LIDS FAQ

Steve Bremer, *steve@clublinux.org* v.01, Dec 18, 2000

This is the Linux Intrusion Detection System (LIDS) FAQ.

Contents

1 Introduction to LIDS

1.1 What is LIDS?

LIDS is an enhancement for the Linux kernel written by *Xie Huagang* and *Philippe Biondi*. It implements several security features that are not in the Linux kernel natively. Some of these include: mandatory access controls (MAC), a port scan detector, file protection (even from root), and process protection.

1.2 Why use LIDS?

The current Linux setup has many problems that are inherit in many version of *nix. Probably the single largest problem is the "all powerful" root account. When a process or user has root privileges, there is little if nothing to prevent that process or user from completely destroying the system. A malicious user/intruder with root access can cause much heartache for us hard working sysadmins. LIDS implements access control lists (ACLs) that will help prevent even those with access to the mighty root account from wrecking havoc on a system. These ACLs allow LIDS to protect files as well as processes.

1.3 Where can I obtain LIDS?

www.lids.org

1.4 Which versions of the Linux kernel are supported?

Currently, LIDS supports the latest 2.2.x kernels as well as the new 2.4 kernel. Xie has expressed interest in making 2.4 the primary kernel for LIDS support. However, he also has stated he would maintain a stable version of LIDS for the 2.2.x series.

1.5 Is there a LIDS mailing list?

Yes. You can post to the list at any time by e-mailing `lids@egroups.com`. However, if you wish to receive messages posted to the mailing list, you must subscribe to it. To subscribe, simply e-mail `lids-subscribe@egroups.com`. You will then receive a confirmation request that you must reply to. To un-subscribe from the list, e-mail `lids-unsubscribe@egroups.com`.

1.6 What about an archive?

The mailing list archive is located at *http://www.egroups.com/list/lids*.

1.7 Copyright & Disclaimer

The information here in this document is, to the best of Steve's knowledge, correct. However, being human, there is the chance that mistakes, bugs, etc. might happen from time to time.

No person, group, or other body is responsible for any damage on your computer(s) and any other losses by using the information on this document. i.e.

> **THE AUTHORS AND ALL MAINTAINERS ARE NOT RESPONSIBLE FOR ANY DAMAGES INCURRED DUE TO ACTIONS TAKEN BASED ON THE INFORMATION IN THIS DOCUMENT.**

1.8 Feedback

If you have any questions, comments, suggestions, or corrections to this document, please feel free to contact me at *steve@clublinux.org*. I always welcome feedback whether it's good or bad!

1.9 Credit

Special thanks go to:

- **Xie Huagang** - Technical editor and LIDS author.
  - 7.4 (LIDS version) question.
  - 4.11 (Subject/object) question.
- **Philippe Biondi** - LIDS author.
- **Andy Harrelson** - Grammar/spelling editor.
- **Rob Willis** - 6.6 (OpenSSH), 6.7 (OpenLDAP), and 6.8 (Port Sentry) configuration examples.
- **Fred Mobach** - Inspiration.

> **Linux is a tradmark of Linus Torvalds**

1.10 To Do

- File inheritance (-i).
- Exec domain feature (-d).
- Configuring e-mail alerts.
- Kernel configuration options.
- LIDS Debug.

1.11 Change Log

The latest version of this FAQ can be found at *http://www.clublinux.org/lids/LIDS-FAQ.html*.

- January 16th, 2001. Version .01 Initial release.

2 Installing LIDS

2.1 How do I apply the LIDS kernel patch?

Xie has included *instructions* on how to patch the kernel in the LIDS download. However, I will briefly cover the necessary steps. This example assumes your kernel sources are installed in /usr/src/linux.

- First you need to download the LIDS patch from *www.lids.org/download.html*. Make sure you get the version that matches your kernel.
- Then, expand the tarball:

  ```
  $ tar zxvf lids-<lids_version>-<kernel_version>.tar.gz
  ```

- Apply the lids patch to the existing kernel sources:

  ```
  $ cd /usr/src/linux
  $ patch -p1 < /path/to/lids/patch/lids-<lids_version>-<kernel_version>.patch
  ```

- Then configure your kernel. For an excellent source of information on recompiling your Linux kernel, see the ***Linux Kernel HOW-TO.*** There are several kernel configuration options for LIDS. In order for LIDS to work, you must make sure the following options are enabled:

  ```
  [*]   Prompt for development and/or incomplete code/drivers
  [*]   Sysctl Support
  ```

2.2 How do I install the LIDS administration utility lidsadm?

The source for the lidsadm utility is located in the directory containing your LIDS source and is called:

```
lidsadm-<lids_version>
```

(**NOTE:** If you are upgrading lidsadm, you should backup everything in the /etc/lids directory first!)

To compile and install lidsadm simply:

```
$ make
# make install
```

from the lidsadm source directory. This will install lidsadm in the /sbin directory. It will also create an /etc/lids directory and place a few default configuration files in it for you.

