security task, such as the installation of a security patch, could leave a site very vulnerable to an intruder attack. Therefore, in order to adequately monitor and protect a large number of systems and users, security tasks must be well organized, efficient, and automated, if at all possible.

This paper will present an overview of how system security is currently maintained on the over 300 systems at the Numerical Aerodynamic Simulation (NAS) facility at NASA Ames Research Center. The discussion is divided into five areas: why security is needed; what policies and procedures are used; what types of security awareness training is provided; the security monitoring and checking tools that are used, and some future plans for security at the NAS facility.

Introduction

Computer security has been a growing concern in the government and private sectors since the Morris Worm Experiment in 1988. That single incident gave birth to many jobs for UNIX security people, including myself. The UNIX security arena has been growing at a steady pace ever since. UNIX system security in the large heterogeneous environments of the 90s seems to be a security administrator's never-ending nightmare. UNIX system security is no longer just ensuring that all accounts have passwords and restricting who has the *root* password. Instead, system security is now a combination of good system administration, well-defined security policies and procedures, and regular security training programs for both the system support personnel and the user community. Neglecting any one of these three areas can create a weak link. Due to the importance of many security tasks, such as monitoring log files and installing patched versions of binaries, they cannot be delayed like many other system administration tasks. Therefore, in order to adequately monitor and protect a large number of systems and users, security tasks must be well organized, efficient, and if at all possible automated.

A prerequisite to gaining efficiency is the knowledge and understanding of the task at hand. In the last several years, there has been a large number of books and papers published on UNIX security, and how to improve security at a site. There has also been a large amount of security related software that has been published on the Internet. With the use of these tools, anyone can effectively monitor and manage security in a large, heterogeneous environment.

The NAS environment is comprised of over 300 computer systems running some flavor of the UNIX operating system. The NAS systems are interconnected into a heterogeneous network using various types of networking hardware and are supported by over 100 personnel, who provide ongoing support and software development. These systems are of differing architectures which include Sun SPARCstations, SGI 4D workstations, SGI 4D/480 multi-user systems, Cray Y-MP C90s, Convex C-3240s and several massively parallel processing machines. The NAS facility, which primarily provides supercomputing services to other NASA sites, large aerospace corporations and a large number of universities, has over 1500 users nationwide. The security on these systems is primarily monitored and maintained by one full time employee. The main objectives of computer security at the NAS facility are to ensure:

- Security meets all NASA headquarters and Ames guidelines and requirements.
- The support staff is capable of responding to a security incident in a timely and orderly manner.
- An adequate level of security monitoring is performed on all systems.

Prior to entering my current position as the NAS security analyst, there were no formal written policies or procedures for maintaining security at the facility. System security was seen as a side task for the system administrators. During my first two years in the position, my main focus was to develop a framework for providing security. The challenge in the task was many methods and procedures which work for a small number of systems do not scale well to several hundred systems. As I began to tackle this monestrous tasks, I saw seven major areas of security which I needed address. These areas are:

- Network access monitoring and restriction
- Files system auditing
- Password validation and management
- Special access management
- User Account management
- User awareness and training
- Policies and procedures to make it all work

The discussion focuses on how these areas are addressed at the NAS facility. First I discuss some of the motivators for providing good system security. Next is a brief discussion on the importance of a security awareness, which includes a description of the NAS program. Then I examine specific security concerns and how they are addressed. This section of the paper includes a discussion of the many security tools used at NAS perform such tasks a network access monitoring and file system administration. Finally, I describe some of the future plans for security at the NAS facility.

Strong Motivators for Good Security

There are a number of reasons why UNIX system security has continued to grow in popularity and importance over the past five years. Prior to the Morris Worm Experiment of 1988, little attention was given to the subject of UNIX security. However, that single event, which brought the Internet to a grinding halt coast to coast in under three hours, made the UNIX world stand up and take notice. Since then, UNIX security has blossomed into a recognized profession.

One of the paramount forces behind the continued growth of the field is the massive proliferation of UNIX systems throughout the nation. As larger numbers of sites acquire UNIX systems, two distinct patterns emerge. First, more and more information is made available on-line. The wealth of information on the Internet is virtually endless. Even the White House is "on-line" these days. As more and more information becomes available on-line, people want access to this information. In some cases, the desire for access to information is for illegal reasons. The second pattern which emerges is the increased popularity of UNIX has led to an established underground community of Internet intruders. These are people who view the Internet as an endless playground. They range from juvenile computer kids who intrude for fun, to professional technology embezzlers who intrude for profit. Most of the work done in the UNIX security world over the last several years has been to provide better protection for on-line information and to provide better methods to keep the Internet intruders at bay.

One of the strongest motivators for good system security at many sites is the need to protect information from disclosure, modification or destruction. The information on the systems may range from student or employee information to vendor marketing analysis and strategies. At the NAS facility, our users' data ranges from C and Fortran source programs to experimental data. While most of this information is not classified, there is a level of sensitivity and a need for protecting the information. Furthermore, NAS users expect their data and information to be safe on our computers.

