



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Securing Linux/Unix (Security 506)"
at <http://www.giac.org/registration/gcux>

Security Analysis Report of the ALS Linux Development Server

Report Introduction

The following report is a description of the security stance of a server running the Red Hat distribution of the Linux operating system within the ALS Company*. This report is organized into the following sections:

- I. A short introductory section giving historical background of the server
 - II. An executive report summary
 - III. A detailed findings section, including explanations on why a condition is insecure
 - IV. A prioritized summary of vulnerabilities
 - V. Prioritized tiered recommendations for resolving the vulnerabilities
 - VI. Appendices with listing and explanatory notes
 - VII. Listing of Internet based information security resources
 - VIII. References
-

I. History

The Linux server under study was originally created for experimentation within the software development division of the ALS Company. The ALS server, which runs the Linux operating system, is a Pentium II computer with 128 megabytes of memory and 8 gigabytes of hard disk space. The server has been operational for 13 months and originally was installed with Red Hat Linux version 5.2, but within the past six months has been upgraded to version 6.0. Because the corporate local area network is Windows NT based, the network administrators have very limited expertise with Linux/Unix boxes. Consequently, the administration of the ALS server has always been the responsibility of the development department.

Experimentation with cross platform development was the servers original charge, but within a short period of time the server was performing critical tasks supporting a distributed database application and daily processing of data files. This transition from an experimental, non-essential server to an essential production machine has put pressure on the server's administrator and the management of the development department to tighten up server operations, procedures and security.

II. Executive Summary

*The ALS Company is a pseudonym for a medium size industrial company that employs over 300 people.

The security analysis of the ALS server revealed numerous and dangerous security problems. When this server was being used for experimental purposes, the need for rigorous administration and strong security was negligible. However, now that the ALS server has moved beyond the experimental stage and is being used for critical information service tasks, the above attitude needs to change. Indeed, if the intention is to keep the server in a production mode and not as an experimental machine, than a rethinking and overhaul of the server's administration and security posture is in order.

The highlights of the analysis of the ALS server have been broken down into the following three broad categories:

Server Administration and Physical Security

1. There are no company wide security or acceptable use policy statements.
2. The physical location of the server is in an unprotected and vulnerable room where accidental or malicious events could damage or incapacitate the server.
3. The administration of the server is done on a part-time basis by a software developer within the company. Not surprisingly, the ALS server administration gets short attention.
4. The server is running Red Hat version 6.0 and has not been patched with newer software nor upgraded to the new version 6.2.
5. System backups are not done according to any set schedule. Tape media is not protected from loss or damage and there is no system for offsite tape storage.
6. No contingency plans are in place in case a malicious or natural disaster incident prevents the server from performing normally.

Server Configuration and Networking

1. The server has machine and operating configuration settings that need to be corrected to provide protection against malicious tampering.
2. There are numerous services that are being provided by the server that are unnecessary. These services need to be turned off.
3. The server does not filter connections made by inside or outside clients. There is no attempt at restricting who is allowed to connect to the server and from where.

Software

1. The Apache Web Server and the Samba (Microsoft file and print sharing) software packages are currently installed and running on the server. These need to be removed.
2. MySQL, a database server, is being used by in-house client application. Connections to this database via the ALS server are not controlled.

In summary, the ALS server has graduated from a “geeks” experimental machine to one that provides essential data processing needs. It now needs to be managed, maintained and secured in a manner that reflects this upgrade in system responsibilities.

III. Report Findings

Introduction

The security analysis of the ALS server revealed numerous security vulnerabilities and problems. Factors ranging from the physical security of the company building, to configuration files on the ALS server were analyzed. The analysis has been broken down into the following seven subject areas:

- Security and Acceptable Use Policies
- Physical Security
- Server Administration
- Server Configuration
- Software Concerns
- Networking
- Backup and Disaster Contingency Planning

Under each of these categories, found vulnerabilities and problems will be discussed. If appropriate, other information will be presented as either background or explanation. There will also be explanations on why a problem *is* a problem and what measures should be taken for correction.

Security and Acceptable Use Policies

1. A company wide computer security policy has not been developed.¹ An acceptable use policy is in force, however this policy statement is poor and is currently being revised.
 - The lack of good security and acceptable use policies leaves the company vulnerable on two fronts. First, the lack of established rules and guidelines prevents system users from knowing what are and what are not permitted activities. Secondly, if security is breached or other malicious damage is done to the system, the lack of a documented and known security policy may make it difficult to pursue legal action against the perpetrators.