If you wish to use the view option with lidsadm, replace the

```
$ make
```

with

```
$ make VIEW=1
```

2.3 What next?

Before you reboot into your LIDS enhanced kernel, you should configure your LIDS ACLs first. Otherwise your system may be unusable when you reboot. Configuring LIDS ACLs is covered 5 (later).

3 lidsadm

3.1 What is lidsadm?

lidsadm is the LIDS administration utility that you will use to configure LIDS to enhance your system security.

3.2 What options are available for lidsadm?

To get a list of the available options, enter the following:

```
# lidsadm -h
```

This will return the following output:

```
lidsadm v0.9 for Linux Intrusion Detection System
        Xie Huagang<xhg@ncic.ac.cn>
                Philippe Biondi <philippe.biondi@webmotion.net>

Usage: lidsadm -A [-s subject] -o object [-t] -j TARGET
       lidsadm -D [-s file] [-o file]
       lidsadm -Z
       lidsadm -U
       lidsadm -L
       lidsadm -P
       lidsadm -[S|I] -- [+|-][CAPABILITY|LIDS_FLAG] [...]
       lidsadm -V
       lidsadm -h

Commands:
       -A  To add an entry
       -D  To delete an entry
       -Z  To delete all entries
       -U  To update dev/inode numbers
       -L  To list all entries
       -P  To encrypt a password with RipeMD-160
       -S  To submit a password to switch some protections
       -I  To switch some protections without submitting password (sealing time)
       -V  To view current LIDS state (caps/flags)
       -h  To list this help

subject:
        can be any program,must be file
```

```
object:
        can be file,directory, or special device
        such as MEM,HD,NET,IO,HIDDEN,KILL
TARGET:
        READ    read only
        APPEND  append only
        WRITE   writable
        IGNORE  ignore protection
        INHERIT the ability to access the object can inherit
        NO_INHERIT      the ability can not be inherited.
TYPE:
            -t  the object is a special device
            -d  the object is a EXEC Domain

Available capabilities:
           CAP_CHOWN chown(2)/chgrp(2)
    CAP_DAC_OVERRIDE DAC access
 CAP_DAC_READ_SEARCH DAC read
          CAP_FOWNER owner ID not equal user ID
          CAP_FSETID effective user ID not equal owner ID
            CAP_KILL real/effective ID not equal process ID
          CAP_SETGID setgid(2)
          CAP_SETUID set*uid(2)
         CAP_SETPCAP transfer capability
 CAP_LINUX_IMMUTABLE immutable and append file attributes
CAP_NET_BIND_SERVICE binding to ports below 1024
   CAP_NET_BROADCAST broadcasting/listening to multi-cast
       CAP_NET_ADMIN interface/firewall/routing changes
         CAP_NET_RAW raw sockets
        CAP_IPC_LOCK locking of shared memory segments
       CAP_IPC_OWNER IPC ownership checks
      CAP_SYS_MODULE insertion and removal of kernel modules
       CAP_SYS_RAWIO ioperm(2)/iopl(2) access
      CAP_SYS_CHROOT chroot(2)
      CAP_SYS_PTRACE ptrace(2)
       CAP_SYS_PACCT configuration of process accounting
       CAP_SYS_ADMIN tons of admin stuff
        CAP_SYS_BOOT reboot(2)
        CAP_SYS_NICE nice(2)
    CAP_SYS_RESOURCE setting resource limits
        CAP_SYS_TIME setting system time
  CAP_SYS_TTY_CONFIG tty configuration
          CAP_HIDDEN Hidden process
       CAP_INIT_KILL Kill init children

Available flags:
         LIDS_GLOBAL LIDS itself

         RELOAD_CONF reload config. file and inode/dev of special programs
                LIDS (de)activate LIDS locally (the shell & childs)
```

3.3 Gee, thanks. What are all these options?

lidsadm has a syntax similar to *IPCHAINS*. Some of the command line switches are the same.

- `-A` = Add a rule.
- `-D` = Delete a rule.
- `-L` = List all existing rules.
- `-h` = lidsadm help.
- `-Z` = Delete all existing rules.
- `-U` = Update the device/inode numbers of all files.
- `-P` = Create/update the LIDS password.
- `-V` = View current LIDS state (capabilities/flags).
- `-S` = Make changes to your LIDS enabled system (requires LIDS password set by option "-P").
- `-s` = Specifies a subject file.
- `-o` = Specifies an object file.
- `-j` = Specifies a target.
- `-t` = Specifies that the object is capability and not a file or device.
- `-I` = Seals the kernel. Used at the end of the startup process.

lidsadm also uses "TARGETS" similar to ipchains. The following targets are allowed:

- `READ -` Set access permissions to read only.
- `APPEND -` Set access permissions to append only(includes read access).
- `WRITE -` Set access permissions to read/write.
- `IGNORE -` Ignore any permissions set on this object.
- `INHERIT -` Children of this process will inherit this capability.
- `NO_INHERIT -` Children of this process will NOT inherit this capability.

NOTE: The last two TARGETS are only available for capabilities.