Another strong motivator for system security is the need to meet regulations and requirements from within the organization. Almost all government organizations and many private sector companies have various regulations and requirements for protecting their computing resources. At the NAS facility, we must meet various regulations and requirements from three different levels of the NASA organizational structure: NASA headquarters, NASA Ames, and our own facility. Without formalized regulations and requirements from higher levels within the organization of a facility, many security procedures and measures may never be put in place.

Finally, another motivator for providing good system security is previous experience with Internet intruders. Unfortunately, many facilities give little thought to computer security until a large-scale incident occurs. A prime example of this phenomenon was the Morris Worm Experiment. Prior to that incident, the NAS facility did not have a position dedicated to computer security. I have heard similar stories from many system administrators I have spoken with at various UNIX conferences.

Policies and Procedures Needed For Efficient Security

Policies and procedures are key elements of any successful organization. Security policies and procedures are especially important as they define the rules to live by and the penalties for breaking those rules. Security policies also help identify security threats and vulnerabilities. They will outline what behavior is, and is not allowed, on the system. Security policies may even help in prosecuting intruders or serious violators of the policies. There are some basic elements that all security policies should include. The most important element is which actions are allowed and not allowed in the local computing environment. For example, are users allowed to play electronic games? Are users allowed to run password crackers on the system? Security policies should also define what actions will be taken against people who violate the policy. For example, will a user's account be disabled because the user was running a password cracking program on the system? Security related policies should also outline the rights and responsibility of the users as well as the system administrators. All policies should be made available to all people who are effected by them.

The NAS facility has a variety of security policies already in place and several new policies in the acceptance process. The intent of all NAS policies is to ensure that computer security on NAS computing resources is maintained according to NASA regulations and policies, while at the same time not impeding the ability of NAS users and support staff to perform their work. All of the NAS computer security related policies are described below.

The first policy is the *Acceptable Use Statement for NAS Systems Division Computing Resources*. The purpose of this policy is to increase the awareness of computer security issues and to ensure all NAS users (scientific users, support personnel and management) use the NAS computing resources and facilities in an efficient, ethical and lawful manner. The policy is a collection of twelve rules. Every NAS user is required to sign this agreement as part of their account approval process. The policy is very explicit about how violations will be handled. It states "Any noncompliance with these requirements will constitute a security violation and will be reported to the management of the NAS user and the NAS DPI-CSO and will result in short-term or permanent loss of access to NAS Systems Division computing systems. Serious violations may result in civil or criminal prosecution." The full text of this policy can be found in Appendix A.

Closely related to the *Acceptable Use Statement* is the *NAS User Account Policy*. This policy outlines the requirements for accessing or requesting a NAS account. Accounts are only issued to people residing in the United States and only for approved projects. Accounts which are inactive for a period of 90 days or more will be disabled, and after an additional 30 days, the account is archived and removed. Dormant accounts can be a source of major problems if the accounts are discovered and compromised by intruders. Many sites, especially universities, will allow users to keep their accounts when they no longer have a valid reason for the account. I consider this to be a poor security practice and something that should be avoided. At the NAS facility, accounts are archived and removed after the completion of a project.

Due to the complexity and organizational structure of the NAS facility, a large number of support personnel require special access (e.g., *root* access). The next two policies were written specifically to deal with special access. The first is the *Special Access Policy*. This policy also provides a set of requirements for the regulation and use of special access on the NAS systems. The policy provides a mechanism for the addition and removal of people from the special access database and a mechanism for periodic reviews of the special access database. As a part of the policy, users are required to sign the *Special Access Guidelines Agreement*. This agreement outlines the many dos and don'ts of using special access on NAS computers. The other policy is titled *Computer Usage Guidelines for NAS Systems Division Personnel*. This is one of the newer policies at NAS and is still in the final approval stage. The purpose of this policy is to establish usage guidelines for support staff with *root* access and is intended to protect the rights and privacy of NAS Systems Division clients as well as those of the NAS staff. The three major areas covered in the usage guidelines are: how to protect the privacy of clients data when working with them on a problem; how to deal with proprietary information that may be stored on the NAS computer systems; and when and how to perform a security investigation on a person suspected of violating policy.

The next two policies deal with system installation and configuration, rather than system usage. The first is the policy *Regulatory Use of the /etc/hosts.equiv, /etc/hosts.lpd and .rhosts Files*. The purpose of this policy is to provide a set of requirements for regulating the use of the system trust files. The policy outlines the necessary requirements for adding a host entry to the

/etc/hosts.equiv or */etc/hosts.lpd* files. The policy also discusses what type of *.rhosts* entries are allowed. The other system configuration related policy is the *NAS Computer Network Connection Policy*. This policy outlines the requirements and contraints for attaching a computer to the *nas.nasa.gov* domain. The intent of the policy is to ensure that all systems installed on the NAS network are configured and maintained at appropriate levels of security, and the security measures implemented meet NASA regulations and policies as well as any Ames regulations and policies.