Physical Security

1. Building Security
 - Entry into the main company building during business hours is via keypad. After hours, the entry keypads no longer function and entry is processed by security staff. All doors that enter the building are monitored with close-circuit television cameras 24 hours a day.
 - During business hours visitors must be signed-in at the main reception area and should be accompanied by a company employee when inside the building.

2. Software Development Department

- The software development department, where the ALS server is located, is a lightly secured, large room having two medium weight doors. The door locks are double bolted and the hinges of both doors are inside the room. During business hours both doors are unlocked with no access restrictions. After hours access requires processing through building security and unlocking the department doors.
- False ceilings extend beyond the walls of the room.
- A basic ABC type fire extinguisher is available in the room, but no other fire fighting devices are available.

The security to enter the main ALS Company building is adequate in contrast to the lax security conditions of the development department. If unauthorized people have access to the ALS server, they could carry out malicious attacks against the system. Further, and perhaps more of a risk, is the potential for accidental incidents that could damage the server. If at all possible, the ALS server should be housed in a locked air-conditioned room, with only authorized staff having access to the room. The room should also have walls that extend all the way to the roof-
-false ceilings are easy to climb through. Installation of a Halon fire retardant system would be a bonus.

Server Administration

1. System administration of the ALS server is carried out by one of the software developers on a part-time basis.
 - This condition is dangerous for a number of reasons. First, if the sysadmin decides to leave the ALS Company and no one knows the setup, configuration and other particulars of the ALS installation (“Who knows the root password?”) managing the server will be very difficult. Second, having an employee, especially a software developer, work part-time on administering a Linux/Unix system is placing too much responsibility on that individual to keep up with patches, users and other administration chores. Solution? Hire a well-trained system administrator or if that is not possible, make sure that the part-time sysadmin has dedicated time to administer the ALS server.
2. The current sysadmin logs into the ALS server as *root* for administrative and non-administrative tasks. (see Appendix A, Listing 1)
 - This practice is dangerous because when the sysadmin is logged in as *root* he/she could accidentally cause significant damage. For instance, if as *root* the sysadmin issued the command `rm -rf *` and was accidentally at the `/usr` directory...instant chaos ensues. Also, it is impossible to track *who* logged into the server as *root*. It is essential to login as *root* only for administrative purposes and to gain *root* access with the `su` command.

The `su` (substitute user) command logs who was successfully able to become another user, including `root`.

3. Server administration is normally performed at the server console or via a telnet connection.
 - For the reasons stated above, console access should be restricted as much as possible. Locating the ALS server out of the development department would inhibit this habit.
 - It is extremely dangerous to `telnet` into the server from a workstation and then gain `root` using `su` because the `root` password is being sent in the clear over the network. This practice should be stopped immediately. If `telnet` access and remote system administration is desired, install `ssh`. (see Appendix C)
4. Regular server log inspections are not performed and logging configurations need to be improved.²
 - Logging is essential in monitoring a Linux/Unix server. The logs should be inspected on a regular basis by either inspecting the logs themselves or with a log inspection tool. (see Appendix C)
5. There are a number of system accounts that are not needed on the server and should be removed (see Appendix A, Listing 2)
6. System vulnerability analysis and file integrity tools are not used on the ALS server.
 - There are tools that help the sysadmin to spot potential security problems with system and service configurations and network access. Other tools are also available to check the integrity (and possible compromise) of files on your system. Install these tools and use them. (See Appendix C)
7. The machine is configured and run as both a server and a workstation. A server is a server and a workstation is a workstation!
 - Remove all workstation related software. (see Appendix A, Listing 5)

