Linux Security Quick Reference Guide

LinuxSecurity.com

The Linux Community's Center for Security

www.LinuxSecurity.com

Introduction:

The intent of this Quick Reference Guide is to provide a starting point for improving the security of your system, to serve as a pointer to more in-depth security information, and to increase security awareness and methods that can be used to improve security. It is not a substitute for reading any of the vast amounts of Linux security documentation that already exists.

In the ever-changing world of global data communications, inexpensive Internet connections, and fast-paced software development, security is becoming more and more of an issue. Security is now a basic requirement because global computing is inherently insecure. As your data goes from point A to point B on the Internet, it may pass through several other points along the way, giving other users the opportunity to intercept, and even alter, your data. Even other users on your system may maliciously transform your data into something you did not intend. Unauthorized access to your system may be obtained by intruders, also known as "crackers", who then use advanced knowledge to impersonate you, steal information from you, or even deny you access to your own resources.

Security involves defense in depth. Approaching security a step at a time, with consistency and vigilance, you can mitigate the security threats, and keep the crackers at bay. Keep your system up to date by making sure you have installed the current versions of software and are aware of all security alerts. Doing this alone will help make your system markedly more secure.

The more secure your system is the more intrusive your security becomes. You need to decide where in this balancing act your system will still be usable yet secure for your purposes.

If you have more than one person logging on to your machine, or machines, you should establish a "Security Policy" stating how much security is required by your site and what auditing is in place to monitor it.

Controlling File Permissions & Attributes:

Monitoring the permissions on system files is crucial to maintain host integrity.

• Regularly audit your systems for any unauthorized and unnecessary use of the setuid or setgid permissions. "Set-user-ID root" programs run as the `root` user, regardless of who is executing them, and are a frequent cause of buffer overflows. Many programs are setuid and setgid to enable a normal user to perform operations that would otherwise require root, and can be removed if your users do not need such permission. Find all setuid and setgid programs on your host and descriminately remove the setuid or setgid permissions on a suspicious program with `chmod`:

```
root# find / -type f -perm +6000 -ls
59520   30 -rwsr-xr-x   1 root   root  30560 Apr 15  1999 /usr/bin/chage
59560   16 -r-sr-sr-x   1 root   lp    15816 Jan  6  2000 /usr/bin/lpq
root# chmod -s /usr/bin/chage /usr/bin/lpq
root# ls -l /usr/bin/lpq /usr/bin/chage
-rwxr-xr-x   1 root     root         30560 Apr 15  1999 /usr/bin/chage
-r-xr-xr-x   1 root     lp           15816 Jan  6  2000 /usr/bin/lpq
```

• World-writable files are easily altered or removed. Locate all world-writable files on your system:

```
root# find / -perm -2 ! -type l -ls
```

In the normal course of operation, several files will be world-writable, including some from /**dev** and the /**tmp** directory itself.

• Locate and identify all files that do not have an owner or belong to a group. Unowned files may also be an indication an intruder has accessed your system.

```
root# find / -nouser -o -nogroup
```

• Using the `lsattr` and `chattr` commands, administrators can modify characteristics of files and directories, including the ability to control deletion and modification above what normal `chmod` provides. The use of "append-only" and "immutable" attributes can be particularly effective in preventing log files from being deleted, or Trojan Horses from being placed on top of trusted binaries. While not a guarantee a system file or log won't be modified, only root has the ability to remove this protection. The `chattr` command is used to add or remove these properties, while the `lsattr` can be used to list them.

Log files can be protected by only permitting appending to them. Once the data has been written, it cannot be removed. While this will require modifications to your log rotation scripts, this can provide additional protection from a cracker attempting to remove his tracks. Once rotated, they should be changed to immutable. Files suitable for these modifications include `/bin/login`, `/bin/rpm`, `/etc/shadow`, and others that should not change frequently.

```
# chattr +i /bin/login
# chattr +a /var/log/messages
# lsattr /bin/login /var/log/messages
----i--- /bin/login
-----a-- /var/log/messages
```

• There should never be a reason for user's to be able to run setuid programs from their home directories. Use the `nosuid` option in `/etc/fstab` for partitions that are writable by others than root. You may also wish to use the `nodev` and `noexec` on user's home partitions, as well as `/var`, which prohibits execution of programs, and creation of character or block devices, which should never be necessary anyway. See the `mount` man page for more information.