In addition to the policies discussed above, there are two very important security procedures implemented at the NAS facility. The first is the *Security Incident Escalation Procedure*. This procedure describes the escalation process for various types of security incidents. The escalation process varies depending upon the severity level of the incident. For example, account sharing is considered a minor (or level 1) incident. If a case of account sharing is discovered during the off hours, it would not be a necessary to contact the security person right away. However, for such incidents as a suspected computer virus, immediate notification of the computer security analyst and other support personnel would be extremely important. The other procedure, which is closely related to the escalation procedure, is the *Security Incident Handling Procedure*. This is a document that every computing facility needs. The purpose of a *Security Incident Handling Procedure* is to provide reasonably detailed instructions on how to handle the various types of security incidents. The NAS procedure outlines the areas or responsibilities for support staff, lists some general procedures, and provides detailed instructions on how handle specific types of incidents. To simplify the response process, security incidents are classified into four areas, and each area is handled differently.

Some of the NAS policies were implemented as the result of various security problems. The *Acceptable Use Statement* was prompted by several incidents where employees were reprimanded because they were reading people’s email and looking at their files. The NAS *Computer Network Connection Policy* was inspired by a major security incident at the NAS facility which involved a poorly administered workstation that was installed and maintained by a supporting vendor and not by the NAS support staff. Security policies and procedures should be written at the beginning of the life cycle of a computing facility, instead of after the fact, or as the result of a security incident. All of the policies described in this section are policies and procedures that are appropriate for any large computing facility.

Security Awareness

In addition to having a set of well-defined security policies and procedures, a successful computing facility needs to implement a security awareness program. One of the major objectives of a security awareness program is to ensure that the users, management and system administrators understand the different roles they play in the overall security of the facility, as well as the policies and procedures that pertain to them. If the users understand the security issues and the motives for implementing the different policies and procedures, they are more likely to adhere to the policies and report suspicious system activity.

A security awareness program can consist of hardcopy reading materials, on-line information, training classes or training videos. All users at a facility should be informed of, and provided access to, all security-related policies and procedures when they first begin their employment. On-line access to information is preferred over hardcopy, since hardcopy materials are usually misplaced or discarded. System support personnel should be provided training on an annual or bi-annual basis. The training provided for the support staff should include a more technical overview of security so they can play an active role in maintaining the security of the facility.

The security awareness program at the NAS facility consists of hardcopy reading materials, on-line information, training classes, and a training video. All of our security-related policies and procedures are available on-line and hardcopy to NAS users. They can view the information on-line using the *gopher* or *Mosaic* browsers. The *NAS User Guide*, a document made available to all NAS users, contains a separate section on system security at NAS. The information available in the *NAS User Guide* includes: the *NAS Account Policy*; tips on selecting and maintaining passwords; setting files permission; auditing of users home directories and environment files; using *.rhosts* files and workstation console security. All new NAS users are made aware of this information when they receive their new account notification. If the need arises to inform users of important security information, a bulletin is posted to our on-line information systems and a note is made in the system MOTD. If a limited number of users are involved, the users are contacted directly via email or a phone call. The security awareness program for the support staff also includes an annual extensive full-day tutorial on UNIX system security. Enrollment is open to all interested persons. The class is held locally at the NAS facility. All members of the NAS support staff are required to attend the class at least once. The full-day class has been video taped and is available to NAS users and NASA employees.

Methods and Tools For Security Monitoring

Over the last several years, there has been a wide variety of security-oriented software which has been made freely available on the Internet. This section discusses some of these tools, as well as some locally developed tools which aid in the security monitoring of the NAS facility. Other available tool are described briefly. The discussion is oriented around security functions and the tools which aid in the monitoring of that specific function.

Our first line of defense at the NAS facility is host-based filtering and monitoring of all network connections (e.g., *rlogin* and *telnet*). The use of a host-based filtering mechanism was prompted by a very large security incident which occurred at the NAS facility in late 1991. The NAS computer systems were the victim of repeated intruder attacks over a two month period. After the third successful break-in, a decision was made to install the *tcp_wrapper* software (i.e., *tcpd*) to provide host-based filtering. In order to reduce the number of NAS hosts which could be reached from the Internet, a filtering scheme was implemented which only allowed remote sites to connect directly to a few of our systems. The NAS workstations were configured to only accept connections from hosts within the local domain. Our main philosophy behind this scheme was to reduce the scope of the problem by disallowing remote connections to the over 300 NAS workstations.

```
[helmut.fim.wpafb.af.mil][3]
Feb15 08:48:43[rsh]           moon.nas.nasa.gov
Feb15 10:24:57[rlogin]        venus.nas.nasa.gov
Feb15 10:38:10[telnet]        moon.nas.nasa.gov

[hercles.dt.navy.mil][2]
Feb15 11:20:11[ftp]           moon.nas.nasa.gov
Feb15 11:20:29[ftp]           mars.arc.nasa.gov

[hertz.risc.rockwell.com][1]
Feb15 07:33:27[telnet]        moon.nas.nasa.gov

[hot.cray.com][2]
Feb15 07:03:30[ultra.ftp]     foobar.nas.nasa.gov
Feb15 08:23:23[ftp]           vee.unf.edu.fi
```