4 LIDS Administration

4.1 How do I set my LIDS password?

Before you reboot into your LIDS enhanced kernel, enter the following at the command prompt:

```
# lidsadm -P
```

You will then be prompted for a LIDS password:

```
MAKE PASSWD
enter password:
Verifying enter password:
```

This will write your RipeMD-160 encrypted password to the `/etc/lids/lids.pw` file.

4.2 How do I change my LIDS password once it is set?

You must first create a 4.2 (LIDS free session). Then set your password using the "-P" option just like you did 4 (the first time) (you will not be prompted for your current password). After resetting your LIDS password, you must tell LIDS to 4.4 (reload it's configuration files).

4.3 What is a LIDS free session and how do I create one?

A LIDS free session (LFS) is a terminal session that is not restricted by LIDS. This option is available so you can administer your system without having to reboot into a non-LIDS kernel. In order for this to work, you must have selected this option when you compiled your LIDS enhanced kernel:

```
[*] Allow switching LIDS protections
```

To create an LFS, enter the following at the prompt:

```
# lidsadm -S -- -LIDS
```

You will then be prompted for your LIDS password. This terminal is now LIDS free. It will remain LIDS free until you:

- Enable LIDS again (`lidsadm -S - +LIDS`).
- Log out of the terminal.

You can only have one LFS active at any one time. Even though `lidsadm -S - -LIDS` will not fail if entered on another terminal, you can have only one LFS.

4.4 I created a LIDS free session, but LIDS still appears to be active! What's wrong?

This can happen if you create a LFS on a virtual console and then switch to another virtual console and try to administer your machine. To clear it up, try enabling LIDS and then disabling it again (entering passwords when prompted):

```
# lidsadm -S -- +LIDS
# lidsadm -S -- -LIDS
```

4.5 How do I tell LIDS to reload it's configuration files?

In order for LIDS to be able to reload it's configuration files, you must enable this option when you configure your LIDS enhanced kernel:

```
[*]  Allow switching LIDS protections
(3)    Number of attempts to submit password
(30)     Time to wait after a fail (seconds)
[ ]    Allow remote users to switch LIDS protections
[ ]    Allow any program to switch LIDS protections
[*]    Allow reloading config. file <--------------------
```

NOTE: You must allow switching LIDS protections in order to enable reloading of configuration files.

The following instructs LIDS to reload it's configuration files:

```
# lidsadm -S -- +RELOAD_CONF
```

This will reload the following configuration files:

- `/etc/lids/lids.conf` - LIDS ACL configuration file.
- `/etc/lids/lids.cap` - LIDS capabilities file.
- `/etc/lids/lids.pw` - LIDS password file.
- `/etc/lids/lids.net` - LIDS mail alert configuration file.

4.6 Help!!! My system is totally unusable! What do I do?

You can reboot into a non-LIDS enhanced kernel, or boot into your LIDS enhanced kernel with LIDS disabled to try and patch things up. To boot with LIDS disabled, specify `security=0` at the lilo prompt. For example, if your LIDS enhanced kernel is called `lids-kernel` you would enter the following at the lilo prompt:

```
lilo: lids-kernel security=0
```

That's the easy part. The difficult part is getting your LIDS enabled system to shutdown. You may not be able to shutdown successfully depending on your LIDS configuration.

WARNING: Rebooting your LIDS enabled system when it is not properly configured can cause file system corruption and/or loss of data!!

4.7 I've updated/moved a system binary. How do I tell LIDS that the file changed/moved?

Whenever the device that a file resides on, or a file's inode number changes, you must update your `/etc/lids/lids.conf` file with the proper information. Fortunately, Xie has provided us with an option just for this occasion:

```
# lidsadm -U
```

You must then 4.4 (reload the configuration files).

4.8 OK, without rebooting, how do I disable LIDS all together?

Besides using a LFS, LIDS can be turned off globally. This will only work if you compiled the option into your kernel.

```
# lidsadm -S -- -LIDS_GLOBAL
```

When `LIDS_GLOBAL` is disabled, your system will operate like a "normal" Linux system. To re-enable LIDS globally, perform the opposite:

```
#lidsadm -S -- +LIDS_GLOBAL
```

NOTE: This will not affect your LFS if you currently have one enabled.

4.9 What does it mean to "seal the kernel"?

At the end of the bootup process, you should seal the kernel. This effectively enables all of LIDS features in order to protect your system. To seal the kernel, put the following at the end of your `rc.local` (assuming SysV style init):

```
/sbin/lidsadm -I
```

WARNING: If you do not seal you kernel at boot time, you will not receive the full benefits of a LIDS enhanced system.

4.10 How do I view the status of my LIDS system?

In order to use the "-V" option, you must have compiled lidsadm with `make VIEW=1` 2.2 ((see above)).