Security Glossary:

• **Buffer Overflow:** A condition that occurs when a user or process attempts to place more data into a program's storage buffer in memory and then overwrites the actual program data with instructions that typically provide a shell owned by root on the server. Accounted for more than 50 percent of all major security bugs leading to security advisories published by CERT. Typically associated with set-user-ID root binaries.

• **Cryptography:** The mathematical science that deals with transforming data to render its meaning unintelligible, prevent its undetected alteration, or prevent its unauthorized use.

• **Denial of Service:** Occurs when a resource is targeted by an intruder to prevent legitimate users from using that resource. They are a threat to the availability of data to all others trying to use that resource. Range from unplugging the network connection to consuming all the available network bandwidth.

• **IP Spoofing:** An attack in which one host masquerades as another. This can be used to route data destined for one host to antoher, thereby allowing attackers to intercept data not originally intended for them. It is typically a one-way attack.

• **Port Scanning:** The process of determining which ports are active on a machine. By probing as many hosts as possible, means to exploit the ones that respond can be developed. It is typically the precursor to an attack.

• **Packet Filtering:** A method of filtering network traffic as it passes between the firewall's interfaces at the network level. The network data is then analyzed according to the information available in the data packet, and access is granted or denied based on the firewall security policy. Usually requires an intimate knowledge of how network protocols work.

• **Proxy Gateway:** Also called Application Gateways, act on behalf of another program. A host with a proxy server installed becomes both a server and a client, and acts as a choke between the final destination and the client. Proxy servers are typically small, carefully-written single-purpose programs that only permit specific services to pass through it. Typically combined with packet filters.

• **Set User-ID (setuid)** / **Set Group-ID (setgid):** Files that everyone can execute as either it's owner or group privileges. Typically, you'll find root-owned setuid files, which means that regardless of who executes them, they obtain `root` permission for the period of time the program is running (or until that program intentionally relinquishes these privileges). These are the types of files that are most often attacked by intruders, because of the potential for obtaining `root` privileges. Commonly associated with buffer overflows.

• **Trojan Horse:** A program that masquerades itself as a benign program, when in fact it is not. A program can be modified by a malicious programmer that purports to do something useful, but in fact contains a malicious program containing hidden functions, exploiting the privileges of the user executing it. A modified version of /**bin**/**ps**, for example, may be used to hide the presence of other programs running on the system.

• **Vulnerability:** A condition that has the potential for allowing security to be compromised. Many different types of network and local vulnerabilities exist and are widely known, and frequently occur on computers regardless of their level of network connectivity, processing speed, or profile.

Kernel Security:

Several kernel configuration options are available to improve security through the `/proc` pseudo-filesystem. Quite a few of the files in `/proc/sys` are directly related to security. Enabled if contains a `1` and disabled if it contains a `0`. Many of the options available in `/proc/sys/net/ipv4` include:

• **icmp_echo_ignore_all**: Ignore all ICMP ECHO requests. Enabling this option will prevent this host from responding to `ping` requests.

• **icmp_echo_ignore_broadcasts**: Ignore ICMP echo requests with a broadcast/multicast destination address. Your network may be used as an exploder for denial of service packet flooding attacks to other hosts.

• **ip_forward**: Enable or disable the forwarding of IP packets between interfaces. Default value is dependent on whether the kernel is configured as host or router.

• **ip_masq_debug**: Enable or disable debugging of IP masquerading.

• **tcp_syncookies**: Protection from the "SYN Attack". Send syncookies when the SYN backlog queue of a socket overflows.

• **rp_filter**: Determines if source address verification is enabled. Enable this option on all routers to prevent IP spoofing attacks against the internal network.

• **secure_redirects**: Accept ICMP redirect messages only for gateways listed in default gateway list.

• **log_martians**: Log packets with impossible addresses to kernel log.

• **accept_source_route**: Determines whether source routed packets are accepted or declined. Should be disabled unless specific reason requires it.

The file /**etc**/**sysctl.conf** on recent Red Hat contains a few default settings and is processed at system startup. The /**sbin**/**sysctl** program can be used to control these parameters. It is also possible to configure their values using /**bin**/**echo**. For example, to disable IP forwarding, as **root** run:

```
echo "0" > /proc/sys/net/ipv4/ip_forward
```

This must written to a system startup file or `/etc/sysctl.conf` on Red Hat to occur after each reboot. More information is available in `proc.txt` file in the kernel `Documentation/` directory.