Figure 1: Excerpt from a *tcpd* connection log

The *tcpd* program is configured to log all network connections which use daemons started by *inetd*. By reviewing the connection log files, and looking at the patterns of connections and the time the connections occurred, it is possible to spot suspicious activity. All remote connections which are filtered using *tcpd* are logged on the local host and well as the central security host. On the central security host, the *tcpd* connections are logged to the console window as well as a log file. We have a locally written suite of perl scripts which reduce the *tcpd* log files to a more readable form. See *figure 1* for a typical excerpt from one of the processed log files. The first line shows the name of the remote host which made the connection, and the number of connects (or attempts). The next lines show the time stamp of the attempt, the type of connection (e.g., *rlogin*) and the local host which received the connection.

There are several other programs available for host-based filtering. The *in.gate* program works in a similar manner as the *tcp_wrapper* software, in that it provides control over which hosts are allowed to use the services provided by *inetd*. The *in.gate* program also uses a separate configuration file like *tcp_wrapper.* There are also replacement programs for the *telnet*, *rlogin*, and *ftp* programs which filter connections as well. In a non-firewall environment, I highly recommend some form of filtering, either host-based or router-based, or a combination of the two.

Our next line of defense and monitoring is the extensive use of system logging information via *syslogd*. The *syslogd* program reads and forwards system messages to the appropriate log files as specified by the */etc/syslog.conf* file. The *syslog*d program has the ability to log many types of messages which range from kernel error messages to system daemons messages (e.g., messages from *ftpd*). Using *syslogd* to log important system messages is a crucial part of security. If system activities are not logged, then it is difficult to know what is happening or recognize unusual system behavior. For example, a log file which shows numerous failed login attempts to a single account might indicate someone trying to compromise the account. An audit trail of who became *root* would help trace actions back to a specific person. Log files provide an audit trail of system activity and are very useful in responding to security incidents.

At the NAS facility, we log as much information as possible. Authentication messages, such a failed login attempts and *su* commands, are logged.We also log all successful logins on the SGI systems, which provide this level of logging. Authentication messages for all NAS hosts are logged to the central security host, as well as the local host. On the central security host, we have a locally written program which reads the authentication logs and produces a summary report twice a day. On each NAS hosts, all general types of messages at the emergency level, all alert level messages and all mail related messages are logged to separate files on the local host. These log files are consulted only if needed. All system log files are archived either on a daily basis or a weekly basis, depending on the growth rate of the log file.

Important system messages, such as failed login attempts, *su*s to *root*, and *inetd* connections should be logged to a central logging host, as well as the local host. Intruders are known for removing or altering log files to hide their tracks. If important information is logged to a central host, loss of information less likely. Log files should be given consistent names and locations across all systems at a site to simply the task of locating similar types of logging information across multiple architectures. All system log file files should be owned by system accounts and should not provide any type of access to normal users. At a minimum, three weeks of logging information should be kept. If the disk or tape resources are available, then increase the amount of archived data. From experience, I find that two or three months of logging information is beneficial.

Efficient and good account administration is also a crucial part of system security. Poorly installed accounts, accounts with poor or no passwords, and dormant accounts are still the most popular means for intruders to gain access to systems. At the NAS facility we have a locally developed central account management system, referred to as LAMS (Local Account Management System). All account activity (e.g., creation, deletion and modification) is done from a central host by automated tools which are a part of LAMS. The use of LAMS ensures that every account installed has an eight character, machine generated password, a home directory with default permission of 700, and a set of standard default environment files, with safe values for all common environment variables (e.g., *path* and *umask*). LAMS also provides the capability to change a user's password. This feature of LAMS has been used during several security incidents at the NAS facility. For example, if we suspect a NAS account has been compromised, we attempt to contact the user. If the user cannot be reached, we will disable the account and wait for the user to contact us. Once we hear from the user, the account will be re-enabled and the password will be changed using LAMS. The user is then notified and requested to change the password again.

The use of a centralized account management tool greatly reduces the time required to perform account operations. With the use of LAMS, accounts at NAS can be created or disabled on multiple hosts within a few minutes. The time factor becomes an important issue when a large number of accounts are involved in a security incident. There have been several cases at NAS where an operation was needed to be performed on several hundred accounts due to a security incident. Without the use of a centralized systems, such as LAMS, such a task would take many hours.