Server Configuration

1. No password authentication was found for the computer system BIOS and the ability to boot the machine from various media had not been disabled.
 - This condition is dangerous, because if someone can reboot the machine and access the BIOS, they can boot the machine with removable media (floppy/CDROM) that may contain “rootkit” executables or hostile Linux boot images. Denial-of-service attacks are also possible.
2. On the ALS system, it is possible to reboot the system using the `Ctrl-Alt-Delete` key combination and not require the root password to enter single user mode.
 - This “feature” can lead to an unauthorized user rebooting into single user mode (`LILLO: linux single`)--which by definition is root. Disable the `Ctrl-Alt-Delete` key sequence and require the root password for single user mode by editing the `/etc/inittab` configuration file. (see Appendix B, Example 1 & 2)
3. The ALS server runs numerous services including `inetd` services, `httpd`, `sendmail`, `mysqld` the MySQL database daemon. (see Appendix A, Listing 3 & 4)
 - There are many services that are running on the ALS server that do not need to be active and should be turned off. These “extra” services were installed at the time the Red Hat distribution was installed. Turning off services that are not needed helps in managing and updating the server, server performance and preventing attacks that are specific to individual services. (see Appendix B, Example 3)
 - All `inetd` services* need to be commented out of `/etc/inetd.conf` and the `inetd` daemon needs to be turned off. The `linuxconf` service that is spawned by `inetd` is for a graphical system configuration tool. The utility in using `linuxconf` is not great enough to warrant leaving `inetd` active.
4. The ALS server on occasion is running X Windows.
 - There is no reason for the ALS server to be running X Windows. This entire package should be disabled and removed.
5. There are many installed software packages on the ALS server that can be removed.

* If in the future it is determined that an `inetd` service is needed, make sure that `/etc/hosts.allow` and `/etc/hosts.deny` are not empty. Red Hat by default runs all `inetd` services via `tcp_wrappers`, but ships with `hosts.deny` and `hosts.allow` empty. When creating access permissions, begin by specifying `ALL:ALL` in `/etc/hosts.deny` and then add specific hosts in `/etc/hosts.allow`.

- These programs are unnecessary for the operation of the server and could harbor hidden security risks. (see Appendix A, Listing 5)

Software Concerns

1. The Apache Web Server, the MySQL database system and Samba the smb server are all running on the ALS.
 - Since the ALS server is not a web server there is no need to have Apache running. It should be removed.
 - MySQL is needed to provide database services for client applications. MySQL has a host/user based security architecture that can provide database connections to clients or users that do not have accounts on the server. However, they still need an account on the MySQL database server. This may or may not be an issue with some installations, but it is advisable to only allow connection permissions to a defined set of hosts using `ipchains`. (see Appendix C) MySQL also has had a few security vulnerabilities that can compromise database integrity and availability. Update to the latest version and apply any needed patches. (see <http://www.ciac.org/ciac/bulletins/k-025.shtml>)
 - Mistakenly, the Samba daemons `smbd` and `nmbd` were thought to be required for server applications needing connectivity with the corporate Microsoft LAN. These daemons are *not* needed to access files and directories on the network. The `smbmount` application that comes with the Samba installation provides shared directory access services. Indeed, the ALS server uses a couple of `smbmount` commands to provide those mounts! The Samba daemons should be disabled. (see Appendix B, Example 4)
2. There are a few in-house programs (see Appendix A, Listing 6) written in the Python programming language that are owned by `user:root group:root`.
 - These programs need to have their ownership changed and run under another user account. Interpreted programming languages like Python, perl or shell scripts that are owned and run by `root` should be scrutinized carefully for problems. They should never be allowed to have the SUID bit set. There are a myriad of ways an attacker can gain `root` from poorly written scripts and file race conditions. Thankfully, Red Hat Linux will ignore shell scripts with the SUID bit set. A good rule of thumb to follow is to be extremely cautious in setting the SUID bit on any file, make sure it is *absolutely* necessary.

Networking

1. The ALS server has an Ethernet connection to the corporate LAN. This connection is needed for the development staff to upload Java applications for cross-platform testing.

- Currently the ALS system does not discriminate where a connection comes from on the local LAN. A judicious use of `ipchains`, may prevent unwelcome in-house packets. (see Appendix C)
2. On a regular basis, the ALS server connects to the Internet for short periods of time (3-5) minutes via an ISDN connection.
 - Although the Internet connections are short, like the previous comment, `ipchains` would help prevent unwanted visitors from trying to access the ALS server.
 3. The server is not using NFS, NIS or BIND. Care should be used if these services are provided in the future.

Backups and Disaster Contingency Planning

1. The ALS server's backup tape mechanism is a DAT drive with a 2 gigabyte capacity. Backups are haphazard and are not done according to any established schedule. There is also no offsite storage or accounting of backup media. Only cursory restore tests have been performed.
 - The neglect of making routine, scheduled backups is a grave threat to the security of the data on the system. Further, the lack of backups could endanger the ability to investigate system security breaches. Backups are an essential and basic element of information security and MUST be performed.
2. There are no contingency plans if a man-made or natural disaster should occur.
 - Planning for untoward events enhances the security of the ALS Server by insuring that critical information will be available as soon as possible after a disastrous event.