General Security Tips:

• `AutoRPM` on Red Hat and `apt-get` on Debian can be used to download and install any packages on your system for which there are updates. Use care when automatically updating production servers.

• IP Masquerading enables a Linux box with multiple interfaces to act as a gateway to remote networks for hosts connected to the Linux box on the internal network interface. See the IP Masquerading HOWTO for implementation information.

• Install `nmap` to determine potential communication channels. Can determine remote OS version, perform "stealth" scans by manipulating ICMP, TCP and UDP, and even potentially determine the remote username running the service. Start with something simple like:

```
# nmap 192.168.1.1
```

• Password-protect LILO for servers in public environments to require authorization when passing LILO command-line kernel parameters at boot time. Add the `password` and `restricted` arguments to `/etc/lilo.conf`, then be sure to re-run `/sbin/lilo`:

```
image = /boot/vmlinuz-2.2.17
          label = Linux
          read-only
          restricted
          password = your-password
```

• The OpenWall kernel patch is a useful set of kernel security improvements that helps to prevent buffer overflows, restrict information in `/proc` available to normal users, and other changes. Requires compiling the kernel, and not for newbies.

• Ensure system clocks are accurate. The time stamps on log files must be accurate so security events can be correlated with remote systems. Inaccurate records make it impossible to build a timeline. For workstations, it is enough to add a crontab entry:

```
0-59/30 * * * * root /usr/sbin/ntpdate -su time.timehost.com
```

• Install and execute the Bastille Linux hardening tool. Bastille is a suite of shell scripts that eliminates many of the vulnerabilities that are common on default Linux installations. It enables users to make educated choices to improve security by asking questions as it interactively steps through securing the host. Features include basic packet filtering, deactivating unnecessary network services, auditing file permissions, and more. Try the non-intrusive test mode first.

• Configure `sudo` (superuser do) to execute privileged commands as a normal user instead of using `su`. The administrator supplies his own password to execute specific commands that would otherwise require `root` access. The file `/etc/sudoers` file controls which users may execute which programs. To permit Dave to only manipulate the printer on **magneto**:

```
Cmnd_Alias LPCMDS = /usr/sbin/lpc, /usr/bin/lprm
dave       magneto = LPCMDS
```

Dave executes `sudo` with the authorized command and enters his own password when prompted:

```
dave$ sudo /usr/sbin/lpc
Password: <password>
lpc>
```

• Password security is the most basic means of authentication, yet the most critical means to protect your system from compromise. It is also one of the most overlooked means. Without an effective well-chosen password, your system is sure to be compromised. Obtaining access to any user account on the system is the tough part. From there, root access is only a step away. Run password-cracking programs such as *John the Ripper* or *Crack* regularly on systems for which you're responsible to ensure password security is maintained. Disable unused accounts using `/usr/bin/passwd -l`. Use the MD5 password during install if your distribution supports it.

• Packet filtering isn't just for firewalls. Using `ipchains`, you can provide a significant amount of protection from external threats on any Linux box. Blocking access to a particular service from connecting outside of your local network you might try:

```
# ipchains -I input -p TCP -s 192.168.1.11 telnet -j DENY -l
```

This will prevent incoming access to the `telnet` port on your local machine if the connection originates from `192.168.1.11`. This is a very simple example. Be sure to read the IP Chains HOWTO before implementing any firewalling.

Network Intrusion Detection:

Intrusion detection devices are an integral part of any network. The Internet is constantly evolving, and new vulnerabilities and exploits are found regularly. They provide an additional level of protection to detect the presence of an intruder, and help to provide accoutability for the attacker's actions.

The `snort` network intrusion detection tool performs real-time traffic analysis, watching for anamolous events that may be considered a potential intrusion attempt. Based on the contents of the network traffic, at either the IP or application level, an alert is generated. It is easily configured, utilizes familiar methods for rule development, and takes only a few minutes to install. Snort currently includes the ability to detect more than 1100 potential vulnerabilities. It is quite feature-packed out of the box:

- Detect and alert based on pattern matching for threats including buffer overflows, stealth port scans, CGI attacks, SMB probes and NetBIOS queries, NMAP and other portscanners, well-known backdoors and system vulnerabilities, DDoS clients, and many more;
- Can be used on an existing workstation to monitor a home DSL connection, or on a dedicated server to monitor a corporate web site.