Monitoring account activity is another aspect of account management. We have a program which runs on a weekly basis to produce a report of all users who have not accessed their account for over 90 days. Due to the security risk of dormant accounts, all dormant accounts are disabled, and after an additional 30 days, the account is removed. If during that additional 30 days, the user contacts the accounts staff, the account will be re-enabled. When the account is disabled, the login shell of the account is set to a special program called *noshell*. The *noshell* program, which was written locally, reports on any attempt to gain access to the account (e.g., *rlogin*, *telnet*, *ftp*, *rsh*, *rcp*). Depending on how the program is configured at installation, a message can be sent to a monitor person, or a message can be logged via the *syslog* command. The message states which account was accessed, the time of access, the remote host (if the information is available), and the remote user (if the information is available). See *Figure 2* for an example message from the *noshell* program. There have been several occasions at the NAS facility where the use of the *noshell* program has revealed a security problem.

```
From: root
To: crabb@nas.nasa.gov
Subject: WARNING - Login to disabled account
**************** SECURITY ALERT *********************
"UNKNOWN    REMOTE    USER"    has    attempted    to    log    into
"badboy.nas.nasa.gov"
using the login name of "fredie"
Time of login: Wed Sep 18 14:46:22 1991
The remote login originated from: foobar.nas.nasa.gov
Internet Address of remote host: 129.160.33.145
Terminal line user attached to: ttyp11.
Please investigate this incident as soon as possible.
```

Figure 2: Example message sent by *the noshell* program

The final aspect of account management at the NAS facility, is monitoring changes to the */etc/passw*d file on all NAS hosts. This function is performed using a locally written program called *getall*. The *getall* program runs from a central location on a nightly basis. It copies down the */etc/passwd* file from each NAS host and compares it with the password file from the previous run. The program reports on additions and deletions to the file, changes in a user's login shell, changes to a user's UID/GID, or changes in *root* passwords. The *getall* program also sends a separate report on any new UID 0 accounts added to a password file.

One of the basic functions of good systems security is file system administration. File system administration includes such tasks as: ensuring system files are owned by system accounts, ensuring system files and directories are not group or world writable, ensuring critical system files are properly read protected, and the monitoring of important system files and directories for changes. At the NAS facility, we use several of the freely available security tools to perform basic file system auditing. The *COPS* package, which is one the most popular security checking tools on the Internet, is used at the NAS facility for the majority of our general file system administration. The *COPS* suite of tools is run on a weekly basis on all NAS hosts and the output reports are sent to a central person. All suspicious reports are investigated as soon as possible. Most of the problems reported by the *COPS* programs tend to be things that the system support people did (e.g., add a file to a system directory).

Auditing and tracking of SUID/SGID files can be a major headache at a site with a large number of hosts of multiple architectures. The *suid.chk* program, which is part of COPS, can be used to audit and track SUID/SGID programs. The *suid.chk* program does an exhaustive search of the file system and creates a list of all SUID/SGID programs. The list is then compared to a master configuration file and any differences are reported. The *suid.chk* program was modified at NAS to drastically reduce the work required to maintain master SUID/SGID files on numerous hosts. Instead of maintaining a separate master file on each host, the *suid.chk* program was modified to allow the use of group master files. There are currently 36 SUID/SGID master configuration files at NAS. A program called *get_group* is run by *suid.chk* to determine the master group for the local host. The *get_group* program also has an option to list all of the hosts of a specific group. This feature is handy if a file mode or permission needs to be updated on a group of hosts. The *suid.chk* program was also modified so the output report would show which master file was used to generate the report.

```
              Suid Audit Report for foobar.nas.nasa.gov
              =======================================
The Baseline file is: /usr/local/utils/cops/suid.files.sgi.group11

These files are newly setuid/setgid:

-rwsr-xr-x 1 root root 278640 Jan 18 16:13 /usr/sbin/xwsh
-rwsr-xr-x 1 root root 278640 Mar 30 1993 /usr/sbin/xwsh.badsum

These files are no longer setuid/setgid:

-rwsr-xr-x 1 root root 278640 Mar 30 1993 /usr/sbin/xwsh
```

Figure 3: Example report from suid.chk program

Generating the master files and determining which hosts belonged in which group was a very arduous and time consuming tasks. However, the weekly processing of reports and maintenance of the master files can now be done in several hours, whereas before it took many hours. The master configuration files, which are kept under RCS control, are updated on a weekly basis as the reports are read. Most of the changes required to the master files involve changing the file date stamp from a time stamp to a year stamp. Occasionally, a new SUID program will be updated or installed on all hosts, which requires the modification of all master configuration files. The master configuration files, and the *get_group* and *suid.chk* programs are kept in the master source tree on a file server. Each week, the files are automatically updated on all hosts, via a cron job, prior to the run of the *suid.chk* program. See *Figure 3* for an example report from a run of *suid.chk*.