IV. Vulnerability Listing

The listing below is a prioritized recap of the issues discussed in the previous section.

1. Poor administration of the server.
2. Physical location of the server is in an unsecured area.
3. No scheduled backups.
4. Easy access to the system console.
5. Unneeded services running (see Appendix A, Listing 3 & 4).
6. No company wide security or acceptable use policies.
7. No filtering of network connections to the server.
8. The use of outdated operating system software.
9. The ability to reboot the server with the `Ctrl-Alt-Delete` key sequence.
10. Not keeping up with software patches.
11. The ability to boot into single user mode without the `root` password.
12. In-house programs being run as `root`
13. The `sysadmin` logs in as `root` for most tasks.
14. No contingency plans if a disaster occurs.
15. No offsite or secured storage of system backup media.
16. Original installation of operating system, occurred with the server connected to the corporate LAN.
17. No monitoring of system logs.
18. Lack of focused system administration.
19. X windows can be run on the server.
20. Apache Web Server and Samba software packages running when they are not needed.
21. No file integrity checking software, such as `tripwire`, is being used.
22. Network and system scanning tools are not used to determine system vulnerabilities.
23. Many unused built-in system accounts remain on the system.
24. Doing development testing on a production server.
25. System is being used as both a workstation and server.

IV. Tiered Recommendation List

The ease and low cost with which a Linux/Unix server can be put together is one of the main reasons for the popularity of the operating system. Unfortunately, this ease makes it extremely easy to overload the server with programs and capabilities that the machine does not need to function as a server. An over arching rule that should be remembered concerning the ALS server is:

The ALS server should be used as a SERVER and not a workstation.

Keeping this in mind, the following prioritized recommendations have been organized into two tiers. Tier One recommendations are the most basic and are required for a bare minimum of security. These recommendations are relatively inexpensive and should be within the ability of the ALS Company to fund. The recommendations that makeup the Tier Two set are more cost intensive and drastic--but they can afford a high level of security and ensure the continued reliable use of the ALS server system. It should be noted that all of the Tier One recommendations should be incorporated within Tier Two.

At a minimum, the Tier One recommendation set should be implemented.

Tier One

1. Backup appropriate data files and rebuild the server. Install Red Hat Linux version 6.2 and be sure to install the latest patches from the Red Hat web site: (<http://www.redhat.com>). The server should be *disconnected* from the corporate LAN and the Internet during reinstallation.³
2. The ALS server should be relocated in a secured area where only access by the authorized staff is permitted.
3. If not done so already, disable all services within `inetd.conf` and turn off the `inetd` daemon.
4. Remove ALL unnecessary packages.
5. `tripwire` should be installed soon after the operating system has been installed.
6. A regular and comprehensive backup schedule needs to be implemented.
7. Tape media needs to be stored in a secure area and offsite storage should be considered.
8. Install `ssh` (Secure Shell) for remote administration and user access and remove the BSD `r` commands. (see Appendix C)
9. Insure that the `sysadmin` has dedicated time to administer the server.
10. Run AutoRPM to automatically update system. (see Appendix C)
11. Create `ipchains` rule sets for Internet and in-house server access.
12. Develop a security policy including acceptable use policies.
13. A “backup” system administrator needs to be commissioned.
14. Although not identified as a problem, make sure that users are using good passwords.
15. Login banner warnings should be written for `/etc/motd` and `tcp_wrappers`.
16. Subscribe to security mailing lists. (see Appendix D)

Tier Two

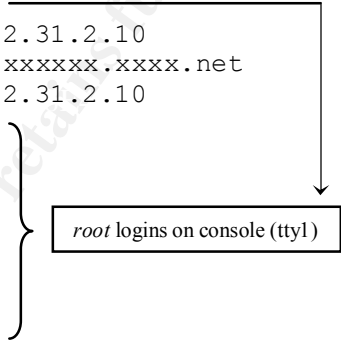
1. Hire a fulltime Linux/Unix system administrator.
2. Install four more servers for a total of five. These servers would be used for:
 - Client database access needs
 - Batch data processing
 - Software development and testing
 - A central logging server to handle logging for the other four machines
 - Proxy services for access to the Internet
3. Install a dedicated firewall for Internet access.
4. Install a Halon fire retardant system to protect the servers.
5. Use the Network Time Protocol to coordinate time among the servers.
6. Install RAID disk arrays on all servers.

© SANS Institute 2000 - 2002, Author retains full rights.