Linux Security Resources:

• Apache directory and password protection
http://www.apacheweek.com/features/userauth

• Bastille Linux Project
http://www.bastille-linux.org

• BugTraq Full Disclosure Mailing List
http://www.securityfocus.com/forums/bugtraq/intro.html

• Building Internet Firewalls, Second Edition
O'Reilly & Assoc, ISBN 1565928717

• CERT Security Improvement Modules
http://www.cert.org/security-improvement

• Introduction to Linux Security
http://www.linux-mag.com/1999-10/security_01.html

• Linux Intrusion Detection Resources
http://www.linuxsecurity.com/intrusion-detection

• John the Ripper Password Cracker
http://www.openwall.com/john

• Linux and Open Source Security Advisories
http://www.linuxsecurity.com/advisories

• LinuxSecurity.com Security Reference Info
http://www.linuxsecurity.com/docs

• LinuxSecurity.com Security Discussion Lists
http://www.linuxsecurity.com/mailing-lists.html

• LinuxSecurity.com Tip of the Day
http://www.linuxsecurity.com/tips

• LinuxSecurity.com Weekly Security Newsletter
http://www.linuxsecurity.com/newsletter.html

• OpenSSH secure remote access tool
http://www.openssh.com

• OpenWall Security Project
http://www.openwall.com

• Network Time Protocol information
http://www.ntp.org

• nmap Port Scanner
http://www.insecure.org/nmap

• Practical UNIX & Internet Security, Second Ed.
O'Reilly & Assoc, ISBN 1565921488

• rsync Incremental File Transfer Utility
http://rsync.samba.org

• Secure Shell FAQ
http://www.employees.org/~satch/ssh/faq/

• Security-related HOWTOs and FAQs
http://www.linuxsecurity.com/docs

• Site Security Handbook (RFC2196)
http://www.linuxsecurity.com/docs/rfcs/rfc2196.txt

• sudo root access control tool
http://www.courtesan.com/sudo

• Snort Network Intrusion Detection System
http://www.snort.org

• Tripwire file integrity tool
http://www.tripwiresecurity.com

• Using Snort
http://www.linuxsecurity.com/using-snort.html

Implementation By Dave Wreski
Concept By Benjamin Thomas
 v1.1

Disable Unnecessary Services:

Disabling or removing unused programs and services from your host is the most effective way to limit threats originating from a remote host. Use your distributions package management tools to scan the list of installed packages, then remove those that are unnecessary.

• Many of the services running from `inetd` are legacy programs, which are hardly ever required, yet typically enabled by default. The file `/etc/inetd.conf` is used to specify which services are offered. Disable all services that you do not want to provide by commenting them out using the `#` character in the first column of the line.

• The `/etc/rc*.d` or `/etc/rc.d/rc*` directories contains shell scripts that control the execution of network and system services during runlevels. Rename or otherwise disable any that are not required or remove the package entirely. Red Hat users can use `/sbin/chkconfig --list` to list which services run in which runlevel, and **/sbin/chkconfig --del <name>** to disable a service.

If you don't understand what a particular service does, disable it until you find out. Use `netstat` and `ps` to confirm they have not been started after a reboot. Use `/bin/netstat -a -p --inet` to determine which are available and the process ID associated with them. A port scanner should also be used to get a view of what remote hosts see.

Checking Package Integrity:

The `md5sum` command is used to compute a 128-bit fingerprint that is strongly dependant upon the contents of the file to which it is applied. It can be used to compare against a previously-generated sum to determine whether the file has changed. It is commonly used to ensure the integrity of updated packages distributed by a vendor:

```
# md5sum package-name
995d4f40cda13eacd2beaf35c1c4d5c2  package-name
```

The string of numbers can then be compared against the MD5 checksum published by the packager. While it does not take into account the possibility that the same person that may have modified a package also may have modified the published checksum, it is especially useful for establishing a great deal of assurance in the integrity of a package before installing it.

