Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

Introduction and background

This paper is written in an attempt to be both useful and educational.
We chose to examine a real life server that provides an array of
services to the members of a large department. Since this is a real
system used by real people we decided to obscure some information in
the interest of privacy. However, the members of the department
computer support group will have access to the complete text of this
report to facilitate corrective actions suggested here.

We will refer to the examined server as 'the local host', 'the host',
or 'the system'. The host is a part of the departmental network that
constitutes a subnet of the university wide network. This subnet will
be referred to as 'the local subnet', 'the subnet', or
'the departmental subnet'. There are 242 active user accounts on
the host. These users will be called 'the local users' or just
'the users'.

The audit of the host is performed and the present report is written
by a single person who is a member of the college computer support
team. Henceforth, this single person refers to himself as 'we'.

The host is a dual 300 MHz processor Alpha server (2100 series) with
1GB of RAM. The storage is provided by a hardware RAID array with 2GB
of system partitions (/ and /usr/), 7GB of local applications
partitions (/usr/local/ and /x/), and 11.5GB users home directories
partition (/usr/users/). The operating system is Tru64 version 4.0F.
A patchkit for the OS was last time applied in June 2000 (a newer
patchkit is freely available from the vendor now!). Both the hardware
and the software of the host are provided by the single vendor,
Compaq.

Major functions performed by the system are:
    file storage and server (NIS and NFS server),
    web server (apache),
    mail server (sendmail, imap, pop servers),
    shell, applications server (text processing, scientific
            computing and visualization tools),
    ftp server (non-anonymous) and other inetd services,
    boot and X window server for several X terminals,
    print server (lpd).

The users of the system are faculty, graduate and undergraduate
students, visitors and staff, total of over 240 people. A typical user
has a home directory, shell access, login from several X terminals,
a web page directory, and a mailbox (e-mail). Most users casually make
telnet, ftp, unencrypted pop and imap connections to and from
the system.

The host is located in a keyed entry room (department's computer lab)
where only faculty, graduate students, and stuff can enter. The floppy
drive and the cdrom drive of the system are exposed, and the console
is available. Unfortunately, relocating to a room where only system
administrators can enter is not possible due to departmental space
restrictions. Physical access to the host means that an attacker
(given (s)he was able to enter the computer room with or without the
key) can disrupt the service by unplugging the power cord or the
network cable or by inflicting a physical harm on the computer. A more
sophisticated attacker could get a hold of Tru64 bootable CD and
reboot the computer with the CD in the CD-ROM thus effetively
bypassing all password protection and gaining root access immediately.
Moreover, if such an unscheduled reboot is not properly investigated
and the system integrity check is not performed, the successful
attacker may install a backdoor in the system that will allow him/her
unauthorized access in the future.

The contents of this report is as follows. In the next section we
present the summary of findings. Then the detailed descriptions of the
network services, filesystem configuration, and other security
relevant settings on the local host follow. Finally, steps for
improving setup correctness and security of the system are given.

Summary of the audit

The audited system provides several services to the members of the
department. It acts as a web server, e-mail server, file server, and
runs a number of other networks services. We examined network
services, filesystem settings, and local applications settings. Both
our findings and recommendations for improving security of the system
are presented in this paper.

We found that several important services (web, ssh, ssl, tcp wrappers)
are provided with obsolete software. Configuration files for many
services are either inconsistent or wrong. Updating the software will

also require reviewing the configuration files.

Software packages provided by the vendor are used for local services such as NFS, NIS, tftp, inetd, X window. These services are running with default access control settings which are too permissive. In most cases, editing appropriate access restrictions in the configuration files will decrease the risk of remote compromise of the system.

The filesystem was reviewed for suid and sgid files, device files, and applications special files. Access permissions for some of these files can be reduced in order to protect the system from intentional or unintentional damage.

A step by step program for improving security of the system is presented in the final section of this report.

Root account

Summary

The root user on a Unix system has complete control over the system. Unlike regular users who can usually only control their personal files, the root user controls the operating system itself and all programs running on the system. Therefore, the power of root account needs to be protected from unauthorized access. Our recommendations are as follows. First, a great care must be taken to pick a non-trivial root password. Second, we recommend restricting access to the root login prompt, only the authorized users of the system should be able to attempt root logins. Therefore, direct root logins need to be disallowed and the su program should be used to facilitate controlled and auditable root access to the authorized users.

Details

Root level access to the host is protected by a password. We want to emphasize that even though a standard Unix (DES) cryptographic hash of the actual password is stored in /etc/passwd file, the /etc/passwd file is readable by all users. This means that a user can run any of the freely available "password crackers" (such as Crack, of John the Ripper) to launch a dictionary attack (or a brute force attack) on the root password. In view of this, we recommend choosing the root password very carefully. It must be 8 characters long (the maximum

allowable by the OS), and it must not be a dictionary word, a proper name or a simple variation thereof.

Currently, the secure shell daemon allows direct root logins from the network. We recommend disabling this by including the DenyUsers directive in the sshd configuration file (in our case, /etc/sshd_config). This may prevent root password guessing attempts by remote attackers. It will also slow down a remote attacker who accidentally learned the root password by other means because (s)he will have to obtain an authorized regular user access before using the root password (one more account to break). Even though the telnet daemon is currently running, it does not allow direct root logins, we checked. On the system, the file /etc/securettys specifies the tty lines that are considered secure and permit logins as root. Currently, this files contains the following lines:

/dev/console
local:0
:0

The first line allows access form the system console (the text only terminal in the computer room with the hardwired serial connection to the system). The second and the third line allow root logins from local and remote X displays. We recommend restricting root logins only to the system console and disabling X root logins by removing the two corresponding lines. Since X connections are unencrypted, a cleartext root password is transmitted on the wire during the authentication phase and it can be vulnerable to eavesdropping. There is no local graphical display, therefore the line that enables it in the /etc/securettys is extraneous anyway.

The su program can be used by users to access root account. The Tru64 implementation of su requires the user to belong to the group number 0 (system) in order to issue su to become root. Therefore, only the authorized users who know the root password must consitute the system group. Currently, the former administrator who left the site almost a year ago is still in the system group. His user name must be removed from the system group. The su program logs all instances of root login attempts to the log file /var/adm/syslog.dated/current/auth.log. This provides capabilites to audit root access on the system especially if the system logs are also sent to a secure remote loghost.

There are 5 people with the knowledge of the current root password. Two of them are system administrators for the university and the other three are faculty at the department. One of the system

administrators (the author of the present report) maintains the OS
and the software products. One of the faculty (who actually has very
little Unix experience) is in charge of adding users to the system.
The other two faculty used to be involved with the system previously,
but recently they no longer worked on the system as administrators.
However, all changes in the system configuration usually needs to be
approved by this "committee". This non-authoritative administration
style suits the academic environment and fits well the historic spirit
of collaborative and friendly exchange of ideas between scholars and
researchers. As a result, there is no approved by the department
policy on creating, maintaining, and retiring user accounts on the
system. These tasks currently are performed by the least skilled
person with root level access using a couple of shell scripts. User
accounts are also created for groups (undergraduate seminar members,
conference participants, research groups). This practice reduces
accountability of individual users and promotes password sharing.
User accounts of people who left the department seem to remain on the
system forever. We know of at least one person who left the department
in 1992 (!) and still has an account on the system even though it has
not been used in years. There is in fact a number of user accounts
that have not been used in 2-3 years. Abandoned accounts like those
can be a convenient target for an attacker because there is a very
little chance that the actual owner of the account will notice a
successful break-in. We recommend developing a policy for user
accounts on the system that will include expiration of user accounts
for people who have left the department and are no longer accountable
to the department. We also recommend developing a set of written
procedures to handle creation, maintenance, and removal of user
accounts and delegating these tasks to a system administrator.

System startup

At the startup a number of services begin to run on the system. These
services are started at boot by links in /sbin/rc3.d to scripts in
/sbin/init.d.

We will identify and comment on these scripts one by one. A more
detailed analysis of activated services is in the following
sections of this report. Note that even though all of the following
scripts are called with 'start' argument at boot, various variable
settings in the system config file /etc/rc.config and in the scripts
themselves prevent the actual services from starting.

S00inet@ -> ../init.d/inet*

This script configures network interfaces and mounts nfs systems if
necessary. The local host has only one interface with a statically
assigned IP address and only local filesystems are used.

        S01quota@ -> ../init.d/quota*

Quotas are not used on the system. In the beginning of the script
/sbin/init.d/quota the variable QUOTA_CONFIG is defined as 'no', and
therefore no quota related actions are performed because
the subsequent standard 'start' and 'stop' actions are only performed
if the variable QUOTA_CONFIG is set to 'yes'.

        S04uucp@ -> ../init.d/uucp*

Uucp is not used on the system. This script only cleans the uucp
directory /var/spool/locks. This directory is owned by user and group
'uucp' and has permissions 775. Since it not writeable by other users,
a /tmp style attack where a user can create a link in /tmp to an
arbitrary file in the system and then wait for the system to clean its
/tmp directory by just 'rm -rf /tmp/*' (without checking for actual
files) will not work here.

        S08startlmf@ -> ../init.d/startlmf*

This script runs a single instance of the license managing facilty
(LMF) program once at each system start up. The LMF program copies the
license details for all enabled products from the license database to
the kernel cache. After that the program is terminated.

        S09syslog@ -> ../init.d/syslog*

This script starts the syslog daemon process syslogd. Syslogd does not
accept messages from remote hosts because the authorization file
/etc/syslog.auth is missing.

        S10binlog@ -> ../init.d/binlog*

This is the Tru64 proprietary daemon (binlogd) for logging kernel
messages.

        S11gateway@ -> ../init.d/gateway*

If a system is a gateway, this script starts the gated routing daemon.
This functionality is disabled for the local host in the file

/etc/rc.config by setting the variable GATED to 'no'.

        S12route@ -> ../init.d/route*

Static network routes are configured here. The routing daemon, routed, is disabled by setting the ROUTED variable to 'no' in the config file /etc/rc.config.

        S13rwho@ -> ../init.d/rwho*

Rwho service is disabled on the host by setting the variable RWHOD to 'no' in the config file /etc/rc.config.

        S14settime@ -> ../init.d/settime*

Sets the timezone, the current time and date. Three organizational ntp servers are polled at this time.

        S15named@ -> ../init.d/named*

The named server is disabled on the host by setting the variable BIND_SERVERTYPE to 'CLIENT'.

        S18nis@ -> ../init.d/nis*

The host is the NIS master for the departmental network. Portmapper, ypbind, ypserv, ypbind, and yppasswd are started. The NIS framework is used to maintain consistent passwords across departmental Tru64, FreeBSD, and OpenBSD workstations.

        S19nfs@ -> ../init.d/nfs*

NFS services (mountd, nfsd, statd, lockd) are started here.

        S20nfsmount@ -> ../init.d/nfsmount*

Automount daemon process is disabled on the system by setting the variable AUTOMOUNT to '0' in the beginning of this script.