Appendix A: Listings From the ALS Server

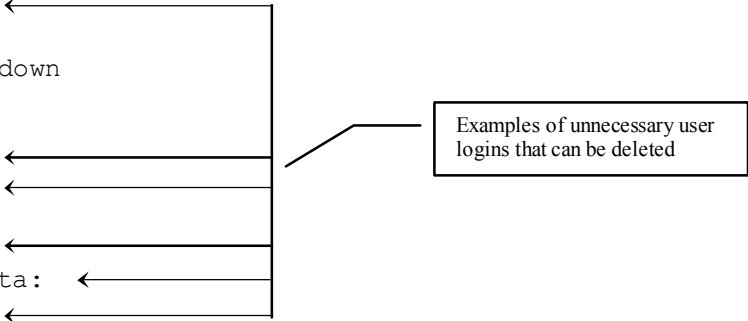
Listing 1: A section of the `/var/log/secure` log file showing `root` logins from the console (tty1).

```
.
.
Jul 18 13:30:44 ALS in.telnetd[652]: connect from 172.31.2.10
Jul 18 13:30:54 ALS login: LOGIN ON 0 BY xxxxxx FROM 172.31.2.10
Jul 18 13:31:18 ALS login: LOGIN ON tty1 BY xxxxxx
Jul 18 13:35:16 ALS in.rlogind[724]: connect from 172.31.2.10
Jul 18 13:35:25 ALS login: LOGIN ON 0 BY xxxxxx FROM 172.31.2.10
Jul 18 14:08:15 ALS login: ROOT LOGIN ON tty1
Jul 18 14:09:42 ALS in.rlogind[782]: connect from 172.31.2.10
Jul 18 14:10:07 ALS login: LOGIN ON 0 BY xxxxxx FROM xxxxxx.xxxx.net
Jul 18 14:10:25 ALS in.rlogind[802]: connect from 172.31.2.10
Jul 18 16:39:22 ALS login: ROOT LOGIN ON tty1
Jul 19 16:22:50 ALS login: ROOT LOGIN ON tty1
Jul 20 13:25:59 ALS login: ROOT LOGIN ON tty1
Jul 20 13:30:07 ALS login: LOGIN ON tty1 BY xxxxxx
Jul 20 13:30:26 ALS login: ROOT LOGIN ON tty1
Jul 20 14:37:26 ALS login: ROOT LOGIN ON tty1
Jul 21 11:14:31 ALS login: ROOT LOGIN ON tty1
Jul 21 14:57:25 ALS login: ROOT LOGIN ON tty1
.
.
```



Listing 2: A portion of the `/etc/passwd` file showing examples of extra user logins that can be eliminated.