Install and Configure OpenSSH:

OpenSSH is a replacement for `telnet` and `ftp` that eliminates eavesdropping, connection hijacking, and encrypts all communication between hosts. One of the most indepensible free security tools in existence.

• Install the OpenSSH and OpenSSL Packages:

```
openssh-<current-version>.rpm
openssh-server-<current-version>.rpm
openssh-clients-<current-version>.rpm
openssl-<current-version>.rpm
```

• Generate Public/Private Key Pair:

OpenSSH uses public key cryptography to provide secure authorization. Generating the public key, which is shared with remote systems, and the private key which is kept on the local system, is done first to configure OpenSSH.

```
orion$ ssh-keygen
Generating RSA keys:  ...ooooooO....ooooooO
Key generation complete.
Enter file in which to save the key (/home/dave/.ssh/identity):
Created directory '/home/dave/.ssh'.
Enter passphrase (empty for no passphrase): <passphrase>
Enter same passphrase again: <passphrase>
Your identification has been saved in /home/dave/.ssh/identity.
Your public key has been saved in /home/dave/.ssh/identity.pub.
The key fingerprint is:
ac:42:11:c8:0d:b6:7e:b4:06:6a:a3:a7:e8:2c:b0:12 dave@orion
```

• Copy Public Key to Remote Host:

```
host2$ mkdir -m 700 ~dave/.ssh
host2$ cp /mnt/floppy/identity.pub ~dave/.ssh/authorized_keys
```

• Log in to Remote Host:

The SSH client (`/usr/bin/ssh`) is a drop-in replacement for `rlogin` and `rsh`. It can be used to securely login to a remote host:

```
orion$ ssh host2
Enter passphrase for RSA key 'dave@orion': <passphrase>
Last login: Sat Aug 15 17:13:01 2000 from orion
No mail.
host2$
```

• Copy Files to Remote Host:

The OpenSSH package also includes `scp`, a secure and improved replacement for `rcp`. This allows you to securely files over a network.

```
orion$ scp /tmp/file.tar.gz host2:/tmp
Enter passphrase for RSA key 'dave@orion:
file.tar.gz  100% |***************************| 98304 00:00
```

It is also possible to encapsulate ordinarily insecure protocols such as IMAP and POP within SSH to prevent transmitting clear text passwords to your mail server. Additionally, the `rsync` incremental file transfer utility can use SSH to securely synchronize two hosts or backup data to a log server securely. SSH can even be used to securely connect two subnets across the Internet, effectively creating a virtual private network. Disable remote root logins and emtpy password ability.

Apache Security:

• Limit Apache to listen only on local interface by configuring `/etc/httpd/conf/httpd.conf` to read:

```
Listen 127.0.0.1:80
```

• Use the following to disable access to the entire filesystem by default, unless explicitly permitted. This will disable printing of indexes if no **index.html** exists, server-side includes, and following symbolic links. Disabling symlinks may impact performance for large sites.

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

• Use the following to control access to the server from limited addresses in `/etc/httpd/conf/access.conf` to read:

```
<Directory /home/httpd/html>
 # Deny all accesses by default
 Order deny,allow
 # Allow access to local machine
 Allow from 127.0.0.1
 # Allow access to entire local network
 Allow from 192.168.1.
 # Allow access to single remote host
 Allow from 192.168.5.3
 # Deny from everyone else
 Deny from all
</Directory>
```

• Use the following to require password authentication when attempting to access a specific directory in `/etc/httpd/conf/access.conf`:

```
<Directory /home/httpd/html/protected>
 Order Deny,Allow
 Deny from All
 Allow from 192.168.1.11
 AuthName "Private Information"
 AuthType Basic
 AuthUserFile /etc/httpd/conf/private-users
 AuthGroupFile /etc/httpd/conf/private-groups
 require group <group-name>
</Directory>
```

Create the `private-groups` file using the following format:

```
group-name: user1 user2 user...
```

Create password entries for each user in the above list:

```
# htpasswd -cm /etc/httpd/conf/private-users user1
New password: <password>
Re-type new password: <password>
Adding password for user user1
```

Be sure to restart apache and test it. This will result in the enabling of double reverse lookups to verify the identity of the remote host. Remove the `-c` option to `htpasswd` after the first user has been added. Be sure the password file you create is not located within the `DocumentRoot` to prevent it from being downloaded.