        S21audit@ -> ../init.d/audit*

The audit daemon, auditd, is not used on the system. The script checks if the variable AUDITD_FLAG is empty in the beginning of this script. If it is empty, the script exits. The variable AUDITD_FLAG can be set at boot in the config file /etc/rc.config. It is not set on our system and the audit daemon is not started.

```
    S25preserve@ -> ../init.d/preserve*
```

This script preserves editor files in /tmp and /var/tmp.

```
    S26security@ -> ../init.d/security*
    S27sia@ -> ../init.d/sia*
```

These scripts set up the proprietary Compaq security integration
architecture (SIA). Even though this may be useful for a site with
mandatory access control and system accounting requirements, this
framework is not practical for the departmental computer systems
setup. Therefore this service need not be activated on the host.
These services are disabled by setting the variable SECURITY to
'BASE' in the config file /etc/rc.config.

```
    S30rmtmpfiles@ -> ../init.d/rmtmpfiles*
```

This script cleans temporary files from /tmp and /var/tmp. Note that
this script actually will only remove actual files and directories
(and no symbolic links!) because both /tmp and /var/tmp are writable
by arbitrary users who could place links to important files in these
directories (see our comments for uucp above).

```
    S36presto@ -> ../init.d/presto*
```

Prestoserve is a performance enhancing file system accelerator. It is
not used on the host. It is disabled by setting the variable
PRESTO_ENABLE to '0' ("no") in the beginning of this script.

```
    S40sendmail@ -> ../init.d/sendmail*
```

This script starts the sendmail daemon.

```
    S45xntpd@ -> ../init.d/xntpd*
```

This script starts the network time protocol daemon.

```
    S46timed@ -> ../init.d/timed*
```

Another version of the network time protocol daemon. This one is not
used on the local host. The timed daemon is only started if the
variable timed_conf is set to 'YES' in the config file /etc/rc.config
which is not the case on our system.

```
    S55inetd@ -> ../init.d/inetd*
```

The internet superserver daemon. Provides a number of services such as
telnet, ftp, pop3, imap, ntalk, finger, and others.

```
    S56dhcp@ -> ../init.d/dhcp*
```

The dhcp server (joind) is disabled by setting the variable JOIND to
an empty string in the system config file /etc/rc.config.

```
    S57cron@ -> ../init.d/cron*
```

Starts the cron daemon.

```
    S58lat@ -> ../init.d/lat*
```

This is not used (local area transport). It is disabled by setting
the variable LAT_SETUP to '0' in the beginning of the script.

```
    S59lsm@ -> ../init.d/lsm*
```

The logical storage manager is not used on the host because
the necessary startup file /etc/vol/volboot is missing from
the system.

```
    S60motd@ -> ../init.d/motd*
```

This script updates the /etc/motd file.

```
    S63write@ -> ../init.d/write*
```

This service is disabled (it would allow users receive messages from
remote hosts) by setting the variable WRITESERV to '0' in
the biginning of the script.

```
    S64apx@ -> ../init.d/apx*
```

This script starts proprietary advanced printing software (APS). APS
is not used on this site. The data directory /var/pd/odb is empty and
the variable APX_USE_IBGW is not set to 'TRUE' in the system config
file /etc/rc.config therefore APS is not started.

```
    S65lpd@ -> ../init.d/lpd*
```

Berkeley printing daemon.

```
    S66sshd@ -> ../init.d/sshd*
```

Secure shell daemon.

    S75acct@ -> ../init.d/acct*

System accounting is disabled because the variable ACCOUNTING is not
set to 'YES' in the system config file /etc/rc.config.

    S80crashdc@ -> ../init.d/crashdc*

If a coredump is present at boot, this script collects the information
about it in /var/adm/crash.

    S85vectored_x@ -> ../init.d/vectored_x*

If the host had an accelerated graphics device this script would
choose the appropriate X server start up configuration (instead
the system has a 'GENERIC' graphics device and it actually has
no graphical display attached).

    S88httpd@ -> ../init.d/httpd*

Apache web server.

    S90ws@ -> ../init.d/ws*

Notifies the configuration manager that multiuser has been reached.

    S94xfontserver@ -> ../init.d/xfontserver*

X window font server.

    S95xlogin@ -> ../init.d/xlogin*

Starts X display manager (dtlogin or xdm).

    S99appletalk@ -> ../init.d/appletalk*

Starts Appletalk services to enable printing to the department's only
Apple laser printer.

Auditing Apache Web Server

Summary

The audit revealed the following problems with the web server.

   - The Apache web server version currently running is obsolete.
     Upgrading to the latest stable release update (1.3.20 as of
     June 2001) is recommended.
   - There are inconsistencies in configuration files. Access
     control options are applied to non-existent directories and
     actual directories are not protected as a result. In
     addition, CGI scripts configuration options are in mutual
     contradiction. Therefore, configuration options should be
     corrected as well.
   - A number of files and directories have incorrect and
     possibly unsafe access permissions that can may result in
     web server subversion by local or remote users.

Several steps for improving the web server setup are proposed.

Details

The version 1.2.4 of the Apache web server is installed. It serves
faculty, grads, undergrads, and staff personal web pages as well as
the departmental page, classes pages for the current and several past
semesters. Both static and dynamic (CGI) content is present.
The departmental page is maintained by a person with the root level
access, however most of the work is done under an unprivileged user id
'wwwadm'. The web server is started at boot by the script in
/sbin/init.d.

   Web server configuration

The apache web server configuration files are httpd.conf, srm.conf,
and access.conf in /usr/local/etc/httpd/conf/. Since the installation
took place, several original (default) options changed. The two most
important ones are listed below:

srm.conf: DocumentRoot /usr/users/wwwadm
srm.conf: UserDir WWW

The first option sets the HTTP document root tree at /usr/users/wwwadm
and the second option allows accessing users personal web pages by

appending ~username to the host's name.

Other important options that have default values are:

srm.conf: AccessFileName .htaccess
srm.conf: ScriptAlias /cgi-bin/ /usr/local/etc/httpd/cgi-bin/
srm.conf: AddHandler cgi-script .cgi
srm.conf: AddType text/html .shtml
srm.conf: AddHandler server-parsed .shtml
httpd.conf: ServerRoot /usr/local/etc/http

Note that there are two options, ScriptAlias and AddHandler, that
enable execution of CGI scripts. The ScriptAlias option is usually
used to confine all CGI scripts on the server to a single directory
under administrative control. However, the AddHandler option tells the
web server to treat any executable file with the name ending in .cgi
as a CGI script. Therefore, regular users can create CGI scripts. If
not written carefully, these programs can provide remote attackers
with the user level access to the local system. Even though the CGI
scripts run currently with the user id of 'httpd', this presents
a security risk as sensitive information may be exposed or corrupted.

If CGI scripts are necessary for this site, it is recommended that
most complex and/or critical scripts be collected in centralized
depository by the sysadmin or the web server admin and scrutinized for
potential security holes. Furthermore, the Apache web server
distribution comes with the suEXEC module that allows execution of
users' CGI scripts with the actual user id (as opposed to the id of
'httpd'.) If used properly, this option may be both more secure and
more convenient for the regular users in the present setup.

Several users web pages make use of server side includes (SSI). This
mechanism potentially allows for executing arbitrary commands on the
local host by the web server and sending the output to remote users.
If this functionality is needed, it is recommended that the
IncludesNOEXEC option be added to the global configuration settings.
This option will disallow #exec and #include commands, but all other
SSI functionality will be preserved. Note that a small local usage
guide states that SSI are disabled whereas actually this feature is
enabled in configuration files.

The following configuration options in the file access.conf refer to
non-existing directories:

<Directory /usr/local/etc/httpd/htdocs>
<...skipped...>

```
<Directory /s/local/etc/httpd/cgi-bin>
<...skipped...>
```

(These are former DocumentRoot and ScriptAlias directories.) As a
result, the intended access permissions for the current directories
are not set. We recommend to create access permissions configuration
options for the current DocumentRoot and ScriptAlias directories. All
access should be denied by default and granted only on the basis of
need. Also, is it recommended that users be not allowed to change
these security settings. One way to achieve this is to put the
following configuration options in the httpd.conf file.

```
<Directory />
    AllowOverride None
    Options None
    Order deny,allow
    Deny from all
</Directory>
```

Then more relaxed settings may be enabled by the administrator for
subdirectories as needed.

Logging is adequate in the current setup. Both access_log
and error_log files are rotated and compressed monthly and stored
for 6 months.

     Filesystem permissions

Several files and directories in the DocumentRoot hierarchy have
unsafe access permissions. The list of group and/or world writable
files and directories is given below (user names are obscured by
asterisks).

```
host:~ # find /usr/users/wwwadm \( -perm -0006 -o -perm -0060 \)
-print 2>/dev/null
/usr/users/wwwadm/.dt/startlog.older
/usr/users/wwwadm/.dt/startlog.old
/usr/users/wwwadm/.dt/startlog
/usr/users/wwwadm/gradprogram
/usr/users/wwwadm/********
/usr/users/wwwadm/********/bin
/usr/users/wwwadm/********/tmp
/usr/users/wwwadm/********/tmp/mathsafe.log1
/usr/users/wwwadm/********/packages
/usr/users/wwwadm/********/******
```

```
/usr/users/wwwadm/********/*****
/usr/users/wwwadm/********/********
/usr/users/wwwadm/********/********
/usr/users/wwwadm/********/******
/usr/users/wwwadm/********/*****
/usr/users/wwwadm/********/index.html
/usr/users/wwwadm/********/mathserv-demos.html
/usr/users/wwwadm/********/mathserv-future.html
/usr/users/wwwadm/********/mathserv-instructions.html
/usr/users/wwwadm/********/mathserv-resources.html
/usr/users/wwwadm/********/********test
/usr/users/wwwadm/.netscape/plugin-list.BAK
/usr/users/wwwadm/.netscape/plugin-list
/usr/users/wwwadm/.netscape/preferences.js
/usr/users/wwwadm/.netscape/registry
/usr/users/wwwadm/.netscape/xover-cache/host-/hostinfo.dat
/usr/users/wwwadm/.netscape/history.list
/usr/users/wwwadm/.netscape/liprefs.js
/usr/users/wwwadm/preprints/preprints.html
/usr/users/wwwadm/events/archive/1998/approx.html
/usr/users/wwwadm/publications/images
/usr/users/wwwadm/publications/index.html
/usr/users/wwwadm/publications/pubs-authors.pl
/usr/users/wwwadm/publications/preprints.html
/usr/users/wwwadm/cgi-bin/test.pl
/usr/users/wwwadm/icons
```

It recommended that group and world write access be disabled for these
files.

The ScriptAlias directory (/usr/local/etc/httpd/cgi-bin) is owned by
user 'wwwadm' and group 'system', and has permissions 777. Therefore
any user can create/delete files from this directory. If these CGI
scripts are necessary for the local site, their permissions and the
ScriptAlias directory permissions should be changed to 755.