At the command line, enter:

```
# lidsadm -V
```

This will produce output similar to the following on a 2.2.x kernel:

```
VIEW
                     CAP_CHOWN 0
                CAP_DAC_OVERRIDE 0
            CAP_DAC_READ_SEARCH 0
                     CAP_FOWNER 0
                     CAP_FSETID 0
                       CAP_KILL 0
                     CAP_SETGID 0
                     CAP_SETUID 0
                    CAP_SETPCAP 0
            CAP_LINUX_IMMUTABLE 0
          CAP_NET_BIND_SERVICE 0
             CAP_NET_BROADCAST 0
                 CAP_NET_ADMIN 0
                   CAP_NET_RAW 0
```

```
          CAP_IPC_LOCK 0
         CAP_IPC_OWNER 0
        CAP_SYS_MODULE 0
         CAP_SYS_RAWIO 0
        CAP_SYS_CHROOT 0
        CAP_SYS_PTRACE 0
         CAP_SYS_PACCT 0
         CAP_SYS_ADMIN 0
          CAP_SYS_BOOT 1
          CAP_SYS_NICE 0
    CAP_SYS_RESOURCE 1
          CAP_SYS_TIME 0
  CAP_SYS_TTY_CONFIG 0
           CAP_HIDDEN 1
       CAP_INIT_KILL 0
         LIDS_GLOBAL 1
                     0
         RELOAD_CONF 0
                LIDS 0
```

As you can see from the output above, this system has a LFS active. However, LIDS is enabled globally. The items with a "1" next to them are enabled, and those items with a "0" next to them are disabled. Except for the last two capabilities, root normally has all of the above capabilities. Thanks to LIDS, root only has capabilities CAP_SYS_BOOT, CAP_SYS_RESOURCE, and CAP_HIDDEN in this particular case (NOTE: CAP_HIDDEN isn't a capability provided by the standard Linux kernel).

4.11 How do I configure the port scan detector in LIDS?

You don't. As long as you selected the option when you configured your LIDS enhanced kernel, the port scan detector is enabled.

```
[*]  Port Scanner Detector in kernel
```

4.12 What are the subject and object in a LIDS ACL?

The subject is a program that can run on a Linux system, such as a binary or shell script. The object is what the subject wants to access. This includes files, directories, capabilities, etc.

5 Configuring LIDS

5.1 How do I protect a file as read only?

```
# lidsadm -A -o /some/file      -j READ
```

This will prevent anyone (including root) from modifying or deleting `/some/file` as long as LIDS is enabled. If you are in a LFS, you are free to modify `/some/file` assuming you have appropriate file system permissions and the partition isn't mounted read-only.

5.2 OK, so how do a protect a directory as read only?

Same as above, only specify `/some/directory`

```
# lidsadm -A -o /some/directory  -j READ
```

When the object is a directory, LIDS protects the directory itself, and it recursively protects everything underneath it *within the same file system.* (e.g.`LIDS ACLs do not cross file system boundries!`). This is very important to remember so you don't accidentally leave part of your system unprotected.

A directory that you may want to protect as read only, is the `/etc` directory.

```
# lidsadm -A -o /etc -j READ
```

5.3 How can I hide a file/directory from everyone?

```
# lidsadm -A -o /some/file_or_directory   -j DENY
```

Again, this will prevent even root from accessing it. And, if it is a directory, all files and directories underneath it are also hidden (within the same file system of course).

5.4 How can I protect log files so they can only be appended to?

```
# lidsadm -A -o /some/log/file  -j APPEND
```

This will allow someone to write to the end of the file while at the same time preventing him/her from erasing or modifying it's existing contents.

An easy way to protect your system logs as append only would be:

```
# lidsadm -A -o /var/log  -j APPEND
```

This will protect all files under `/var/log` as append only. As with READ and DENY, this target is also recursive.

5.5 If nothing is allowed to read my /etc/shadow file, how can I authenticate myself to the system?

In order to allow users to authenticate themselves to the system, it is necessary to give certain programs read only access to the `/etc/shadow`. Some of the programs you may want to consider giving read access to are: `login`, `sshd`, `su`, and `vlock`.

To allow the login program to read `/etc/shadow`, use the following ACL:

```
# lidsadm -A -s /bin/login -o /etc/shadow -j READ
```

The "-s" option specifies a subject, which is `/bin/login` in this case. We are giving the subject read only access to the object (`/etc/shadow` in this case).

5.6 If I protect /etc as read only, how will mount be able to write to /etc/mtab?

It won't. You can give `mount` and `umount` write access to `/etc/mtab` like this:

```
# lidsadm -A -s /bin/mount -o /etc/mtab  -j WRITE
```

or you can remove the `/etc/mtab` file and replace it with a symbolic link to `/proc/mounts`. In order for this to work, you must modify your startup scripts to use the "-n" option with every `mount` and `umount` command. This tells `mount` and `umount` not to update the `/etc/mtab` file.

5.7 LIDS complains that it can't write to my modules.dep file during startup. What's wrong?

This happens when you protect `/lib` as read only (a good thing to do). The error received is something similar to:

```
LIDS: depmod (3 12 inode 16119) pid 13203 user (0/0) on tty2:  Try to open
/lib/modules/2.2.18/modules.dep for writing,flag=578
```

This occurs during startup because the `/etc/rc.d/rc.sysinit` init script tries to recreate all of your module dependencies. Normally this is not needed because the module dependencies don't change unless add,change, or delete modules. The error is harmless, but if you don't like seeing it, you can simply comment out the line in your `/etc/rc.d/rc.sysinit` script that recreates the module dependencies (Look for `depmod -a` or something similar).