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
adm:x:3:4:adm:/var/adm:
lp:x:4:7:lp:/var/spool/lpd:
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:
operator:x:11:0:operator:/root:
games:x:12:100:games:/usr/games:
gopher:x:13:30:gopher:/usr/lib/gopher-data:
ftp:x:14:50:FTP User:/home/ftp:
nobody:x:99:99:Nobody:/:
xfs:x:100:101:X Font Server:/etc/X11/fs:/bin/false
gdm:x:42:42:./home/gdm:/bin/bash
xxx:x:500:500:xxx:/home/xxx:/bin/bash
xxxmanftp:x:501:501:xxx:/home/kermanftp:/bin/bash
xxxlan:x:502:502:xxx:/home/xxxlan:/bin/bash
xxxner:x:503:502:xxx:/home/xxxner:/bin/bash
xxxton:x:515:502:xxx:/home/xxxton:/bin/bash
xxxmad:x:516:502:xxx:/home/xxxmad:/bin/bash
.
.
```



Listing 3: Using the `lsof` (List Open Files) command shows a listing of open files for various processes. `lsof` comes with the Red Hat distribution and is located in: `/usr/sbin`.

Command: `lsof -i`

COMMAND	PID	USER	FD	TYPE	DEVICE	SIZE	NODE	NAME
portmap	288	root	4u	IPv4	289		UDP	*:sunrpc [portmapper]
portmap	288	root	5u	IPv4	290		TCP	*:sunrpc [portmapper] (LISTEN)
inetd	406	root	5u	IPv4	404		TCP	*:ftp (LISTEN)
inetd	406	root	6u	IPv4	405		TCP	*:shell (LISTEN)
inetd	406	root	7u	IPv4	406		TCP	*:login (LISTEN)
inetd	406	root	9u	IPv4	407		UDP	*:talk
inetd	406	root	10u	IPv4	408		UDP	*:ntalk
inetd	406	root	11u	IPv4	409		TCP	*:finger (LISTEN)
inetd	406	root	12u	IPv4	410		TCP	*:auth (LISTEN)
inetd	406	root	13u	IPv4	411		TCP	*:linuxconf (LISTEN)
inetd	406	root	14u	IPv4	737		TCP	*:telnet (LISTEN)
lpd	422	root	6u	IPv4	431		TCP	*:printer (LISTEN)
httpd	461	root	16u	IPv4	468		TCP	*:www (LISTEN)
httpd	465	root	16u	IPv4	468		TCP	*:www (LISTEN)
httpd	466	root	16u	IPv4	468		TCP	*:www (LISTEN)
httpd	467	root	16u	IPv4	468		TCP	*:www (LISTEN)
httpd	468	root	16u	IPv4	468		TCP	*:www (LISTEN)
httpd	469	root	16u	IPv4	468		TCP	*:www (LISTEN)
httpd	470	root	16u	IPv4	468		TCP	*:www (LISTEN)
httpd	471	root	16u	IPv4	468		TCP	*:www (LISTEN)
httpd	472	root	16u	IPv4	468		TCP	*:www (LISTEN)
httpd	473	root	16u	IPv4	468		TCP	*:www (LISTEN)
httpd	474	root	16u	IPv4	468		TCP	*:www (LISTEN)
mysqld	507	root	3u	IPv4	502		TCP	*:mysql (LISTEN)
mysqld	512	root	3u	IPv4	502		TCP	*:mysql (LISTEN)
mysqld	513	root	3u	IPv4	502		TCP	*:mysql (LISTEN)
smbd	1652	root	6u	IPv4	270277		TCP	*:netbios-ssn (LISTEN)
nmbd	1663	root	6u	IPv4	270285		UDP	*:netbios-ns
nmbd	1663	root	7u	IPv4	270287		UDP	*:netbios-dgm
nmbd	1663	root	9u	IPv4	270290		UDP	xxx.xxx.net:netbios-ns
nmbd	1663	root	10u	IPv4	270292		UDP	xxx.xxx.net:netbios-dgm
mount.smb	564	root	3u	IPv4	541		TCP	ALS.xxx.net:1024->xxx.xxx.net:netbios-ssn (CLOSE)
mount.smb	566	root	3u	IPv4	542		TCP	ALS.xxx.net:1025->xxx.xxx.net:netbios-ssn (CLOSE)
X	594	root	0u	IPv4	576		TCP	*:6000 (LISTEN)

inetd services that can be disabled and then turn `inetd` off and replace with SSH.

Apache httpd server should be disabled and the installation removed.

Samba daemons

smb (Samba) mounts

Listing 4: Similar to the lsof listing above, the netstat command displays process and port information. These particular parameters show TCP ports. Command: netstat -atp

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 *:netbios-ssn          *:                       LISTEN      1652/smbd
tcp        0      0 Xxx.xxxx.net:telnet    xx.xxxx.net:1102       ESTABLISHED 679/in.telnetd
tcp       12      0 Xxx.xxxx.net:1025      xxxx.xxxx.:netbios-ssn CLOSE       572/mount.smbfs
tcp       12      0 Xxx.xxxx.net:1024      xxxx.xxxx.n:netbios-ssn CLOSE       574/mount.smbfs
tcp        0      0 *:mysql                *:                       LISTEN      515/mysqld
tcp        0      0 *:www                  *:                       LISTEN      465/httpd
tcp        0      0 *:printer              *:                       LISTEN      426/lpd
tcp        0      0 *:linuxconf            *:                       LISTEN      410/inetd
tcp        0      0 *:auth                 *:                       LISTEN      410/inetd
tcp        0      0 *:finger               *:                       LISTEN      410/inetd
tcp        0      0 *:login                *:                       LISTEN      410/inetd
tcp        0      0 *:shell                *:                       LISTEN      410/inetd
tcp        0      0 *:telnet               *:                       LISTEN      410/inetd
tcp        0      0 *:ftp                  *:                       LISTEN      410/inetd
tcp        0      0 *:sunrpc               *:                       LISTEN      292/portmap
```

inetd and Apache httpd services that can be disabled See Listing 3.