Most CGI scripts found on the local site were downloaded from the
Internet as far as 4 years ago and have not changed since. At least
one of the CGI scripts (the infamous FormMail.pl) was recently abused
by a remote spammer. This script was since disabled. However, complete
audit of the remaining CGI scripts is suggested.

        E-mail services

Summary

The sendmail version 8.9.3 is currently running on the host. This is a stable version. A number of inconsistencies is present in sendmail configuration files and several users mailboxes have incorrect access permissions.

Details

The host is acting as an e-mail hub for the department. It receives, stores, and sends e-mail for the local users. The sendmail program (version 8.9.3) operates SMTP services and two additional daemons (ipop3d and imapd) provide e-mail access and retrieval to the users.

Hub for the local net

The host is a mailhub for the other computers on the local net. Those machines are configured as e-mail null clients. All mail originating from the null clients is delivered to the host and then sent to the proper destination. No other machine on the local network which is accessible to the local users is able to receive e-mail.

Configuration

The incoming mail spool directory (local users mailboxes) is /var/spool/mail. The outgoing mail spool directory is /var/spool/mqueue. Sendmail is started at boot time by a script from /sbin/init.d. The (default) configuration file is /etc/sendmail.cf (linked to /etc/mail/sendmail.cf.)

The configuration files reside in /etc/mail. The main sendmail configuration file sendmail.cf was generated from an m4 macros file with the following features (irrelevant version control info was removed).

```
OSTYPE(osf1)dnl
DOMAIN(VU_CAS)dnl
FEATURE(masquerade_envelope)dnl
MASQUERADE_AS(math.Vanderbilt.Edu)dnl
MAILER(local)dnl
```

```
MAILER(smtp)dnl
FEATURE(`access_db',`dbm /etc/mail/access')
FEATURE(`blacklist_recipients')
```

According to author's (Eric Allman) notes the MAILER macros need to be
moved to the end of the configuration file.

The OSTYPE line pulls in the following config options (from the OSF/1
specific file in the cf/ostype directory).

```
define(`ALIAS_FILE', ifdef(`_USE_ETC_MAIL_', `/etc/mail/aliases',
    `/usr/adm/sendmail/aliases'))dnl
ifdef(`STATUS_FILE',, `define(`STATUS_FILE', ifdef(`_USE_ETC_MAIL_',
    `/etc/mail/statistics', `/usr/adm/sendmail/sendmail.st'))')dnl
ifdef(`HELP_FILE',, `define(`HELP_FILE', ifdef(`_USE_ETC_MAIL_',
    `/etc/mail/helpfile', `/usr/share/lib/sendmail.hf'))')dnl
define(`confDEF_USER_ID', `daemon')
```

The DOMAIN line includes the following file (again, irrelevant options
are skipped.) Actual hostnames are obscured by asterisks.

```
FEATURE(redirect)dnl
FEATURE(use_cw_file)dnl
FEATURE(always_add_domain)dnl
define(`confCW_FILE', `/etc/mail/sendmail.cw')dnl
define(`BITNET_RELAY', `******.**********.***')dnl
define(`confLOG_LEVEL', `12')dnl
define(`confPRIVACY_FLAGS',`goaway,restrictmailq,restrictqrun')dnl
define(`confFORWARD_PATH', `$z/.forward.$w:$z/.forward')dnl
```

The 'redirect' feature allows aliasing e-mail addresses of people who
have left the department to the <their new e-mail address>.REDIRECT,
and then the e-mail will bounce with the error giving the new e-mail
address. This may be a useful feature, but the current alias file on
the host does not make use of it.

The 'use_cw_file' feature provides for alternate names for the host.
The subsequent define tells to look at the file /etc/mail/sendmail.cw
for these names. This file contains several hostnames that no longer
correspond to actual hosts. The sendmail.cw file needs to be updated.

The 'always_add_domain' feature makes sendmail to include the local
host domain even on locally delivered mail. This option may be useful
when some users have different login names on several machines that
share an e-mail hub. The mail storage on the local site is shared
among four systems, however they all use NIS and therefore this

feature is not needed.

The BITNET_RELAY macro is obsolete for the site and needs to be
removed.

The 'confLOG_LEVEL' macro sets the logging level to 12 that is all
debugging information is logged. Unless the site is experiencing
problems with e-mail it is recommended to lower the debugging level to
a lower value of 9 (the information needed to trace messages will
still be logged).

The 'confPRIVACY_FLAGS' macro allows to restrict the amount of
information about users and the sendmail configuration that is
exposed.
The option 'goaway' disables SMTP status queries, 'restrictmailq' will
only allow mqueue directory's group to see the queue, and
'restrictqrun' allows only mqueue directory's group to run the queue.

The 'confFORWARD_PATH' macro allows users to forward all mail from
their local accounts to other e-mail addresses of their choice using
the standard .forward files.

The 'masquerade_envelope' and the MASQUERADE_AS macro specify a single
domain name that all machines on the local network hide behind when
they send mail through our host. We recommend adding the
'allmasquerade' feature for consistency in rewriting envelopes and
headers.

MAILER options for 'smtp' and 'local' allow for e-mail deliveries on
local machine as well as to and from remote mail servers.

The FEATURE(`access_db',`dbm /etc/mail/access') line turns on the
access database feature. This is a very convenient way to manage
access to sendmail from a single source file.

The FEATURE(`blacklist_recipients') line turns on the ability to block
incoming mail for certain recipient usernames, hostnames, addresses in
the access database. We recommend using this feature to block mail
from known spammers.

The site also maintains the /etc/mail/relay-domains file. Its contents
are identical to the RELAY hosts from the access database. We
recommend removing this file and using the access database to
centralize all access control configuration in one place.

Filesystem permissions

The mail spool directory is /var/spool/mail.

```
atlas $ ls -ld /var/spool/mail
drwxrwxrwt   3 root      system        8192 Jun 13 15:51 /var/spool/mail
```

The sticky bit is on. This prevents users from being able to delete
other users mailboxes. Access permissions on most users mailboxes are
set to the correct value, however there are a few exceptions (actual
user names are obscured by asterisks):

```
-rwxrwxrwx   1 ******    mail        177020 Jun 13 15:11 ******
-rw-rw----   1 ***       bin              0 Jan 20 15:22 root
-rw-r--r--   1 ***       users      4829378 Aug  4  2000 ********
```

Root's mailbox needs to be protected better!

The access permissions of aliases file and its derivatives are
acceptable, but for uniformity, we recommend to change them all to 755
and ownership to root:system.

```
-rw-r--r--   1 root      system        2080 Feb  6 09:27 aliases
-rw-r--r--   1 root      system       32768 Jun 18 15:21 aliases.db
-rw-r--r--   1 bin       bin              0 Jun 17  1998 aliases.dir
-rwxr-xr-x   1 bin       bin            876 Nov 15  1996 aliases.dist*
-rw-r--r--   1 root      system        1530 Sep 26  1997 aliases.old
-rw-r--r--   1 bin       bin           1024 Jun 17  1998 aliases.pag
```

The contents of the /etc/mail domain also have acceptable access
permissions.

```
-rw-r--r--   1 root      system          49 Jun 18 15:20 access
-rw-r--r--   1 root      system        4096 Jun 18 15:20 access.dir
-rw-r--r--   1 root      system        1024 Jun 18 15:20 access.pag
-rw-r--r--   1 root      system          11 Apr 16 09:59 relay-domains
-rw-r--r--   1 root      system       32706 Apr 16 09:33 sendmail.cf
-r--r--r--   1 root      system       32706 Apr 16 09:33 sendmail.cf.NEW
-rw-r--r--   1 root      system         205 Jul 30  1998 sendmail.cw
```

Imapd and ipop2d/ipop3d

The mail retrieving services are provided my the imapd, ipop3d, and
ipop2d daemons that are part of the Pine e-mail client distribution
from the Washington University. Many local users use pine for handling

their mail. The pine version 4.30 is installed. The latest released version of pine is 4.33, it is a bug-release, we recommend upgrading.

Presently, remote users connect to the host from arbitrary remote sites to retrieve their e-mail messages. These connections (POP and IMAP) are not encrypted. Both usernames/passwords and e-mail messages are transmitted in the cleartext. We recommend using SSL for securing remote connections. The local site is using the Washington University The newer version of imapd is now available which includes support for SSL. We recommend upgrading. The POP service is provided by ipop2d and ipop3d daemons from the same distribution. Both IMAP and POP services can be secured by SSL.

Internet services

Summary

The internet superserver inetd is activated at system startup. Inetd monitors a number of well known ports for connections and starts the corresponding service daemon once a connection is requested. The configuration file /etc/inetd.conf was examined.
Recommendations:
    - restrict access to the tftp service using TCP Wrappers,
    - disable or restrict access to finger, comsat, ntalk, dtspc,
            and rpc.cmsd services,
    - consider secure (using encryption) replacements for telnet,
            ftp, imap, and pop3d services.

Details

The internet superserver inetd is controlling access to internet services. The following services are enabled (the relevant lines from the /etc/inetd.conf file are given):

```
ftp     stream  tcp  nowait  root  /usr/local/sbin/tcpd      ftpd -l
telnet  stream  tcp  nowait  root  /usr/local/sbin/tcpd      telnetd
pop3    stream  tcp  nowait  root  /usr/local/sbin/tcpd
   /usr/local/sbin/ipop3d
imap    stream  tcp  nowait  root  /usr/local/sbin/tcpd
   /usr/local/sbin/imapd
```

```
finger stream  tcp  nowait  root  /usr/sbin/fingerd     fingerd
tftp   dgram   udp  wait    root  /usr/sbin/tftpd
   tftpd -r /usr/local/tftp /usr/local/tftp /tmp
comsat dgram   udp  wait    root  /usr/sbin/comsat    comsat
ntalk  dgram   udp  wait    root  /usr/sbin/ntalkd    ntalkd
dtspc  stream  tcp  nowait  root  /usr/dt/bin/dtspcd    dtspcd
rpc.cmsd/2-4 dgram rpc/udp wait root /usr/dt/bin/rpc.cmsd rpc.cmsd
```

   Telnet and ftp

People with local user accounts on the host need ftp and telnet
access. These services operate in plaintext and can be vulnerable to
snooping usernames and passwords on the wire as well as eavesdropping
at the user's session. The alternative safer ssh and sftp access
methods are also provided, however many users need these legacy access
modes. On the positive side, access to these services is controlled by
TCP wrappers, and we recommend using TCP wrappers to limit access and
to enhance logging. Access to the root account should be disabled for
both telnet and ftp (root is present in /etc/ftpusers file). We also
recommend creating an appropriate /etc/issue.net file with appropriate
warnings that will be displayed with each login prompts, and editing
the login prompt entry in /etc/gettydefs to mask the operating system
type.