5.8 If I protect my logs as append only, how will logrotate rotate my logs?

It won't. Log rotation is something that will have to be done manually by executing your log rotation utility when LIDS_GLOBAL is disabled. You should disable the cron job that initiates log rotation.

5.9 Why can't I just give my log rotation utility write access to the directory containing my log files so it can rotate them?

You can, but it's not recommended. If someone were to break into you system, even though they couldn't modify your logs, they could rotate them enough times that the log containing the information gathered during the intrusion is dropped off the face of the earth. This is part of the price you pay for high security.

5.10 When LIDS is active, my file systems won't unmount during shutdown. What do I do?

This happens when you have disabled the CAP_SYS_ADMIN capability globally and have not given the proper authority to unmount your file systems to your shutdown script(s). For example, on Red Hat 6.2, the `/etc/rc.d/init.d/halt` script unmounts your file systems. You must give it the CAP_SYS_ADMIN capability so it can unmount your file systems:

```
# lidsadm -A -s /etc/rc.d/init.d/halt -t -o CAP_SYS_ADMIN  -j INHERIT
```

The "-t" option tells LIDS that the object is a capability and not a device or file. The target "INHERIT" tells LIDS that all processes started by the `halt` script will inherit this capability.

Beware that this also allows anyone who can execute your `/etc/rc.d/init.d/halt` script to unmount your file systems. If you have physical access to your box, you may just want to turn off LIDS_GLOBAL before shutting down your system rather than grant capabilities to your shutdown scripts. However, if you have a UPS that can shutdown your system in case of power failure, you may not be around to disable LIDS_GLOBAL.

5.11 Why can't I start a service that runs on a privileged port as root?

Services that run a privileged port (those below 1024) require the CAP_NET_BIND_SERVICE capability in order to bind to the port. If you have disabled this capability globally in the `/etc/lids/lids.cap` file, you must either grant the program that capability

```
# lidsadm -A -s /usr/local/bin/apache -t -o CAP_NET_BIND_SERVICE -j NO_INHERIT
```

or, start the service when LIDS_GLOBAL is disabled.

5.12 Why can't I start a service that runs on a privileged port from a LFS?

A LFS applies to a single terminal session. A daemon forks itself in order to separate itself from the controlling terminal. Once this happens, it is no longer connected to the LFS on your terminal and is now protected by LIDS.

5.13 How do I disable/enable capabilities?

The `/etc/lids/lids.cap` file contains a list of all the capabilities available under a LIDS enhanced Linux kernel. Those that have a "+" in front of them are enabled, and those with a "-" in front of them are disabled. To change the status of a capability, simple edit the text file and change the "+" to a "-" to disable a capability and vice-versa to enable it. After you're done editing the file, you must tell LIDS to 4.4 (reload) the configuration files.

5.14 Why won't the X Window System work with LIDS enabled?

The X server that you are using requires the CAP_SYS_RAWIO capability. Try

```
# lidsadm -A -s /path/to/your/X_server -t -o CAP_SYS_RAWIO -j NO_INHERIT
```

5.15 With all of these ACLs, how can I possibly keep track of my configuration?

It is recommended that you create a shell script of all the ACLs that you wish to add to your system. That way you don't accidentally leave something unprotected when you make changes to your system. You can start the script out by flushing your old ACLs so you don't create duplicates.

```
# lidsadm -Z
```

5.16 I can't see my /etc/lids directory when LIDS is enabled. What's going on?

LIDS automatically protects the `/etc/lids` directory with DENY.

5.17 How can I give init write access to /etc/initrunlvl so LIDS doesn't complain about it during startup and shutdown?

Unfortunately, there isn't much you can do about this. Because `init` recreates this file each time you boot, it will have a different inode number every time. This makes it difficult for LIDS to handle. It is a harmless error, and your system will still function properly without `/etc/initrunlvl`.