Listing 5: Using the Red Hat Package Manager (rpm) installed packages can easily be determined. Command: rpm -qa

```
aktion-0.3.6-3
anonftp-2.8-1
apache-1.3.9-4
apmd-3.0beta9-3
arpwatch-2.1a4-16
ash-0.2-18
.
.
.xrn-9.01-3
xscreensaver-3.17-4
xsri-1.0-4
xxgdb-1.12-10
ypbind-3.3-24
yp-tools-2.3-2
zip-2.2-1
zlib-1.1.3-5
zlib-devel-1.1.3-5
.
.
```

Examples of package listing. Many of the packages installed on the ALS server can be removed.

Listing 6: Directory listing, showing Python program files which are run as *root*.

```
total 300
drwxr-xr-x  2 root    root    4096 Jun 30 13:09 .
drwxr-xr-x  4 root    root    4096 May 29 13:10 ..
-rwxr-xr-x  1 root    root     362 Jun 30 13:08 MakeDoc.py
-rwxr-xr-x  1 root    root    1414 Jun 30 13:08 bserPrtBill.py
-rwxr-xr-x  1 root    root    1022 Jun 30 13:08 bserPrtBill.py~
-rwxr-xr-x  1 root    root    6699 Jun 30 13:08 bservDecrypt.py
-rwxr-xr-x  1 root    root    6624 Jun 30 13:08 bservDecrypt.py~
-rwxr-xr-x  1 root    root    2643 Jun 30 13:08 bservFilePrep.py
-rwxr-xr-x  1 root    root    3750 Jun 30 16:03 bservFilePrep.pyc
-rwxr-xr-x  1 root    root    2643 Jun 30 13:08 bservFilePrep.py~
-rwxr-xr-x  1 root    root    4499 Jun 30 13:08 bservParse.py
-rwxr-xr-x  1 root    root    5218 Jun 30 13:08 bservParse.py~
-rwxr-xr-x  1 root    root    6022 Jun 30 13:08 bservProcess.py
-rwxr-xr-x  1 root    root    7114 Jun 30 13:08 bservProcess.pyc
-rwxr-xr-x  1 root    root    1433 Jun 30 13:08 bservPrtBill.py
-rwxr-xr-x  1 root    root    1375 Jun 30 13:08 bservPrtBill.py~
-rwxr-xr-x  1 root    root    3993 Jun 30 13:08 bservSQL.py
-rwxr-xr-x  1 root    root    5054 Jun 30 16:03 bservSQL.pyc
-rwxr-xr-x  1 root    root    3965 Jun 30 13:08 bservSQL.py~
-rwxr-xr-x  1 root    root   14500 Jun 30 13:08 ppp.txt
-rwxr-xr-x  1 root    root   31403 Jun 30 13:08 runLog.txt
```

```
.
.
```

© SANS Institute 2000 - 2002, All Rights Reserved.

Appendix B: Procedure Examples

Example 1: Use the following to disable rebooting the machine with `Ctrl-Alt-Delete` key sequences:

1. Find the following line in `/etc/inittab`: `ca::ctrlaltdel:/sbin/shutdown -t3 -r now`
2. Comment it out with a hash mark: `#ca::ctrlaltdel:/sbin/shutdown -t3 -r now`

Example 2: To require the root password to enter into single user mode:

1. Locate the line in `/etc/inittab` that begins with `si::sysint`.
2. Immediately below that entry add: `~~:S:wait:/sbin/sulogin`

Example 3: To disable and remove services follow these steps:

1. Stop a service with the init script: `/etc/rc.d/init.d/httpd stop`
2. Use `chkconfig` to remove the link from the runlevel (`rc.*`) directories: `/sbin/chkconfig httpd off`
3. When all of the desired services are turned off, check there status by running `chkconfig` and `grep`: `chkconfig -list | grep -v on`

Note: `chkconfig` is specific to Red Hat. It is also possible to manually remove the links in each runlevel directory with `rm`.

Example 4: Turning off the Samba daemons with Red Hat is easy: `samba stop`

© SANS Institute 2000 - 2002. Author retains full rights.