   Pop3 and imap

These services provide mailbox access to authorized users. Both
services operate in plaintext and therefore are potentially vulnerable
to snooping. There are SSL-enabled versions of these programs, we
recommend investigating possibility of using them. Access to these
services can be controlled and logged using TCP wrappers.

   Finger

Finger provides information about users that are currently logged in
the system. This service is accessible by arbitrary remote users,
because no authentication is required. We recommend turning this
service off (by commenting out or removing the corresponding line from
/etc/inetd.conf.)

   Tftp (trivial file transfer protocol)

These service is used by several X terminals on the local net. Access to this service is currently unrestricted. We recommend to allow access to the tftp server only to the selected set of X terminals. TCP wrappers can be used to achieve that.

Comsat

This service receives reports of incoming mail and notifies users who request that using the biff command. We recommend restricting access to this service to local clients only. This is a UDP based service, so to control access to it the latest version of TCP wrappers is needed. Unless it is really necessary, we recommend disabling this service (by commenting out the corresponding line in inetd.conf).

Ntalk

This service notifies a user when another user (possibly on a remote host) wants to initiate a conversation using the talk command and then facilitates a bi-directional conversation. We recommend disabling this service unless it is really necessary for the site.

Dtspc

This is the CDE subprocess control service. Since local users do not use CDE, we recommend either turning off this service, or using TCP wrappers to restrict access to the set of local machines that import users home directories and may run CDE.

Rpc.cmsd

This is the CDE calendar manager service daemon. This utility is registered as rpc/udp and as such can not be managed by TCP wrappers. We recommend turning this service off.

NFS and NIS

Summary

Even though original implementations of NFS and NIS did not provide
for security, access to NFS and NIS services on the local host can be
limited in /etc/exports and /var/yp/securenets files. The NFS
configuration file /etc/exports contains two extraneous lines that can
be removed, root level access from remote hosts to the exported
file systems can be disabled.

Details

   NIS

The host is the master NIS server for the local domain. The local
domain includes 3 workstations on the local net. This is a legacy
solution to password management problem across different platforms. In
the local case, there are Tru64 Alpha based systems and PC based
FreeBSD and OpenBSD systems.

The Tru64 NIS implementation allows for restricting access to NIS
services by means of the /etc/yp/securenets file. In the local case,
this file restricts access to the hosts from the local subnet. Tru64
NIS implementation however allows for more fine grained access control
in the /etc/yp/securenets file by means of specifying a network mask
as well as an IP address. In such manner only the hosts that actually
need NIS services have access to them on our host. For example, the
following line put in the NIS access control file /etc/yp/securenets
will grant access to the single host with IP 10.1.2.3,

255.255.255.255 10.1.2.3

We recommend identiyying all the hosts on the departmental network that
need to use NIS services on the host and enable access to those and
only those hosts by adding lines to /etc/yp/securenets that are
similar to our example above.

Root account is not handled by NIS but rather by the local /etc/passwd
files on each machine. This is a good practice. However, unless access
to the NIS services on our host is restricted using technique shown
above, an arbitrary host on the local subnet will be able to bind to
the NIS server and obtain NIS maps, in particular, the password file.
Once the password file with encrypted user passwords is obtained,
the attacker may proceed with "cracking" encrypted passwords (using
one of several freely available cracking programs such as "John the
Ripper" or "Crack"). Two factors simplify the attacker's task, first,
NIS is using the weakest form of password encryption (DES) for
interplatform compatibility reasons, and, second, practice shows that

unsophisticated users often tend to choose easily guessable passwords such as proper names or dictionary words. Crackers may pre-encrypt dictionaries and slight variations of dictionary words and proper names and in this way speed up password cracking significantly. And only one password needs to be compromised for an attacker to gain access to the system under the guises of a legitimate user.

NFS

The host is a file server for another Tru64 workstation, two BSD workstations, nine X terminals, and two PC's. The non-comment lines from the /etc/exports follow (host names are obscured with asterisks).

```
/usr/users              -root=******:*****:***** ****** ***** *****
/usr/local        -root=****** ******
/var/spool/mail         -root=******:*****:***** ****** ***** *****
/x                      -root=****** ******

/usr/lib/X11/ncd  ******* ******* ******* ******* *******
   ******* *** *** *******

/usr/users/***                ********
/usr/users/***                ******
```

The first group of lines export users home directories (/usr/users) and the users mailbox directory (/var/spool/mail) to the workstations. The applications directories (/usr/local and /x) are exported only to the workstation of the same architecture as the host. The root user on each of these workstations is mapped to the root user on the file server. Unless necessary for normal operations, we recommend to map the root user to the user 'nobody' which is the default setting. Mapping the root user on an NFS client to the root user on the NFS server presents a security risk, because of the trust relationship that arises from this arrangement. If an attacker was successful in breaking the root account on a client system, he automatically gets root level access to the exported directories on the NFS server. If executable applications directories were exported with root mapping to root options then an attacker may change executables on the server and take over the server as well. If an attacker already has access to the NFS server as an unprivileged user then he only needs to create an SUID root shell on the exported partition where he already has root level access and as a result he obtains root level access on the NFS server as well.

In the spirit of granting only necessary privileges, we recommend

exporting applications directories (/usr/local, /x) read only.

The second group of lines enables nine X terminals that also boot from this host.

The third group refers to the machines that are no longer on the local network. We recommend removing these lines altogether.

As can be seen from the /etc/exports file given above, Access to the NFS server is restricted only to the NFS clients on the local subnet that actually need this access. This is a good practice.

Printing

The host acts as an intermediate print server for several departmental Unix/Linux/BSD workstations. The vendor's version of the Berkeley printing daemon lpd manages four networked departmental laser printers.

Remote access to lpd is controlled by two files, /etc/hosts.equiv and /etc/hosts.lpd. Any host whose name is present in these files will be allowed to connect to the lpd daemon. Currently, both files contain only hostnames of selected hosts from the local net. Some of these names belong to hosts that no longer exist on the local net. We recommend correcting the access files so that they contain only the names of hosts that actually use lpd services on the system. The fact that lpd grants access to the hosts listed in /etc/hosts.equiv (in addition to the ones in /etc/hosts.lpd) often comes as a surprise to the people who only have experience with SysV flavors of Unix. The Tru64 Unix shows its BSD side here similar to its predecessor, Ultrix. The following is a quote from the lpd(8) man page for Tru64 4.0F:

   "All requests must originate from one of the machines listed
   in the /etc/hosts.equiv or /etc/hosts.lpd file."

The author of the present report conducted the following experiment with the host. A (Linux) host on a local net was chosen, and its name was removed from /etc/hosts.lpd and added to /etc/hosts.equiv and lpd on the host that is the subject of this report was restarted. After that print requests from the Linux host have been accepted and processed by our host. We would also like to note that all open source BSD systems (FreeBSD, NetBSD, and OpenBSD) exhibit similar

behaviour. Despite that property of lpd we strongly recommend to keep
the access list for lpd in one file, /etc/hosts.lpd. The file
/etc/hosts.equiv historically contains the names of the hosts that
are trusted by r-services (rlogin, rshell, rexec). These services
present security risks (such as transmitting sensitive authentication
information in the cleartext) and should be disabled. The secure shell
can provide similar functionality in much more secure manner. We
recommend to maintain an empty file /etc/hosts.equiv and make
unwritable by anybody (chmod 444 /etc/hosts.equiv).

The legacy Apple laser printer is served by aarpd (AppleTalk address
resolution protocol daemon). This daemon runs with root privileges
and it listens on tcp port 973. The design of this program does not
allow for restricting access to only hosts on the local network,
therefore we recommend either disabling it altogether or modifying
the source code to include such access control.

Security related software and services

Summary

A number of security related programs that are installed on the host
have newer versions available. It is recommended to update OpenSSL,
OpenSSH, and TCP wrappers packages to the latest released versions. To
store the log files safely, we recommend setting up a hardened log
host on the local network and configuring syslogd for remote logging
to that host. Since switching to the shadow password file is not
practical for the site, we recommend choosing the root password
carefully. A handy utility, lsof (it lists open files and sockets as
well as processes that opened those), was installed and used in
the process of auditing the local host. We recommend using it for
daily system maintenance to watch sockets and files usage.

Details

OpenSSL

OpenSSL version 0.9.5a (released in April 2000) is installed. Since
the newer version 0.9.6b is freely available, we recommend upgrading.

OpenSSH

OpenSSH version 2.2.0p1 is installed. The newer version 2.9 is freely

available (that also includes support for sftp, a secure replacement for ftp), we recommend upgrading.

   Syslogd and binlogd

The syslog logs kern, user, mail, daemon, auth, syslog, lpr messages at the debug level into local files. The log files are rotated daily and kept for a week. We recommend setting a separate machine that will be a centralized loghost for the department. All logs should be sent to the loghost and periodically processed by some log watching utility to spot potential problems. The logs on the loghost can be kept for a long time, for example for several months.

The vendor supplied syslogd program does not accept log messages from remote hosts by default. This is a good practice! If the local host has to start accepting log messages from remote hosts, then the names of the authorized remote hosts must be in the syslogd access control file /etc/syslog.auth. Currently, this file is absent from the local host and therefore no log messages from remote hosts are accepted.

Binlogd is a proprietary daemon that logs kernel messages on Tru64 systems. The log file (/var/adm/binary.errlog) has binary format and can be read by a special utility (uerf or dia). Binlogd is capable of listening on UDP port 706 for binlogd messages from other Tru64 hosts. In the current configuration, however, binlogd does accept messages on the UDP port 709, this is confirmed by the output of netstat and lsof.

   Password files

The regular /etc/passwd file is used for root password. User passwords are stored in NIS password map. In either case these files are accessible to all local users. Since NIS only restricts access to the local subnet, a person with a computer on a local subnet can get NIS password map even without a local user account on the host. Unfortunately, the system allows to use shadow password files only in the context of the proprietary vendor's security integration architecture (SIA). The administrative overhead and overall management complexity associated with SIA make the switch to SIA not practical for this site. In view of this we recommend choosing the root password especially carefully to make a dictionary attack less effective. The standard recommendations are presence of both upper and lower case characters, special symbols, digits. The password must be eight characters long (the system maximum) and must not be a dictionary word or a proper name.

TCP wrappers

The TCP wrappers program is installed (version 7.5). The newer version
7.6 is freely available, we recommend upgrading. The newer version
allows for the finer access control and better logging facilities. It
also can handle UDP as well as TCP connections.

NTP (Network time daemon)

The host is running the network time daemon xntpd in the client mode.
The time is synchronized with 3 organization wide time servers.
The xntpd is vendor's stock version. Recently a vulnerability in an
open source xntpd implementation was discovered that may affect vendor
versions of xntpd as well. We recommend applying the latest patchkit
from the vendor. Running a network time client is a good practice.