6 Sample Configurations

6.1 Basic System Setup

The following is a sample configuration for basic system setup.

```
# Protect System Binaries
#
/sbin/lidsadm -A -o /sbin                                -j READ
/sbin/lidsadm -A -o /bin                                 -j READ

# Protect all of /usr and /usr/local
#
/sbin/lidsadm -A -o /usr                                 -j READ
/sbin/lidsadm -A -o /usr/local                           -j READ

# Protect the System Libraries (/usr/lib is protected above)
#
/sbin/lidsadm -A -o /lib                                 -j READ

# Protect System Configuration files
#
/sbin/lidsadm -A -o /etc                                 -j READ
/sbin/lidsadm -A -o /usr/local/etc                       -j READ
/sbin/lidsadm -A -o /etc/shadow                          -j DENY
/sbin/lidsadm -A -o /etc/lilo.conf                       -j DENY

# Enable system authentication
#
/sbin/lidsadm -A -s /bin/login -o /etc/shadow            -j READ
/sbin/lidsadm -A -s /usr/bin/vlock -o /etc/shadow        -j READ
/sbin/lidsadm -A -s /bin/su -o /etc/shadow               -j READ
/sbin/lidsadm -A -s /bin/su \
                 -t -o CAP_SETUID                        -j NO_INHERIT
/sbin/lidsadm -A -s /bin/su \
                 -t -o CAP_SETGID                        -j NO_INHERIT

# Protect the boot partition
#
/sbin/lidsadm -A -o /boot                                -j READ

# Protect root's home dir, but allow bash history
#
```

```
/sbin/lidsadm -A -o /root                               -j READ
/sbin/lidsadm -A -s /bin/bash -o /root/.bash_history    -j WRITE

# Protect system logs
#
/sbin/lidsadm -A -o /var/log                            -j APPEND
/sbin/lidsadm -A -s /bin/login -o /var/log/wtmp         -j WRITE
/sbin/lidsadm -A -s /bin/login -o /var/log/lastlog      -j WRITE
/sbin/lidsadm -A -s /sbin/init -o /var/log/wtmp         -j WRITE
/sbin/lidsadm -A -s /sbin/init -o /var/log/lastlog      -j WRITE
/sbin/lidsadm -A -s /sbin/halt -o /var/log/wtmp         -j WRITE
/sbin/lidsadm -A -s /sbin/halt -o /var/log/lastlog      -j WRITE
/sbin/lidsadm -A -s /etc/rc.d/rc.sysinit \
                 -o /var/log/wtmp                       -j WRITE
/sbin/lidsadm -A -s /etc/rc.d/rc.sysinit \
                 -o /var/log/lastlog                    -j WRITE

# Startup
#
/sbin/lidsadm -A -s /sbin/hwclock -o /etc/adjtime        -j WRITE


# Shutdown
#
/sbin/lidsadm -A -s /sbin/init -t -o CAP_INIT_KILL              -j NO_INHERIT
/sbin/lidsadm -A -s /sbin/init -t -o CAP_KILL                   -j NO_INHERIT

# Give the following init script the proper privileges to kill processes and
# unmount the file systems.  However, anyone who can execute these scripts
# by themselves can effectively kill your processes.  It's better than
# the alternative however.
#
# Any ideas on how to get around this are welcome!
#
/sbin/lidsadm -A -s /etc/rc.d/init.d/halt \
                 -t -o CAP_INIT_KILL                            -j INHERIT
/sbin/lidsadm -A -s /etc/rc.d/init.d/halt \
                 -t -o CAP_KILL                                 -j INHERIT
/sbin/lidsadm -A -s /etc/rc.d/init.d/halt \
                 -t -o CAP_NET_ADMIN                            -j INHERIT
/sbin/lidsadm -A -s /etc/rc.d/init.d/halt \
                 -t -o CAP_SYS_ADMIN                            -j INHERIT

# Other
#
/sbin/lidsadm -A -s /sbin/update -t -o CAP_SYS_ADMIN            -j INHERIT
```

6.2 Apache

This sample configuration assumes Apache was installed in `/usr/local/apache` with a log directory of `/var/log/httpd` and a configuration directory of `/etc/httpd`. You can adjust the paths in the ACLs to match your own configuration. With this configuration, Apache must be started prior to sealing the kernel, or when LIDS_GLOBAL is disabled so it can bind to port 80 (and possibly 443).

```
/sbin/lidsadm -A -s /usr/local/apache/bin/httpd \
                 -t -o CAP_SETUID                        -j NO_INHERIT
/sbin/lidsadm -A -s /usr/local/apache/bin/httpd \
                 -t -o CAP_SETGID                        -j NO_INHERIT

# Config files
/sbin/lidsadm -A -o /etc/httpd                           -j DENY
/sbin/lidsadm -A -s /usr/local/apache/bin/httpd \
                 -o /etc/httpd                           -j READ

# Server Root
/sbin/lidsadm -A -o /usr/local/apache                    -j READ
/sbin/lidsadm -A -o /usr/local/apache/bin                -j READ
/sbin/lidsadm -A -s /usr/local/apache/bin/httpd \
                 -o /usr/local/apache                    -j READ

# Log Files
/sbin/lidsadm -A -o /var/log/httpd                       -j DENY
/sbin/lidsadm -A -s /usr/local/apache/bin/httpd \
                 -o /var/log/httpd                       -j APPEND
/sbin/lidsadm -A -s /usr/local/apache/bin/httpd \
                 -o /usr/local/apache/logs               -j WRITE
```