Another aspect of file system auditing involves the checking of permissions on users' home directories and their environment files (e.g., *.login*). One of the *COPS* routines, *home.chk* checks for permission and existence of users' home directories. However, in place of that program, we use *homecheck*. The *homecheck* program checks the permission and ownership modes of users' environment files as well as their home directories. The program also checks the permission and ownership modes of the parent directory of each home directory, and it can be configured to ignore certain accounts.

```
                  Rhosts File Audit Report For foobar
                 -------------------------------------
WARNING:  Possible  illegal  or  malformed  .rhosts  entries  for  user
johndoe:
 sunny.larc.nasa.gov majdi

WARNING:  Possible  illegal  or  malformed  .rhosts  entries  for  user
janedoe:
 uxh.cso.uiuc.edu aae391ac

WARNING:  Possible  illegal  or  malformed  .rhosts  entries  for  user
jnsmith:
 rtccd.arc.nasa.gov *
```

Figure 4: Example output from the *raudit* program

The *.rhosts* and *.netrc* user environment files are audited using separate tools due to the security risk these files can create. The *raudit* program is used to audit users' *.rhosts* files. The program is run on a weekly basis to provide a list of all *.rhosts* entries which appear to be illegal (i.e., the login name in the entry does
not match the account login name). The *raudit* program incorporates the use of an alternate login id database to reduce the report of false-positive illegal *.rhosts* entries. The *raudit* program is also used to audit all *.rhosts* files for the presence of specific hosts. This is done in cases where a security incident has been reported at a remote site where NAS users have accounts. For example, if I receive a report that systems at Berkeley were compromised and I know some NAS users have accounts on those systems, I will run the *raudit* program to look for *.rhosts* entries with *berkeley.edu* in the host field. The *raudit* program can also be used to produce a full report on *.rhosts* files with various statistics about them. For more information on auditing and managing *.rhosts* files, refer to the paper "Who's Trusting Whom? How to Audit and Manage Users' .rhosts Files" [1]. See *figure 4* for an example report from the *raudit* program.

The *netrc_chk* program, which is still under development, is used to report on users who have passwords in their *.netrc* files. Currently, the output from *.netrc* will show how many *.netrc* entries contain passwords for each user. The *netrc_chk* program ignores passwords that are of the form *username@host*. See *Figure* 5 for example output from the *netrc_chk* program. Both the *raudit* and *netrc_chk* programs were developed at NAS and are available on request.

```
                 Netrc File Audit Report for Foobar
           ----------------------------------------------
User johndoe has 3 passwords in his/her netrc file.
User janedoe has 1 passwords in his/her netrc file.
User jsmith has 5 passwords in his/her netrc file.
```

Figure 5: Example output from the *netrc_chk* program

Special access control and management on a large number of hosts can be a major problem, especially when a large number of people need the *root* password on multiple hosts. I define special access as the privilege to use one or more of the accounts necessary for support of the computer facility. The *root* account is one example. To reduce the difficulty associated with special access control in a large environment, there are several key tasks which should be performed. The first is the creation a database of users who have special access, why they need it and when their access expires. The database should also include all special access accounts and passwords. Systems of a like architecture (e.g., all SGI workstations) should share a common *root* password in order to reduce the need for several hundred different *root* passwords. A formal procedure for changing all special access passwords and distributing them should also be implemented. Special access passwords should be changed on a periodic basis (e.g., every three months or less).

At the NAS facility, we have 38 special access passwords (mostly *root* accounts) and over 100 people who require special access on one or more hosts. As a general rule, the special access passwords are changed every one to two months or whenever a person with special access leaves the project. The NAS special access password database is created and maintained on a Macintosh computer using the Hypercard application. Each user has a separate "card" in the database which lists the password groups the user has been approved to receive. Each time the passwords are changed, the user receives a new hardcopy password sheet. The password sheet contains the password for each group; however, the password printed on the sheet is not the actual password. The actual password is obtained by applying an algorithm to the password written on the sheet. The password algorithm is not written on the sheet and is the only item support personnel are required to remember. A full description of the NAS password management scheme can be found in the paper "Password Security In A Large Distributed Environment" [2].

Until recently, the special access passwords at the NAS facility were changed manually and the task usually took two people almost ten hours to complete. We began using LAMS to change passwords for all non-root special access accounts; however, it still took many hours to change the *root* password on over 300 systems. The task of changing *root* passwords was automated by the development of a small group of programs that use the client daemon from LAMS. The master driver program contains the hostname and password groups for all NAS hosts. The master program is used to create the lists of hosts for each password group, and the shell scripts which perform the actual password change. The password changing program works by modifying the password file to insert the new encrypted password string. The seven line program is shown in the *figure* 6. The use of this program requires the ability to write to the */etc/password* file and it requires every host to trust one host. LAMS was perfect for such a task, since it used a client-server daemon scheme to perform tasks, and trust was not an issue. The only manual task of a password change in the current scheme is to update the master program to include all new hosts, password classes, and the encrypted password strings. Now, all *root* passwords at the NAS facility can be changed in less than a hour!

```
#!/bin/sh
cp /etc/passwd /etc/passwd.bak

sed    's/^root:[^:]*/root:r9Xs5puJ9Obmo/g'    /etc/passwd.bak    >
/etc/passwd

if [ !-s /etc/passwd ]
then
 cp /etc/passwd.bak /etc/passwd
 echo "Password Update Failed"
fi
```