Lsof utility

The lsof utility is an open source program available for several
flavors of Unix. We downloaded, compiled and installed this utility in
the process of conducting the audit of the local host. This program
can access system's kernel memory and extract information about open
files and sockets. An administrator can then examine in details what
programs on his system open which files and sockets. We recommend
keeping this utility on the system and making use of it in daily
maintenance tasks.

X Window security

Summary

The host provides X Window services and X font server services to
nine X terminals on the local network. Configuration files are
accessed through several layers of indirection using symbolic links
and the recommended scheme from the system manuals is ignored. We
recommend to rearrange configuration files to comply with the vendor's
scheme. In addition, the Xaccess file can be used better to control
access to the host.

Details

The host provides X window services to several X terminals on

the local network.

Dtlogin

The dtlogin program manages user access to the server. Its
configuration file is /usr/dt/config/Xconfig. However, this is a link
to another location which in turn is also a symbolic link to yet
another file. We recommend to simplify the configuration files by
leaving all default files in the default directory, /usr/dt/config,
and putting all locally customized files in /etc/dt/config.

The Xaccess file defines access control features. Currently, all
hosts are granted access and all services. We recommend restricting
access to hosts on the local subnet, the only ones that need
the service.

Font server

The X Window font server (xfs) is started at boot. Its (default)
configuration file is /var/X11/fs/config, and it is a symbolic link to
the file config.v2 in the same directory. Unfortunately, the vendor's
implementation of the font server does not provide any means for
controlling access. We recommend cooperating with the local network
operations department in using organization's firewalls for
restricting access to the font server from outside hosts.

Filesystem security

Summary

A number of files have their suid and sgid permission bits set that
do not actually need these permissions. We recommend disabling suid
and sgid bits for these files. Currently, the site has no backup
solution. We recommend acquiring a tape backup drive and setting up
regular backups.

Details

Set user id files

The executable set user id (suid) files allow regular users execute
programs with another user's (usually root) privileges. The suid
programs are convenient in some cases (the password program is an

example), however they may present security risk if not properly used. A faulty suid root program may provide a regular user with root level access to the system. We must note that if an SUID root program is not supposed to be used by regular users, then its SUID bit can be turned off (chmod u-s program). After that root will be able to use these programs as usual, but the regualr users may no longer be able to use them.

Below is the complete list of suid root files on the system.

System commands:

/sbin/df

This command displays statistics on free disk space. It needs to be SUID root in order to access kernel filesystem data structures.

/sbin/killall

This command terminates all processes started by the user, except the calling process. It provides a convenient means of killing all processes created by the shell that the user controls. We are not sure why this command needs to be SUID (probably to access kernel process control data), but if SUID root is disabled, then regular users can not successfully use killall.

/sbin/ps

This program displays current process status. It needs to be SUID root in order to access kernel process control data.

/sbin/ping

This program tests network connectivity by sending ICMP ECHO_REQUEST packets to network hosts. It needs to be SUID root in order to access raw sockets (to create ICMP packets). However, its man page (ping(8)) does not recommend using this program for purposes other than testing and debugging the network, so its SUID root bit can be turned off.

/sbin/showfdmn
/sbin/showfsets

These two commands display file domain attributes and information filesets in a file domain, respectively. These utilities are parts of administrative interface of AdvFS, the Tru64 proprietary advanced file system. It is very likely that only the superuser will need to

use these utilities, therefore the SUID root bits can be turned off.

/usr/local/stow/top/bin/top

This program displays periodically refreshed information about currently running processes. It needs SUID root bit in order to access kernel's process control data structures from /dev/kmem.

/usr/var/adm/ris/bin/ris_pax

This program is a part of the RIS (Remote Installation Service) system package. The RIS package is not used on our host. We recommend either removing the RIS package or turning the SUID and SGID bits on this program off.

Various users at jobs (user names are obscured by asterisks):
/usr/var/spool/cron/atjobs/***.995605261.f
/usr/var/spool/cron/atjobs/***.997851660.f
/usr/var/spool/cron/atjobs/***.995605260.f
/usr/var/spool/cron/atjobs/***.998283660.f
/usr/var/spool/cron/atjobs/***.995778060.f
/usr/var/spool/cron/atjobs/***.996266940.f
/usr/var/spool/cron/atjobs/***.997647300.f

These programs are SUID to the corresponding regular user id and therefore are not considered as dangerous as SUID root programs.

/usr/sbin/acct/accton

This program truns on/off system process accounting. We think that only the root user should handle this anyway and therefore recommend turning the SUID bit off.

/usr/sbin/killall

This is he same program as /sbin/killall.

/usr/sbin/ping

This is a dynamically linked version of /sbin/ping (which is linked statically). Same recommendation as for /sbin/ping.

/usr/sbin/pppd

This is the Point-to-Point Protocol daemon. Only root should be able to run this daemon and, furthermore, this program is not useful for

the local site. Therefore, we recommend turning the SUID and SGID bits
on pppd off.

/usr/sbin/startslip

This program configures SLIP (serial line IP) connections. It is used
by the UUCP system, but the UUCP system itself if not used by
the local site. Therefore, we recommend either turning SUID and SGID
bits on startslip or uninstalling SLIP and UUCP packages alltogether.

/usr/sbin/timedc

This is a control interface to the timed daemon. Having an accurate
system time is crucial to the security of the system (for an audit
trace), therefore we recommend removing the SUID bit from timedc.

/usr/sbin/traceroute

This program traces an approximate route that packets travel from
the local host to a remote host. It needs the SUID root bit in order
to create custom UDP packets with increasing TTL (time-to-live)
fields. Bounces from intermediate routers trace the path that packets
travel. If this program is not needed by regular users (very likely),
then we recommend turning the SUID bit off on traceroute.

/usr/sbin/lpc

This is a control interface to the printing daemon, lpd. We believe
that the superuser (root) only should be able to control the printing
daemon, therefore the SUID root bit can be removed from lpc.

/usr/sbin/dop

This command is a part of the system manual package according to the
software manager database in /usr/.smdb., however there is no man
page on dop on the system nor on Compaq's online man page reference
site. However, after disabling SUID bit on dop, regular users are
still able to run man ansd man -k, therefore we recommend disabling
SUID bit on dop.

/usr/sbin/sendmail

This is the infamous e-mail all-in-one program, the Berkeley sendmail.
This program needs to be SUID root in order to place users mail on
the outgoing mail queue. We devote a separate section to sendmail in
this report, please consult that section for further information about

sendmail.

/usr/sbin/runclass

This program runs a command in a specific scheduler class. Regular
users of the host do not need this functionality and we recommend
turning off the SUID bit on runclass.

/usr/sbin/mailq.PreUPD

This is an obsolete copy of sendmail that remained after past
upgrades. It can be safely removed.

/usr/sbin/smtpd

This is an old link to an old sendmail program. It can be safely
removed.

/usr/sbin/sendmail.v8.8.8

This is another old sendmail copy. It can be safely removed.

/usr/sbin/collect

This program collects system activity statistics and periodically
displays it on the terminal. It needs the SUID root bit in order to
access kernel process control data. If regular users of the system
do not need to use this program, then the SUID bit can be turned off.

/usr/bin/X11/dxconsole
/usr/bin/X11/dxterm
/usr/bin/X11/xterm
/usr/bin/X11/dxpause
/usr/bin/X11/dxsysinfo
/usr/bin/X11/xconsole
/usr/dt/bin/dtaction
/usr/dt/bin/dtappgather
/usr/dt/bin/dtprintinfo
/usr/dt/bin/dtsession
/usr/dt/bin/dtterm

These programs are part of the X Window system. These programs need
SUID root bits in order to access terminal and screen devices.

/usr/bin/mh/inc
/usr/bin/mh/msgchk

These programs are parts of the mh (graphical Mail Handler) package
that comes with CDE. These programs incorporate and check for new mail
and they need SUID bits only if one regular user will need to check
other people's mail. This is not the case on the local site, and
therefore we recommend removing SUID root bits from these programs.

/usr/bin/at

This program runs commands at a later time. It needs SUID root bit
to create users at files in the system at spool directory
/usr/var/spool/cron/atjobs.

/usr/bin/binmail

This is another name for /usr/bin/mail.

/usr/bin/chfn
/usr/bin/chsh
/usr/bin/passwd

These programs change password file insformation. They need the SUID
root bits in order to update the password file.

/usr/bin/crontab

This program submits a schedule of commands to cron. It needs the SUID
bit to create users crontab files in /usr/var/spool/cron/crontabs,
the system crontab spool directory.

/usr/bin/ipcs

This program reports Interprocess Communication (IPC) facility status.
It is unlikely to be used by the regular users and therefore the SUID
root bit (otherwise necessary to access kernel data structures) can be
turned off.

/usr/bin/login

This program signs the user on to the system. It needs SUID root bit
to update accounting files and to manipulate the terminal devices.

/usr/bin/newgrp

This command changes primary group identification of a shell process.
It needs SUID root in order to access and write kernel process status

structures.

/usr/bin/ps

This is a dynamically linked version of ps. Same recommendations as
for /sbin/ps.

/usr/bin/su

This command substitutes user id temporarily. It needs SUID root bit
to function properly.

/usr/bin/uptime

This command shows how long a system has been running. It needs SUID
root bit to access kernel data structures.

/usr/bin/vmstat

This program reports virtual memory statistics. It needs SUID root bit
to access kernel data structures.

/usr/bin/ct

This command dials an attached terminal and issues a login process.
It needs the SUID root bit to manipulate the serial device (modem),
however it is useless for the local site, and we recommend to turn
the SUID root bit off.

/usr/bin/rcp
/usr/bin/rlogin
/usr/bin/rsh

These three commands are the so called r-commands, remote copy,
remote login, remote shell. These programs are obsoleted by the
secure shell program which provides the same (and more) functionality
over an encrypted channel as opposed to the cleartext.

/usr/bin/nrdist
/usr/bin/rdist

These are remote file distribution programs. The SUID root bits on
these programs allow regular users to maintain identical directories
on different hosts, but if this functionality is not needed and only
root is using (n)rdist, then the SUID root bits can be removed.

```
/usr/bin/lpq
/usr/bin/lpr
/usr/bin/lprm
```

These are user level utilities that manupulate printer queues. These
programs need SUID root bits to operate properly.

```
/usr/bin/llogin
```

This command connects to a LAT (Local Area Transport) service.
Since LAT is not used on the local site, the SUID root bit can be
turned off.

```
/usr/bin/uucp
/usr/bin/uustat
/usr/bin/uux
```

These commands are parts of the UUCP package. As with other parts of
UUCP package, we recommend either removing the SUID root bits or
uninstalling the UUCP package altogether.

```
/usr/bin/mail
```

This program sends and displays e-mail messages. It needs SUID root
bit to inject messages into the outgoing queue.

```
/usr/bin/w
```

This command prints the summary of current system activity. It needs
the SUID root bit to access the relevant information in kernel data
structures.

```
/usr/bin/ssh
```

This is the SSH client program. It needs SUID root bit for the same
reasons as the login program.

```
/usr/lbin/chgpt
/usr/lbin/rdsym
/usr/lbin/slvmod
```

These are undocumented binaries, they are registered in the system
software database in /usr/.smdb. as a part of OSFBASE440 package.
Neither system manpages nor online Compaq version of man pages contain
info about these programs. We recommend temporarily disabling the SUID

root bit on these programs and see if regular user will have problems
possibly related to these programs.