Configuring TCP Wrappers:

Frequently used to monitor and control access to services listed in `/etc/inetd.conf`. The `in.ftpd` service might be wrapped using:

```
ftp stream tcp nowait root /usr/sbin/tcpd  in.ftpd -l -L -i -o
```

Before the `in.telnetd` daemon is spawned, `tcpd` first determines if the source is a permitted host. Connection attempts are sent to `syslogd`. All services should be disabled by default in `/etc/hosts.deny` using the following:

```
ALL: ALL
```

To send an email to the admin and report failed connection attempt:

```
ALL: ALL: /bin/mail \
  -s "%s connection attempt from %c" admin@mydom.com
```

Enable specific services in `/etc/hosts.allow` using the service name followed by the host:

```
sshd: magneto.mydom.com, juggernaut.mydom.com
in.ftpd: 192.168.1.
```

Trailing period indicates entire network should be permitted. Use `tcpdchk` to verify your access files. A syslog entry will be created for failed attempts. Access control is performed in the following order:

- Access will be granted when a daemon/client pair matches an entry in the `/etc/hosts.allow` file.
- Otherwise, access will be denied when a daemon/client pair matches an entry in the `/etc/hosts.deny` file.
- Otherwise, access will be granted.

A non-existing access control file is treated as if it were an empty file. Thus, access control will be turned off if no access control files are present!

Using RPM and dpkg:

The `/bin/rpm` program on Red Hat and derivitives and the `/usr/bin/dpkg` on Debian and derivitives are used to control the management of packages.

• Remove a package

```
# rpm -e <package-name>
# dpkg -r <package-name>
```

• List contents of entire package

```
# rpm -qvl <package-name.rpm>
# dpkg -c <package-name.deb>
```

• List all installed packages with info about each

```
# rpm -qvia
# dpkg -l
```

• List contents of a package

```
# rpm -qvpl <package-name.rpm>
# dpkg -c <package-name.deb>
```

• Print information about a package

```
# rpm -qpi <package-name.rpm>
# dpkg -I <package-name.deb>
```

• Verify package characteristics (basic integrity check)

```
# rpm -Va
# debsums -a
```

• Determine to which package a file belongs

```
# rpm -qf </path/to/file>
# dpkg -S </path/to/file>
```

• Install new package

```
# rpm -Uvh <package-name.rpm>
# dpkg -i <package-name.deb>
```

Configuring Syslog:

The `syslogd` is responsible for capturing logging information generated by system processes. The `klogd` is responsible for capturing logging information generated by the kernel. System logs provide the primary indication of a potential problem.

• Fine-tune the default `/etc/syslog.conf` to send log information to specific files for easier analysis.

```
# Monitor authentication attempts
auth.*;authpriv.*                    /var/log/authlog

# Monitor all kernel messages
kern.*                               /var/log/kernlog

# Monitor all warning and error messages
*.warn;*.err                         /var/log/syslog

# Send a copy to remote loghost.  Configure syslogd init
# script to run with -r -s domain.com options on log
# server. Ensure a high level of security on the log
# server!
*.info                               @loghost
auth.*;authpriv.*                    @loghost
```

• Restrict access to log directory and syslog files for normal users using:

```
# chmod 751 /var/log /etc/logrotate.d
# chmod 640 /etc/syslog.conf /etc/logrotate.conf
# chmod 640 /var/log/*log
```

Install and Configure Tripwire:

Tripwire is a program that monitors file integrity by maintaining a database of cryptographic signatures for programs and configuration files installed on the system, and reports changes in any of these files.

A database of checksums and other characteristics for the files listed in the configuration file is created. Each subsequent run compares any differences to the reference database, and the administrator is notified.

The greatest level of assurance that can be provided occurs if Tripwire is run immediately after Linux has been installed and security updates applied, and before it is connected to a network.