Appendix C: Linux/Unix Tools

Most, if not all of the following tools are available free of cost or come with the Red Hat distribution.⁴ An excellent WWW site to find links to many tools is:

<http://www.alw.nih.gov/Security/>

- An excellent file integrity checking tool is `tripwire` by Gene Kim and Gene Spafford. By taking checksumming (there are 8 types available) the contents of a file system and then comparing it with previous checksums, a change in contents can be detected:
Footnote: <ftp://coast.cs.purdue.edu/pub/tools/unix/Tripwire>
- There are a number of network port scanners that analysis sytems for vulnerabilities from a network presepective. The following is a representative example:
 - `nmap` from Fyodor: <http://www.insecure.org>
 - `nessus` by Renaud Dearaison: <http://www.nessus.org>
 - `sara` by Bob Todd (after Dan Farmer & Weitse Venema):
<http://www-arc.com/sara/>
- Tools are available to have the system analyze itself. Examples of which are:
 - `tiger` from Douglas Schales, Dave Hess, Khalid Warraich and Dave Safford:
<ftp://net.tamu.edu/pub/security/TAMU/>
 - `cops` by Dan Farmer: <ftp://ftp.cerias.purdue.edu/pub/tools/unix/scanners/cops/>
- The `ssh` (Secure Shell) suite of programs allows secure logins over network or Internet connections. Replaces the `rlogin`, `rcp` and `rsh` programs with secure alternatives. It can provide strong RSA⁵ based and Kerberos⁶ (plus other types) of authentication, full session encryption and secure port forwarding of X11 and TCP/IP connections.⁷
- `ipchains` is a tool for implementing firewall functionality on a Linux server⁸. Care should be taken in setting up rule sets and a through understanding of firewalls is recommended.
- `tcp_wrappers`, which is automatically deployed with the Red Had distribution, is a software buffer between `inetd` and the services it spawns. It provides access control, logging and other functionality that help make using the various services controlled by `inetd` more secure. Although the ALS server will not need to run any of the `inetd` services, it would be a good idea to become familiar with this tool for possible future use.
- `AutoRPM` by Kirk Bauer is a utility that keep installed RPMs in sync with an FTP site or local directory. This is very useful in keeping the ALS server up-to-date with the latest software patches. <http://www.kaybee.org/~kirk/thml/linux.html>

- A number of good logging tools are available to help analyze and report on logging information. These include:

- `logcheck` by Craig Rowland: <http://www.psionic.com>
- `swatch` by Stephen Hansen and E.Todd Atkins: <ftp://coast.cs.purdue.edu/pub/tools/unix/swatch>

© SANS Institute 2000 - 2002, Author retains full rights.

Appendix D: Internet Based Information Security Resources

<http://www.ieee-security.org/index.html>
<http://www.cerias.purdue.edu/coast/>
<http://packetstorm.securify.com/>
<http://www.sans.org/newlook/home.htm>
<http://www.securityfocus.com>
<http://www.securityfocues.com>
<http://www.isc2.org/>
<http://csrc.nist.gov/>
<http://www.rootshell.org/beta/news.html>
<http://www.2600.com/index.html>
<http://www.attrition.org/>
<http://www.alw.nih.gov/Security/>
<http://www.isse.gmu.edu/~csis/>
<http://www.isaac.cs.berkeley.edu/>

© SANS Institute 2000 - 2002, Author retains full rights.

References

¹ S. Garfinkel, G. Spafford. *Practical Unix & Internet Security*, O'Reilly & Associates, Sebastopol, 2nd Edition, 1996, pp. 35-40

² L. Brotzman, D. Ranch, Editors, *Securing Linux Step-By-Step, Version 1.0*, The SANS Institute, 1999-2000, pp. 13-17

³ Ibid.

⁴ The tools presented (and many others) were described at the SANS DC 2000 Security Conference by Matt Bishop of the University of California, Davis. SANS can be contacted at <http://www.sans.org>.

⁵ B. Schneier. *Applied Cryptography*. John Wiley & Sons, New York, 2nd Edition, 1996, pp. 466-474.

⁶ Ibid., pp. 566-571

⁷ Anonymous, *Maximum Linux Security*. SAMS, Indianapolis, Indiana, 2000, pp. 290-316

⁸ L. Brotzman, D. Ranch, Editors, *Securing Linux Step-By-Step, Version 1.0*, The SANS Institute, 1999-2000, pp. 67-70

© SANS Institute 2000 - 2002, Author retains full rights.

Upcoming Training

Click Here to
{Get CERTIFIED!}



SANSFIRE 2017	Washington, DC	Jul 22, 2017 - Jul 29, 2017	Live Event
SANS Network Security 2017	Las Vegas, NV	Sep 10, 2017 - Sep 17, 2017	Live Event
SANS OnDemand	Online	Anytime	Self Paced
SANS SelfStudy	Books & MP3s Only	Anytime	Self Paced