6.3 qmail

These ACLs were written for a qmail setup that was installed according to Dave Sill's Life with qmail. With this configuration, qmail must be started prior to sealing the kernel, or when LIDS_GLOBAL is disabled so tcpserver can bind to port 25.

```
# setup
/sbin/lidsadm -A -o /var/qmail                           -j READ
/sbin/lidsadm -A -s /usr/local/bin/multilog \
                 -o /var/log/qmail                       -j WRITE
/sbin/lidsadm -A -s /usr/local/bin/svc \
                 -o /var/qmail/supervise                 -j WRITE

# queue access
#
/sbin/lidsadm -A -s /var/qmail/bin/qmail-inject \
                 -o /var/qmail/queue                     -j WRITE
/sbin/lidsadm -A -s /var/qmail/bin/qmail-rspawn \
                 -o /var/qmail/queue                     -j WRITE
/sbin/lidsadm -A -s /var/qmail/bin/qmail-lspawn \
```

```
                -o /var/qmail/queue                    -j WRITE
/sbin/lidsadm -A -s /var/qmail/bin/qmail-queue \
                -o /var/qmail/queue                    -j WRITE
/sbin/lidsadm -A -s /var/qmail/bin/qmail-clean \
                -o /var/qmail/queue                    -j WRITE
/sbin/lidsadm -A -s /var/qmail/bin/qmail-send \
                -o /var/qmail/queue                    -j WRITE
/sbin/lidsadm -A -s /var/qmail/bin/qmail-remote \
                -o /var/qmail/queue                    -j WRITE

# Access to local mail boxes
/sbin/lidsadm -A -s /var/qmail/bin/qmail-lspawn \
                -t -o CAP_SETUID                       -j INHERIT
/sbin/lidsadm -A -s /var/qmail/bin/qmail-lspawn \
                -t -o CAP_SETGID                       -j INHERIT
/sbin/lidsadm -A -s /var/qmail/bin/qmail-lspawn \
                -t -o CAP_DAC_OVERRIDE                 -j INHERIT
/sbin/lidsadm -A -s /var/qmail/bin/qmail-lspawn \
                -t -o CAP_DAC_READ_SEARCH              -j INHERIT


# Remote delivery
/sbin/lidsadm -A -s /var/qmail/bin/qmail-rspawn \
                -t -o CAP_NET_BIND_SERVICE             -j INHERIT

# supervise

/sbin/lidsadm -A -s /usr/local/bin/supervise \
                -o /var/qmail/supervise/qmail-smtpd/supervise     -j WRITE
/sbin/lidsadm -A -s /usr/local/bin/supervise \
                -o /var/qmail/supervise/qmail-smtpd/log/supervise -j WRITE
/sbin/lidsadm -A -s /usr/local/bin/supervise \
                -o /var/qmail/supervise/qmail-send/supervise      -j WRITE
/sbin/lidsadm -A -s /usr/local/bin/supervise \
                -o /var/qmail/supervise/qmail-send/log/supervise  -j WRITE
```

6.4 dnscache & tinydns (djbdns)

The following ACLs were written for a djbdns setup based on Jeremy Rauch's *Installing djbdns (DNScache) for Name Service* parts *1* & *2*. With this configuration, dnscache and tinydns must be started prior to sealing the kernel, or when LIDS_GLOBAL is disabled so they can bind to port 53.

```
# dnscache
#
/sbin/lidsadm -A -o /var/dnscache                       -j READ
/sbin/lidsadm -A -s /usr/local/bin/supervise \
                -o /var/dnscache/dnscache/supervise     -j WRITE
/sbin/lidsadm -A -s /usr/local/bin/supervise \
                -o /var/dnscache/dnscache/log/supervise -j WRITE
/sbin/lidsadm -A -s /usr/local/bin/multilog \
```

```
                -o /var/dnscache/dnscache/log/main      -j WRITE

# tinydns
#
/bin/echo "tinydns"

/sbin/lidsadm -A -s /usr/local/bin/supervise \
                -o /var/dnscache/tinydns/supervise      -j WRITE
/sbin/lidsadm -A -s /usr/local/bin/supervise \
                -o /var/dnscache/tinydns/log/supervise  -j WRITE
/sbin/lidsadm -A -s /usr/local/bin/multilog \
                -o /var/dnscache/tinydns/log/main       -j WRITE
```

6.5 Courier-imap

The following ACLs assume courier-imap was installed into `/usr/local/courier-imap`. With this configuration, courier-imap must be started prior to sealing the kernel, or when LIDS_GLOBAL is disabled so it can bind to port 143.

```
/sbin/lidsadm -A -o /usr/local/courier-imap                     -j READ

/sbin/lidsadm -A -s /usr/local/courier-imap/sbin/imaplogin \
                -o /etc/shadow                                  -j READ

/sbin/lidsadm -A -s /usr/local/courier-imap/libexec/authlib/authpam \
                -o /etc/shadow                                  -j READ

/sbin/lidsadm -A -s /usr/local/courier-imap/libexec/couriertcpd \
                -t -o CAP_SETUID                                -j INHERIT
/sbin/lidsadm -A -s /usr/local/courier-imap/libexec/couriertcpd \
                -t -o CAP_SETGID                                -j INHERIT
/sbin/lidsadm -A -s /usr/local/courier-imap/libexec/couriertcpd \
                -t -o CAP_DAC_OVERRIDE                          -j INHERIT
/sbin/lidsadm -A -s /usr/local/courier-imap/libexec/couriertcpd \
                -t -o CAP_DAC_READ_SEARCH                       -j INHERIT
```