A text configuration file, called a policy file, is used to define the characteristics for each file that are tracked. Your level of paranoia determines the frequency in which the integrity of the files are checked. Administration requires constant attention to the system changes, and can be time-consuming if used for many systems. Available in unsupported commercial binary for Red Hat and similar.

```
# Create policy file from text file
/usr/TSS/bin/twadmin -m P policy.txt

# Initialize database according to policy file
/usr/TSS/bin/tripwire -init

# Print database
/usr/TSS/bin/twprint -m d

# Generate daily report file
/usr/TSS/bin/tripwire -m c -t 1 -M

# Update database according to policy file and report file
/usr/TSS/bin/tripwire --update --polfile policy/tw.pol \
   --twrfile report/<hostname>-<date>.twr
```

DNS Security:

• Zone transfers should only be permitted by master name servers to update the zone (domain) information in their slave servers. Failure to do so may result in IP numbers and hostnames being revealed to unauthorized users. Restrict queries to only public domains. Suitable for name servers with both public and private zones.

```
// Allow transfer only to our slave name server. Allow queries
// only by hosts in the 192.168.1.0 network.
zone "mydomain.com" {
    type master;
    file "master/db.mydomain.com";
    allow-transfer { 192.168.1.6; };
    allow-query { 192.168.1.0/24; };
};
```

• Deny and log queries for our version number except from the local host. The ability to determine the bind version enables an attacker to find the corresponding exploit for that version.

```
// Disable the ability to determine the version of BIND running
zone "bind" chaos {
    type master;
    file "master/bind";
    allow-query { localhost; };
};
```

The `./master/bind` file should then contain:

```
$TTL 1d
@     CHAOS   SOA     localhost. root.localhost. (
            1          ; serial
           3H          ; refresh
          15M          ; retry
           1W          ; expire
           1D )        ; minimum
           NS          localhost.
```

• Control which interfaces `named` listens on. Restricting the interfaces on which **named** runs can limit the exposure to only the necessary networks.

```
listen-on { 192.168.1.1; };
```

• Use Access Control Lists to classify groups of hosts with differing degrees of trust. The "internal" ACL label might be used to describe internal hosts that are permitted a greater degree of access to the information than other hosts might be. Before it can be used it must be defined:

```
acl "internal" {
      { 192.168.1.0/24; 192.168.2.11; };
};
```

It can then be used in "zone" statements or the main "options" statement:

```
zone "inside.mynet.com" {
       type master;
       file "master/inside.mynet.com";
       allow-query { "internal"; };
};
```

• Configure BIND to run as a normal user. Once BIND has been started, it has the ability to relinquish its privileges, and run as a user with limited abilities instead of **root**.

```
# useradd -M -r -d /var/named -s /bin/false named
# groupadd -r named
```

This account should be used for nothing other than running the name server. Ensure the zone files are readable by the `named` user. It is then necessary to modify the default `named` init script, typically found in `/etc/rc.d/init.d/named` on Red Hat or `/etc/init.d/named` on Debian:

```
/usr/sbin/named -u named -g named
```

It is also possible to run `named` in a "chroot jail" which helps to restrict the damage that can be done should `named` be subverted.

Critical System Files:

File/Directory	Perms	Description
`/var/log`	751	Directory containing all log files
`/var/log/messages`	644	System messages
`/etc/crontab`	600	System-wide crontab file
`/etc/syslog.conf`	640	Syslog daemon configuration file
`/etc/logrotate.conf`	640	Controls rotating of system log files
`/var/log/wtmp`	660	Who is logged in now. Use `who` to view
`/var/log/lastlog`	640	Who has logged in before. Use `last` to view
`/etc/ftpusers`	600	List of users that *cannot* FTP
`/etc/passwd`	644	List of the system's user accounts
`/etc/shadow`	600	Contains encrypted account passwords
`/etc/pam.d`	750	PAM configuration files
`/etc/hosts.allow`	600	Access control file
`/etc/hosts.deny`	600	Access control file
`/etc/lilo.conf`	600	Boot loader configuration file
`/etc/securetty`	600	TTY interfaces that allow root logins
`/etc/shutdown.allow`	400	Users permitted to ctrl-alt-del
`/etc/security`	700	System access security policy files
`/etc/rc.d/init.d`	750	Program start-up files on Red Hat systems
`/etc/init.d`	750	Program start-up files on Debian systems
`/etc/sysconfig`	751	System and network config files on Red Hat
`/etc/inetd.conf`	600	Internet SuperServer configuration file
`/etc/cron.allow`	400	List of users permitted to use `cron`
`/etc/cron.deny`	400	List of users denied access to `cron`
`/etc/ssh`	750	Secure Shell configuration files
`/etc/sysctl.conf`	400	Contains kernel tunable options on recent Red Hat