/usr/lbin/ex3.7preserve
/usr/lbin/ex3.7recover
/usr/lbin/expreserve
/usr/lbin/exrecover

These commands preserve/recover files edited by the ex/vi editor when
the editor exits abnormally. Since on the local host the temporary
directories /tmp and /var/tmp have read/write permission for all users
and the sticky bit is set on these directories as well we recommend
removing the SUID root bit from these commands. Regular users should
be able to continue to use these programs (indirectly from vi) without
problems.

/usr/lbin/lpd

The printing daemon itself. We recommend truning the SUID bit off,
since only root should be able to control the printing daemon.

/usr/lib/mh/spop

This is a POP mailer. This program delivers mail to the POP spool
area on the local system. It needs SUID root bit to function properly.

/usr/lib/uucp/uucico
/usr/lib/uucp/uumonitor
/usr/lib/uucp/uuxqt

These commands are parts of the UUCP package. Since the UUCP package
is not needed by the local site we recommend either removing it or
turning the SUID bits off.

/usr/opt/s5/bin/df
/usr/opt/s5/sbin/killall

These are the same as /usr/bin/df and /usr/sbin/killall.

/usr/opt/BRX440/BRXSOAKIT440/bin/asavegrp
/usr/opt/BRX440/BRXSOAKIT440/bin/nsralist
/usr/opt/BRX440/BRXSOAKIT440/bin/nsrarchive
/usr/opt/BRX440/BRXSOAKIT440/bin/nsrmm
/usr/opt/BRX440/BRXSOAKIT440/bin/nwrecover
/usr/opt/BRX440/BRXSOAKIT440/bin/nwretrieve
/usr/opt/BRX440/BRXSOAKIT440/bin/recover

```
/usr/opt/BRX440/BRXSOAKIT440/bin/save
/usr/opt/BRX440/BRXSOAKIT440/bin/savefs
/usr/opt/BRX440/BRXSOAKIT440/bin/savegrp
/usr/opt/BRX440/BRXSOAKIT440/bin/srecover
```

These programs constitute parts of a backup package. Usually backup is
done by a cron program running under a regualr user's (backup) id, and
the SUID bits are needed in order to access system devices that are
being backed up. However, the local site does not currently use this
package (even though it is successfully installed) and therefore we
recommend either removing this package alltogether or removing SUID
root bits from these commands. A potential security problem with these
programs (and SUID root backup programs in general) is that
an arbitrary user can create backups to a file and than read any files
from the backed up partition. If the backed up partition contained the
shadow password file for example we have a serious problem. We must
note, however, that our host is using a plain /etc/passwd file and
the NIS system to store encrypted passwords. In either case, all
encrypted passwords are already available to the regular users.
A regular user may however use the above tools to create backups of
the system partitions (/, /var, /usr), then edit the backup files,
and finally "restore" the modified partition, thus effectively
overriding any filesystem access protection. This is a very SERIOUS
security problem!

```
/usr/opt/pm/bin/pmgr
```

This is the performance manager. It is used to monitor nodes in a
distributed computer environment. Since the local system is standalone
we recommend turning off the SUID bit on this program. This program
is not a part of the base system and it can be removed (packages
PMGRBASE425 and PMGRGUI425).

```
/usr/tcb/bin/edauth
/usr/tcb/bin/dxchpwd
```

These are parts of the Trusted Computing Base system (enchanced
security, package OSFC2SEC440). This feature is not used on the local
host, therefore the SUID bits on these programs can be turned off.

The following are files in a user's home directory. They are suid this
user. The user's name is obscured by asterisks.

```
/usr/users/********/bin/artemis/mntpanda
/usr/users/********/bin/artemis/umtpanda
/usr/users/********/bin/athena/mntpanda
```

/usr/users/********/bin/athena/umtpanda

The following two files are actually empty. They can be removed.
/x/pkg/pkg/build/bin/test/a.0
/x/pkg/pkg/build/bin/test/a

So far, most of the suid files are suid root files which belong to
the system packages. The exceptions are the four user files and
the two empty files listed above. These files either can be removed
or their permissions can be modified to disable the suid bit.

    Set group id files

Below are all set group id files on the system.

An sgid 'mem' system binary:
/sbin/arp

This program manages address resolution protocol system tables.
We believe that only root should be able to manage arp tables,
therefore the SGID bit can be turned off.

An sgid 'mail' add-on program:
/usr/local/bin/elm

The elm program usually needs SGID mail bit if the system mail
spool is not world writable. However, /var/spool/mail is world
writable (with the sticky bit set, so that only owners can remove
their files), therefore SGID mail bit is not necessary and it can
be turned off.

These directories and files are wrongfully sgid 'system'. They are
source files and they do not need execute permissions at all.
/usr/local/src/openssh/openssl-0.9.6/MacOS
/usr/local/src/openssh/openssl-0.9.6/MacOS/GetHTTPS.src
/usr/local/src/openssh/openssl-0.9.6/VMS
/usr/local/src/openssh/openssl-0.9.6/apps
/usr/local/src/openssh/openssl-0.9.6/apps/demoCA
/usr/local/src/openssh/openssl-0.9.6/apps/demoCA/private
/usr/local/src/openssh/openssl-0.9.6/apps/set
/usr/local/src/openssh/openssl-0.9.6/bugs
/usr/local/src/openssh/openssl-0.9.6/certs
/usr/local/src/openssh/openssl-0.9.6/certs/expired
/usr/local/src/openssh/openssl-0.9.6/crypto

```
/usr/local/src/openssh/openssl-0.9.6/crypto/asn1
/usr/local/src/openssh/openssl-0.9.6/crypto/bf
/usr/local/src/openssh/openssl-0.9.6/crypto/bf/asm
/usr/local/src/openssh/openssl-0.9.6/crypto/bio
/usr/local/src/openssh/openssl-0.9.6/crypto/bn
/usr/local/src/openssh/openssl-0.9.6/crypto/bn/asm
/usr/local/src/openssh/openssl-0.9.6/crypto/bn/asm/alpha
/usr/local/src/openssh/openssl-0.9.6/crypto/bn/asm/alpha.works
/usr/local/src/openssh/openssl-0.9.6/crypto/bn/asm/x86
/usr/local/src/openssh/openssl-0.9.6/crypto/buffer
/usr/local/src/openssh/openssl-0.9.6/crypto/cast
/usr/local/src/openssh/openssl-0.9.6/crypto/cast/asm
/usr/local/src/openssh/openssl-0.9.6/crypto/comp
/usr/local/src/openssh/openssl-0.9.6/crypto/conf
/usr/local/src/openssh/openssl-0.9.6/crypto/des
/usr/local/src/openssh/openssl-0.9.6/crypto/des/asm
/usr/local/src/openssh/openssl-0.9.6/crypto/des/t
/usr/local/src/openssh/openssl-0.9.6/crypto/des/times
/usr/local/src/openssh/openssl-0.9.6/crypto/dh
/usr/local/src/openssh/openssl-0.9.6/crypto/dsa
/usr/local/src/openssh/openssl-0.9.6/crypto/dso
/usr/local/src/openssh/openssl-0.9.6/crypto/err
/usr/local/src/openssh/openssl-0.9.6/crypto/evp
/usr/local/src/openssh/openssl-0.9.6/crypto/hmac
/usr/local/src/openssh/openssl-0.9.6/crypto/idea
/usr/local/src/openssh/openssl-0.9.6/crypto/lhash
/usr/local/src/openssh/openssl-0.9.6/crypto/md2
/usr/local/src/openssh/openssl-0.9.6/crypto/md4
/usr/local/src/openssh/openssl-0.9.6/crypto/md5
/usr/local/src/openssh/openssl-0.9.6/crypto/md5/asm
/usr/local/src/openssh/openssl-0.9.6/crypto/mdc2
/usr/local/src/openssh/openssl-0.9.6/crypto/objects
/usr/local/src/openssh/openssl-0.9.6/crypto/pem
/usr/local/src/openssh/openssl-0.9.6/crypto/perlasm
/usr/local/src/openssh/openssl-0.9.6/crypto/pkcs12
/usr/local/src/openssh/openssl-0.9.6/crypto/pkcs7
/usr/local/src/openssh/openssl-0.9.6/crypto/pkcs7/p7
/usr/local/src/openssh/openssl-0.9.6/crypto/pkcs7/t
/usr/local/src/openssh/openssl-0.9.6/crypto/rand
/usr/local/src/openssh/openssl-0.9.6/crypto/rc2
/usr/local/src/openssh/openssl-0.9.6/crypto/rc4
/usr/local/src/openssh/openssl-0.9.6/crypto/rc4/asm
/usr/local/src/openssh/openssl-0.9.6/crypto/rc5
/usr/local/src/openssh/openssl-0.9.6/crypto/rc5/asm
/usr/local/src/openssh/openssl-0.9.6/crypto/ripemd
/usr/local/src/openssh/openssl-0.9.6/crypto/ripemd/asm
```

```
/usr/local/src/openssh/openssl-0.9.6/crypto/rsa
/usr/local/src/openssh/openssl-0.9.6/crypto/sha
/usr/local/src/openssh/openssl-0.9.6/crypto/sha/asm
/usr/local/src/openssh/openssl-0.9.6/crypto/stack
/usr/local/src/openssh/openssl-0.9.6/crypto/threads
/usr/local/src/openssh/openssl-0.9.6/crypto/txt_db
/usr/local/src/openssh/openssl-0.9.6/crypto/x509
/usr/local/src/openssh/openssl-0.9.6/crypto/x509v3
/usr/local/src/openssh/openssl-0.9.6/demos
/usr/local/src/openssh/openssl-0.9.6/demos/bio
/usr/local/src/openssh/openssl-0.9.6/demos/eay
/usr/local/src/openssh/openssl-0.9.6/demos/maurice
/usr/local/src/openssh/openssl-0.9.6/demos/pkcs12
/usr/local/src/openssh/openssl-0.9.6/demos/prime
/usr/local/src/openssh/openssl-0.9.6/demos/sign
/usr/local/src/openssh/openssl-0.9.6/demos/ssl
/usr/local/src/openssh/openssl-0.9.6/demos/state_machine
/usr/local/src/openssh/openssl-0.9.6/doc
/usr/local/src/openssh/openssl-0.9.6/doc/apps
/usr/local/src/openssh/openssl-0.9.6/doc/crypto
/usr/local/src/openssh/openssl-0.9.6/doc/ssl
/usr/local/src/openssh/openssl-0.9.6/include
/usr/local/src/openssh/openssl-0.9.6/include/openssl
/usr/local/src/openssh/openssl-0.9.6/ms
/usr/local/src/openssh/openssl-0.9.6/perl
/usr/local/src/openssh/openssl-0.9.6/perl/t
/usr/local/src/openssh/openssl-0.9.6/rsaref
/usr/local/src/openssh/openssl-0.9.6/shlib
/usr/local/src/openssh/openssl-0.9.6/ssl
/usr/local/src/openssh/openssl-0.9.6/test
/usr/local/src/openssh/openssl-0.9.6/times
/usr/local/src/openssh/openssl-0.9.6/times/090
/usr/local/src/openssh/openssl-0.9.6/times/091
/usr/local/src/openssh/openssl-0.9.6/times/x86
/usr/local/src/openssh/openssl-0.9.6/tools
/usr/local/src/openssh/openssl-0.9.6/util
/usr/local/src/openssh/openssl-0.9.6/util/pl
/usr/local/src/elm2.5.3
/usr/local/src/elm2.5.3/doc
/usr/local/src/elm2.5.3/hdrs
/usr/local/src/elm2.5.3/lib
/usr/local/src/elm2.5.3/nls
/usr/local/src/elm2.5.3/nls/gencat
/usr/local/src/elm2.5.3/nls/C
/usr/local/src/elm2.5.3/nls/C/C
/usr/local/src/elm2.5.3/nls/C/C/C
```

```
/usr/local/src/elm2.5.3/src
/usr/local/src/elm2.5.3/test
/usr/local/src/elm2.5.3/utils
```