Figure 6: Password file update script

Password validation, which is the process of ensuring passwords are well constructed and cannot be easily guessed by a cracking program, is an important security task because poor passwords are still one of the most popular methods used to gain access to a system. There are two types of password validation, pro-active and re-active. Pro-active password validation is the process of ensuring a password meets specific construction rules at the time the password is set. Password construction rules include password length, and types of character. Re-active password validation is the process of verifying a password cannot be guessed by a password cracking program. Password cracking is less of a problem at sites that use a password shadowing mechanism. Many vendors now provide password shadowing as a standard part of the operating system; however, most flavors of UNIX still have weak password construction enforcement rules. Pro-active password validation requires the replacement of the */bin/passwd* program. There are three replacements available on the Internet: *password-plus*, *anl-password* and *npasswd*. All three of these programs operate in a similar manner. Each program has an external configuration file which specifies the password construction rules. Each program also allows the inclusion of a dictionary of words to check against when verifying a new password.

Re-active password validation is done using a password cracking program. In the last several years, there have been very large improvements in password cracking programs and several of them are freely available on the Internet. Several programs have even been written for the Connection Machine (CM-5). One of the more popular password cracking programs is *crack*. This program is available from CERT and several other sources. Matt Bishop also has a password cracking program which is available with his *Deszip* package. The Internet intruders have a very high-speed, widely used cracking program called *kc* (killer crack). They even have password cracking services available where you can email in a password file and they email back any passwords that were guessed by the program. Most of the password cracking services are operated by and used by the Internet intruders.

At the NAS facility, we are currently using both methods of password validation. We use *passwd-plus* as a */bin/passwd* replacement for all Sun and SGI systems. Currently, NAS passwords must be a minimum of six character long and must contain two of three groups of characters (letters, numbers and special characters). The */bin/passwd* replacement program is not installed on the Cray systems at this time because the Crays use password file shadowing, and the Cray version of */bin/passwd* has sufficient password construction enforcement rules. A recent policy was implemented at Ames Research Center regarding the construction of passwords. The new policy requires all passwords be eight characters in length and contain characters from three of the four groups (lower case, upper case, numbers, special characters). As a result of this new policy, we will be installing a new */bin/passwd* replacement program on all NAS hosts. Re-active password validation is done at the NAS facility using Matt Bishop's *Deszip* package. Re-active validation is usually only done when we suspect a NAS password file has been acquired by an intruder. The use of pro-active password validation has substantially improved the types of passwords used at the NAS facility. In the last three years, no passwords have been cracked during the re-active validation process.

There are a variety of other security tools freely available on the Internet aside from the tools mentioned here. There are several tools which can run for a single host to check the state of security on numerous hosts (e.g., a subnet of hosts). One of these tools is the *ISS* (Internet Security Scanner), written by Christopher Klaus, which can be used to scan a list of hosts for the presence of known security vulnerabilities or weaknesses in the system configuration (e.g., world exported file systems). An older program, called *sweep*, will scan a list of hosts and check for the vulnerabilities that were exploited by the Morris Worm. The *sweep* program was written at the Ballistic Research Lab in Maryland. The *SPI* (Security Profile Inspector), developed at Lawrence Livermore National Labs, performs similar functions to the *COPS* package. The list of security tools discussed here is by no means an exhaustive list of all the tools available.

Future Directions

There are still a number of improvements which can be made to the overall scheme of providing system security at the NAS facility. A number of projects are in the planning stage and several have been scheduled. One project involves the implementation of the *tripwire* program across all NAS hosts. The *tripwire* program is used to monitor a designated set of files and directories for any changes (e.g., unauthorized modification of files). Another project, which is currently in progress, is changing the systems default *umask* value. On most UNIX systems, the default *umask* set in the kernel is 0, which allows the creation of world-writable files. The user's *umask* value will be set at login time by reading a global environment file such as */etc/cshrc* or */etc/profile* or is set in the *login* or *cshrc* program. However, this does not effect files which are created by system processes. Our plan is to set the system-wide default *umask* value to 077 by modifying the CMASK value in the kernel. Another project under consideration is the redesign of our network connection filtering scheme. Currently we are using a host-based scheme, via the *tcpd* program. We are interested in using a combination of router-based and host-based filtering to increase the effectiveness of filtering network connections.

Conclusion

Monitoring and managing security in a large heterogenous environment is not a simple task. However, there are a large number of security programs available which can make the task manageable and efficient. There is also a wealth of information on the Internet on how to maintain and improve the security of your site. The information is available for those willing to ask and to take the time to acquire the information. In this paper I have presented an overview of how the security is maintained the NAS facility, which can be viewed as a typical large scale heterogenous environment. While the methods and tools used to monitor and manage security at NAS may not be appropriate for every facility, they are a good starting place. As time and patience allow, new tools and procedures will be implemented at the NAS facility to further improve the security of the facility and decrease the time required to perform the work.