6.6 MySQL

The following ACLs assume MySQL was installed into `/usr/local/mysql`. With this configuration, MySQL must be started prior to sealing the kernel, or when LIDS_GLOBAL is disabled so it can bind to port 3306.

```
/sbin/lidsadm -A -o /usr/local/mysql/var                -j APPEND

/sbin/lidsadm -A -o /usr/local/mysql                    -j READ
/sbin/lidsadm -A -o /usr/local/mysql/libexec            -j READ
/sbin/lidsadm -A -s /usr/local/mysql/libexec/mysqld \
                -o /usr/local/mysql                     -j READ
/sbin/lidsadm -A -s /usr/local/mysql/libexec/mysqld \
                -o /usr/local/mysql/var                 -j WRITE
```

6.7 OpenSSH

The following configuration will work after boot and while LIDS_GLOBAL is on because it gives sshd the CAP_NET_BIND_SERVICE capability.

```
/sbin/lidsadm -A -s /usr/sbin/sshd -o /etc/shadow       -j READ

/sbin/lidsadm -A -o /usr/local/etc/sshd_config          -j DENY
/sbin/lidsadm -A -o /usr/local/etc/ssh_host_key         -j DENY
/sbin/lidsadm -A -o /usr/local/etc/ssh_host_dsa_key     -j DENY

/sbin/lidsadm -A -s /usr/local/sbin/sshd \
                 -o /usr/local/etc/sshd_config          -j READ
/sbin/lidsadm -A -s /usr/local/sbin/sshd \
                 -o /usr/local/etc/ssh_host_key         -j READ
/sbin/lidsadm -A -s /usr/local/sbin/sshd \
                 -o /usr/local/etc/ssh_host_dsa_key     -j READ

/sbin/lidsadm -A -s /usr/local/sbin/sshd \
                 -t -o CAP_SETUID                       -j NO_INHERIT
/sbin/lidsadm -A -s /usr/local/sbin/sshd \
                 -t -o CAP_SETGID                       -j NO_INHERIT
/sbin/lidsadm -A -s /usr/local/sbin/sshd \
                 -t -o CAP_NET_BIND_SERVICE             -j NO_INHERIT

/sbin/lidsadm -A -s /usr/local/sbin/sshd \
                 -o /var/log/wtmp                       -j WRITE
/sbin/lidsadm -A -s /usr/local/sbin/sshd \
                 -o /var/log/lastlog                    -j WRITE
```

6.8 OpenLDAP (slapd)

The following configuration will work after boot and while LIDS_GLOBAL is on because it gives slapd the CAP_NET_BIND_SERVICE capability.

```
/sbin/lidsadm -A -s /usr/local/libexec/slapd \
                 -o /usr/local/ldapdb                   -j WRITE
/sbin/lidsadm -A -s /usr/local/libexec/slapd \
                 -t -o CAP_NET_BIND_SERVICE             -j INHERIT
/sbin/lidsadm -A -s /usr/local/libexec/slapd \
                 -t -o CAP_INIT_KILL                    -j INHERIT
/sbin/lidsadm -A -s /usr/local/libexec/slapd \
                 -t -o CAP_SYS_MODULE                   -j INHERIT
```

6.9 Port Sentry

The following configuration will work after boot and while LIDS_GLOBAL is on because it gives portsentry the CAP_NET_BIND_SERVICE capability.

```
/sbin/lidsadm -A -s /usr/local/psionic/portsentry/portsentry \
                 -o /usr/local/psionic/portsentry              -j WRITE
/sbin/lidsadm -A -s /usr/local/psionic/portsentry/portsentry \
                 -o /var/log                                   -j WRITE
/sbin/lidsadm -A -s /usr/local/psionic/portsentry/portsentry \
                 -t -o CAP_NET_BIND_SERVICE                    -j INHERIT
```

7 LIDS Technical

7.1 Will LIDS work with a file system other than ext2?

Yes. To quote LIDS co-author Philippe Biondi:

> "LIDS works on top of the VFS layer, so that it can handle every fs linux supports."

7.2 Will LIDS run on an SMP system?

There have been problems reported with SMP systems running LIDS. Many of the problems have been fixed, so it is recommended that you try out the latest version and see for yourself. Xie and Philippe are very dedicated to fixing any such problems, so please make sure to report any to the LIDS mailing list.

7.3 Will LIDS coexist with Solar Designer's Openwall patch?

Yes. If you apply both the LIDS and Openwall patches yourself, one of the hunks will fail (as of release 0.9.11 for kernel 2.2.18). It is a minor error that won't affect your system security. However, if you don't like the error, you can visit *http://root-it.be/community/lids* and download the combined LIDS + Openwall patch. Wim Vandersmissen was nice enough to combine the patches and fix the error for us. Wim also offers several other combo patches on his site that include LIDS.

7.4 Will LIDS run on non-Intel hardware?

I'm not aware of any confirmed success stories on other hardware platforms. If you get LIDS to work on another architecture, be sure to let everyone know of your efforts

7.5 What is the difference between the 0.9.x and 1.0.x versions of LIDS?

LIDS `0.9.x` is for the `2.2.x` Linux kernel, and LIDS `1.0.x` is for the `2.4.x` Linux kernel.