SGID programs:

```
/usr/var/adm/ris/bin/ris_pax
```

See our remarks about this program in the previous section on SUID
files.

```
/usr/var/spool/calendar
```

This is the spool directory for the CDE calendar program. The SGID bit
on this directory sets ownership of files to the group 'daemon' as
opposed to the standard group 'users'.

```
/usr/sbin/quot
```

This is an SGID mem program. Its manual page specifies that only root
may use this program, therefore the SGID bit is extraneous and can be
removed.

```
/usr/sbin/arp
```

This is the same file as /sbin/arp considered above.

```
/usr/sbin/wall
```

This program sends a message to all users on the local system. It is
an SGID terminal program. It needs the SGID bit to be able to write
to the terminals owned by other users.

```
/usr/sbin/netstat
```

This program displays network statistics. It needs the SGID mem bit
to access kernel data structures through the /dev/kmem device.

```
/usr/sbin/srconfig
```

This program displays and controls source routing finctions and
parameters for communication on token ring networks. Since token ring
networks are not used at the local site, this program is not needed.
We recommend removing the SGID bit from srconfig. Besides, only root
should be able to control such parameters anyway.

/usr/sbin/pppd
/usr/sbin/startslip

These programs control serial link communications. See remarks about
these programs in the previous section on SUID files.

/usr/sbin/trpt

This program transliterates protocol trace and is used for debugging
sockets. It needs the SGID mem bit to access kernel's buffers and
protocol control blocks.

/usr/sbin/atmsig

This is an ATM interface configuration program. ATM is not used at
the local site, and besides we believe that only root should be able
to configure network interfaces, therefore we recommend turning
the SGID bit off.

/usr/sbin/vquota

This SGID operator program displays disk usage and limits quota info.
The local side does not use quota, therefore this program is not
necessary for the normal function of the system and the SGID bit can
be removed.

/usr/bin/X11/demos/cpuinfo

This is a graphical cpu load display program. It needs the SGID mem
bit to access kernel data structures.

/usr/bin/pfstat

This program prints packet filter status information. Packet filters
are not used on the local host, therefore we recommend turning off
the SGID bit on pfstat.

/usr/bin/write

This program sends message to other users. It is similar to the wall
program above.

/usr/bin/cu
/usr/bin/tip

These two porgrams are used to connect to remote systems over serial

lines. Serial line connectivity is not used on the local host, so
the SGID bits can be removed.

/usr/bin/nfsstat

This SGID mem program displays NFS statistics. It needs teh SGID mem
bit to access kernel data structures through the /dev/kmem device.

/usr/bin/uucp
/usr/bin/uuname
/usr/bin/uustat
/usr/bin/uux
/usr/lib/uucp/uucico
/usr/lib/uucp/uuxqt

These programs are parts of UUCP package. We recommend removing the
UUCP programs altogether or at least disabling their SGID bits.

/usr/dt/bin/dtaction

This is an SGID system program. It invokes a CDE action with specified
arguments.

/usr/dt/bin/dtmail
/usr/dt/bin/dtmailpr
/usr/dt/bin/mailcv

These are SGID mail programs. The SGID mail bit is actually not
necessary, see our remarks on the elm program above.

/usr/opt/pm/bin/pmgr

See our remarks on this program in the previous section on SUID files.

These are at jobs sgid 'users' (user names are obscured by asterisks):
/usr/var/spool/cron/atjobs/***.995605261.f
/usr/var/spool/cron/atjobs/***.997851660.f
/usr/var/spool/cron/atjobs/***.995605260.f
/usr/var/spool/cron/atjobs/***.998283660.f
/usr/var/spool/cron/atjobs/***.995778060.f
/usr/var/spool/cron/atjobs/***.996266940.f
/usr/var/spool/cron/atjobs/***.997647300.f

These directories are wrongfully sgid 'users' (user names are obscured
by asterisks):

```
/usr/users/*******/nauty/gtools/gtools10
/usr/users/*******/nauty/nauty20
```

Empty files, can be removed:
```
/x/pkg/pkg/build/bin/test/a.0
/x/pkg/pkg/build/bin/test/a
```

Likewise, the empty files can be removed and sgid 'users' bits can be turned off.

   Backup

No filesystem backup procedures are currently implemented. A full set of backup tapes could improve chances of successful recovery in a case of system's compromise or malfunction. We think that backup is essential for reliable functioning of the system and we recommend getting a tape backup drive. (The author of this report regularly creates backup copies of user home directories partition and stores them on his workstation, but this cannot replace regular full backups on tape.)

         TCP Wrappers

We recommend using TCP wrappers to control access to a number of inetd based network services and, in general, any TCP wrappers enabled network services. In this section we will briefly describe how to set up TCP wrappers on the system.

The latest release of TCP wrappers is 7.6 and it can be obtained from the author's (Wietze Venema) site at ftp://ftp.porcupine.org in /pub/security/tcp_wrappers_7.6.tar.gz. Once unzipped and untarred, the tcp_wrappers_7.6 source directory contains the README file that covers installation, configuration, and design of TCP wrappers. We will not duplicate the instructions here, it suffices to say that TCP wrappers compile and function in a uniform way on the majority of Unix platforms.

What can TCP wrappers do for the security of the system? In short, it provides access control to network services, logging of attemted connections (audit track), and setting up customized commands to process connections. Access control is provided by means of /etc/hosts.allow and /etc/hosts.deny files. A system administrator

can specify what services are allowed (or prohibited) to accept
connections from specified network addresses, together with logging
options and custom commands. Despite its name, the latest TCP
wrappers package can also be used to control access to UDP based
services as well. We must note however, that some UDP services may
"linger" after processing connection (wait option in inetd.conf),
and therefore they may bypass TCP wrappers access control features.
Comsat is an example of such behavior.

Let us illustrate a possible use of TCP wrappers to protect the finger
service. The finger daemon (usually called fingerd or in.fingerd)
listens for TCP connections on port 79 and displays information about
the users of the system to the clients. Usually, full user names,
home directories, login shells, time of last login (or current logins)
are displayed as well as the contents of .plan and .project files if
they are present in user's home directory. Some administartors may
feel that this information is security sensitive and should not be
provided without restrictions. Customarily, fingerd is started by
the inetd daemon. The inetd configuration file /etc/inetd.conf may
contain a line similar to the following:

finger   stream   tcp   nowait   root   /usr/sbin/fingerd   fingerd

The sixth field in this line specifies that once the inetd daemon
accepts a network connection on port 79 it hands it over to fingerd
for processing. In order for TCP wrappers to act as an intermediate
program we should replace this line with the following:

finger   stream   tcp   nowait   root   /usr/local/sbin/tcpd   fingerd

Here /usr/local/sbin/tcpd is the location of the main TCP wrappers
program, tcpd. This location may vary from system to system. After
that inetd must be restarted (or sent the HUP signal to reread its
configuration file /etc/inetd.conf). The access control rules need
to be specified in /etc/hosts.allow and /etc/hosts.deny. Suppose
that we only want to allow finger requests from our internal subnet
10.2.3.0 with mask 255.255.255.0 and deny all others. Moreover, we
would like to be notified when an unauthorized fingerd connection
is attempted. The following lines in access control files will
provide for this:

In /etc/hosts.allow (ALLOW is redundant and can be omitted):

fingerd: 10.2.3.0/255.255.255.0 : ALLOW

And in /etc/hosts.deny (DENY is redundant and can be omitted):

```
fingerd: ALL: DENY: spawn (echo "Finger probe from %h" | mail root) &
```

Now when inetd accepts a TCP connection on port 79, it will run tcpd (the TCP wrappers program) instead of fingerd. The tcpd program will check the source address of the incoming connection against the line above from /etc/hosts.allow and if it matched, start fingerd. In this case, tcpd is transparent from the client's point of view. However, if the client is not in the authorized address range, the connection will dropped and a warning message with the client's address will be sent to the administrator of the host according to the corresponding line from /etc/hosts.deny.

Another case when /etc/hosts.allow and /etc/hosts.deny files can be used for access control, is with libwrap (TCP wrappers library) enabled network services such as OpenSSH (it is a compile time option for OpenSSH). For example, suppose that we want to only allow ssh connections from adminbox.someschool.edu and deny all others. Then analogously to the previous example the following lines in /etc/hosts.allow and /etc/hosts.deny will do the job:

In /etc/hosts.allow (ALLOW is redundant and can be omitted):

```
sshd: adminbox.someschool.edu : ALLOW
```

And in /etc/hosts.deny (DENY is redundant and can be omitted):

```
sshd: ALL: DENY: spawn (echo "Ssh connection from %h" | mail root) &
```

For more information and exact configuration options we recommend reading the README file that comes with the TCP wrappers distribution and the corresponding man pages hosts_access(3), hosts_access(5), hosts_options(5), and tcpd(8).

The TCP wrappers package also comes with useful utilities, tcpdchk(8) and tcpdmatch(8). These allow checking tcpd configuration and testing it before activating TCP wrappers.

                    Recommendations for improving security (recap)

In this section we repeat the recommendations given in the previous sections for the ease of reference. The order is linear.
In the following section we will present a brief overview of security in general followed by a prioritized list of steps to improve security

of the system.

Operating system:

    a) download and apply the latest patch kit from the vendor.

Apache:

    a) upgrade to the latest stable Apache version (1.3.20),
       install only standard (necessary) modules;

    b) configure access denied by default and specifically enable
       it on the basis of need;

    c) consider either restricting CGI scripts to a directory
       controlled by the sysadmin or using suEXEC to run scripts
       with the actual user identity;

    d) change permissions on the ScriptAlias directory to a safer
       755, make files owned by root and also change permissions
       to 755;

    e) clean up the CGI scripts in the ScriptAlias directory,
       remove all but necessary scripts.

Sendmail:

    a) update /etc/mail/sendmail.cw with the most current
       hostnames of null-clients on the local subnet;

    b) remove obsolete (BITNET_RELAY) and unnecessary
          (/etc/mail/relay-domain) configuration options;

    c) correct file access permissions for several users mailboxes
       including root.

Inetd:

    a) enable access control and logging using TCP wrappers
       for telnet, ftp, pop3, imap, and tftp services;

    b) disable finger, comsat, ntalk, dtspc, and rpc.cmsd services
       by commenting out the corresponding entries in
       /etc/inetd.conf.

Security-related:

    a) upgrade the software (OpenSSL, OpenSSH, TCP wrappers) to
       the latest stable versions (OpenSSL-0.9.6b, OpenSSH-2.9,
       TCPwrappers-7.6);

    b) keep the syslogd logs on the host for a month as opposed to
       only a week (this requires editing the relevant parts of
       the root's crontab file), an even better approach is
       setting up a secure log host on the local network that will
       collect and automatically process log messages from
       selected hosts on the local subnet;

    c) configure and install the lsof utility along with other
       system administration programs;

    d) create and publish secure password choosing guidelines for
       local users, since switching to the shadow password file
       based scheme is not practical for the site.

X window system:

    a) rearrange X window and CDE configuration files according to
       the vendor recommendations: keep the default configuration
       files in /usr/dt/config/ and all the locally customized
       files in /etc/dt/config/;

    b) restrict access to the XDMCP port to only the local hosts
       using the Xaccess file as recommended by the vendor.

NFS and NIS (aka Yellow Pages):

    a) export applications directories (/usr/local and /x) read
       only;

    b) disable root access to exported systems from remote hosts,
       remove the '-root' option from /etc/exports;

    c) remove lines from /etc/exports that refer to the hosts that
       no longer exist on the network;

Filesystem:

    a) remove set user id and set group id access permissions from
       files and directories that are not needed for normal
       functioning of the system;

b) create a backup solution for system files and user files.

Printing services:

    a) update the contents of the access files, /etc/hosts.equiv
       and /etc/hosts.lpd, eliminate stale entries that correspond
       to the hosts that either do not exist or no longer use
       the lpd service;

    b) consider removing the legacy aarpd daemon from the system
       or modifying its source to restrict access to only
       the hosts from the local net.

What can be improved?

What is security in terms of the local site? We have a system and its
users. The users have their files on the system and they use services
provided by the system. The system itself is maintained by a group of
people, system administrators. System administrators should be able to
change the system configuration if necessary, regular users shouldn't.
Unauthorized users should not be able to access the system at all.

Below we present six steps in descending priority order that in our
view will help to improve security of the host.

The role of a security administrator is to make sure that the system
maintains these properties. In order to decide what is and what isn't
an authorized access to the system, the security administrator needs
a policy from the authoritative source for the local site. In other
words, the dean or director of the department needs to authorize usage
guidelines or policies for the system. Only after that the current
state of the system can be compared to the "ideal" state described
in the policy statement. Without a policy statement the administrator
can only use his common sense (which may be quite good, actually) in
his decisions. This brings us to the first necessary step:

    Step 1: Creating a site policy.

    A site policy specifies how user accounts are managed, what
    services the system provides to its users, who is responsible
    for different aspects of system maintenance. The system usage
    guidelines need to be worked out and presented to all users
    of the system. Users must adhere to those guidelines.

The site policy and usage guidelines may be fairly general but these documents will help to clarify resposibilities of people who are involved in maintaining and using the system.

In our view, the most important part of keeping our host secure is maintaining current system software with all known patches applied. Therefore, the next step is

Step 2: Updating system software.

This involves updating the OS with the patchkit freely available from the vendor and updating added software (Apache, Sendmail, Openssh, Openssl, TCP Wrappers).

In the previous sections of this report we outlined problems with versions of this software that are installed on the system currently. Of course, after updating the software all configuration files have to be revied according to our recommendations on the previous sections. Obtaining the patchkit from the vendor and the latest versions of software from the corresponding project sites is usually a straightforward procedure (use anonymous ftp).

Unfortunately, software are not perfect and sometimes default system configuration may be to permissive. Especially in the case of SUID and SGID files, one can not be too careful. The SUID and SGID tricks were designed to (temporarily) grant superior priveleges to regular users. Having too many SUID root programs in the system increases chances that a local attacker can take advantage of a bug or a feature in an SUID root program and elevate his privileges. So,

Step 3: Review our list of SUID and SGID programs and remove unneeded SUID/SGID bits. Sometimes the whole package that contains unnecessary SUID/SGID binaries can be removed.

We provided a complete list of SUID/SGID programs on the system in one of the previous sections of these report together with descriptions of these programs.

Unauthorized users should be able to access system pretending to be authorized users. Therefore, any network services have to be configured to provide only necessary access and nothing else.

Step 4: Reconfigure network services (NIS, NFS, inetd, and others that are described in our report) to prevent unauthorized access to the system.

Authorized users accounts are protected by passwords. It is important
that users are aware of security issues.

   Step 4: Create security recommendations for users. They can
   include password choice guidelines, web security/privacy
   issues, e-mail (virus) information. Also, promote encrypted
   communication programs (SSH, SSL) over unecrypted (telnet,
   ftp, etc).

The steps described so far represent a passive approach to security
and fall under the category of prevention.

Bruce Schneier in his latest book on computer security presents 3
aspects of security: prevention, detection, and response to security
incedents. Traditionally, many security recommendations focus on
prevention almost exclisively. However, as practice shows detection
and response play a big role in minimizing the damage incurred by
secuirty incidents. Our recommendation in light of this is

   Step 5: Create a secure log host on the local network and copy
   all system logs to this log host. There are several log
   checking programs that can sieve through logs for signs of
   suspicious activity, and many administrators tend to come up
   with their own log analizing tools.

Successful attackers are known to go after logs first and delete
the evidence of break-ins. Storing logs on a secure host can help
in reconstructing the break in and will aid the recovery process.

As far as responding to security incidents, educational institutions
are traditionaly very laid back and will rather tolerate occasional
abuse than introduce strict security rules and spend effort on
pursuing attackers. Therefore, system administrators need to be able
to restore their systems to the known good state if a break-in
happened.

   Step 6: Implement a reliable backup scheme for system files
   and user data files.

Periodically backed up snapshots of the system can also be helpful
for users who may accidentally delete files that they will need later,
this happens often with their e-mail folders when a misconfigured
e-mail client removes e-mail from the server.

We believe that implementing these 6 steps will significantly increase

robustness and security of our host. At the same time, none of these steps reduces functionality or inconveniences the users, so the system can continue funcioning as before from the users point of view.

References

Technical references:

1. Apache 1.2 User Guide, as distributed electronically with Apache web server.

   Apache web server distributes with a manual that describes configuration options, programming interfaces, and gives security recommendations. For any administrator who sets up apache web server we recommend reading at least through this manual in order to get familiar with potential security issues.

   Apache web server web site: http://www.apache.org.

2. Sendmail README, by Eric Allman, distributed as cf/README with sendmail-8.9.3.

   This quite long document contains explanation of numerous sendmail configuration options and it also discusses their security implications. Newer releases of sendmail have even more information in this file, one has to start with the README file in the sendmail source directory, this file contains pointers for further reading.

   Sendmail web site: http://www.sendmail.org.

3. Unix System Administration Handbook, Third Edition, by Evi Nemeth, Garth Snyder, Scott Seebass, Trent R.Hein.

   This is an excellent book for all Unix administrators. It covers several popular Unix flavors and the majority of Unix administration tasks. It not only contains a thorough coverage of most aspects of Unix administration but also discusses security issues associated with them. Strongly recommended.

4. The Washington University IMAP server distribution documentation.

   There is a number of documents in the docs/ directory in the source directory of the WU imapd distribution. These documents explain how to install, configure, and enable SSL (encryption) for imapd.

Recommended read for imap administrators.

WU imapd web site: http://www.washington.edu/imap/

5. Tru64 release 4.0F system man pages, online documents at
   http://tru64unix.compaq.com/faqs/publications/pub_page/V40F_DOCS.HTM

Unfortunately, very few books dedicated specifically to Tru64 Unix
have been published (search on Amazon.com results in just 2 books
and none of them is on the topic of security). The vendor's site,
however contains a lot of useful information including all manual
pages, and User's, Administrator's, and Programmer's guides
collections. We recommend to any Tru64 following the link above
for these resources.

General security references:

1. Secrets and Lies, by Bruce Schneier, published by John Wiley &
   Sons, Inc, 2000.

Bruce Schneier is the author of Applied Cryptography, a well known
book in that area. In his latest book, Secrets and Lies, he
presents his views on computer security in a way accessible to
general public not only to the specialists. He discusses
the landscape of digital security (what th threats are and what we
need to deal with them), technologies and their limitations, and
finally the strategies for digital security (what to do given the
requirements and available technologies). This is an excellent
read and we highly recommend it to all system administrators even
if their job title does not have the word 'security' in it.
[The author of this report read this book several times and every
time he learned something new that he missed at the previous
reading.]

2. http://www.securityfocus.com

Securityfocus.com is an excellent resource for all system
administrators. The site contains a wealth of information about
computer security with large sections dedicated to Microsoft,
Sun, and Linux specific issues. This site also provide convenient
browsing and searching interface to the famous Bugtraq mailing
list where security related information about many systems is
disclosed and discussed. The papers library section of this site
contains a large variety of security procedures and practices
guides as well as research papers. We recommend visiting this site
and using its resources to all system administrators.

Afterword

It is probably impossible to create security guidelines that will
equally well serve for all kinds of systems. We wrote this report
with a specific academic system in mind. Our analysis and
recommendations may be suitable for other academic sites as well.
In our view, academic institutions benefit from open systems with
minimal necessary restrictions on users as that promotes cooperation
and exchange of ideas. Often the data do not have a large monetary
value (privacy however is always important!) and academics tend to
emphasize openness and interoperation with others. We tried to keep
these goals in mind when writing this report. Security can not be
ignored even in an educational institution and cases when attackers
used compromized machines in universities to launch attacks on other
sites are numerous. Therefore, we tried to present security as a means
for achieving academic goals by providing functional, robust and
secure computing environment.

However, we understand that a military site or a financial institution
would require a different approach and different goals. Mandatory
access control and audit tracks may be required and a different
audit and accountability policy may have to be implemented. Official
security guidelines (rainbow book series) may have to be used in
order to create and maintain a compliant secure environment. This was
outside of the scope of this report and we have not touched on that
here.