Availability

All of the programs discuss in this paper that were developed at the NAS facility, are freely available from NAS. To request the source for any one of the programs or policies discussed, send an email request to *doc-center@nas.nasa.gov.* If you have any questions regarding the use of the security programs discussed in this paper or how security is handled at the NAS facility, send email to *crabb@nas.nasa.gov.*

Author Information

Michele Crabb has been the primary computer security analyst for the NAS Facility at NASA Ames Research Center for over four years. During her nine years at Ames, Michele has worked in several divisions, in a variety of positions ranging from applications programming to UNIX system support. Prior to becoming the NAS security analyst, she was actively involved in providing system administration support for the large number of workstations at the NAS facility. Michele can be reached via electronic mail at *crabb@nas.nasa.gov*, or via US Mail at NASA Ames Research Center, Mail Stop 258-6, Moffett Field, CA 94035-1000.

References

1. Michele D. Crabb, "Who's Trusting Whom? How To Audit and Manage Users' .rhosts Files," *Proceedings of the Third Annual System Administration, Networking and Security Conference*, The Open Systems Conference Board, Apr. 1994.
2. Michele D. Crabb, "Password Security in a Large Distributed Environment," *Proceedings of the Second Workshop on Unix Security*, pp. 17-29, The Usenix Association, Aug. 1990.

APPENDEX A

Acceptable Use Statement for NAS Systems Division Computing Resources

The following document outlines guidelines for use of the computing systems, resources and facilities located at and/or operated by the Numerical Aerodynamic Simulation (NAS) Systems Division at NASA Ames Research Center. The purpose of these guidelines is increase awareness of computer security issues and to ensure that all NAS users (scientific users, support personnel and management) use the NAS Systems Division computing systems, resources and facilities in a efficient, ethical and lawful manner.

NAS accounts are to be used only for the purpose for which they are authorized and are not to be used for non-NAS related activities. Unauthorized use of a NAS account/system is in violation of Section 799, Title 18, U.S. Code, and constitutes theft and is punishable by law. Therefore, unauthorized use of NAS Systems Division computing system, resources s and facilities may constitute grounds for either civil or criminal prosecution.

In the text below, “users” refers to users of the NAS Systems Division computing systems, resources and facilities.

1. Users are responsible for using the NAS computing systems, resources and facilities in an efficient and effective manner.

2. The NAS Systems Division computing systems are unclassified systems. Therefore, classified information may not be processed, entered or stored on a NAS Systems Division computing system. Information is considered “classified” if it is Top Secret, Secret and/or Confidential information which requires safeguarding in the interest of National Security.

3. Users are responsible for protecting any information used and/or stored on/in their NAS accounts. Consult the NAS User Guide for guidelines on protecting your account and information using the standard system protection mechanisms.

4. Users are requested to report any weaknesses in NAS computer security, any incidents of possible misuse or violation of this agreement to the proper authorities by contacting NAS User Services or by sending electronic mail to *security@nas.nasa.gov.*

5. Users shall not attempt to access any data or programs contained on NAS systems for which they do not have authorization or explicit consent of the owner of the data/program, the NAS Division Chief or the NAS Data Processing Installation Computer Security Officer (DPI-CSO).

6. Users shall not divulge access information (e.g., Dialup or Dialback modem phone numbers, or lists of user accounts).

7. Users shall not share their NAS account(s) with anyone. This includes sharing the password to the account, providing access via an .rhost entry or other means of sharing.

8. Users shall not make unauthorized copies of copyrighted software, except as permitted by law or by the owner of the copyright.

9. Users shall not make copies of system configuration files (e.g. */etc/passwd*) for unauthorized personal use or to provide to other people/users for unauthorized uses.

10. Users shall not purposely engage in activities to: harass other users; degrade the performance of systems; deprive an authorized NAS user access to a NAS resource; obtain extra resources, beyond those allocated; circumvent NAS computer security measures or gain access to a NAS system for which proper authorization has not been given.

11. Electronic communication facilities (such as Email or Netnews) are for authorized government use only. Fraudulent, harassing or obscene messages and/or materials shall not be sent from, to or stored on NAS systems.

12. Users shall not down-load, install or run security programs or utilities which reveal weaknesses in the security of a system. For example, NAS users shall not run password cracking programs on NAS Systems Division computing systems.

Any noncompliance with these requirements will constitute a security violation and will be reported to the management of the NAS user and the NAS DPI-CSO and will result in short-term or permanent loss of access to NAS Systems Division computing systems. Serious violations may result in civil or criminal prosecution.

I have read and understand the Acceptable Use Statement for NAS Systems Division Computing Resources for use of the NAS computing facility and agree to abide by it.

Requestor's Signature: **Date:**

RNS TECHNICAL REPORT

Title: Guarding The Fortress

Efficient Methods to Monitor Security on 300 Systems

Author(s): Michele D. Crabb

Clearance:

Form 427 has been filed with the division secretary. This report is unclassified. _______ Author's initials.

Reviewers:

"I have carefully and thoroughly reviewed this technical report. I have worked with the author(s) to ensure clarity of presentation and technical accuracy. I take personal responsibility for the quality of this document."

Signed: ________________________________

Name: ________________________________

Signed: ________________________________

Name: ________________________________

Branch Chief:

Approved: ________________________________

Date & TR Number: