Installing Security Enhanced Linux
and
Pluggable Authentication Modules
on Slackware 9

Authored
and
Maintained
by
Timothy Wood
diyab@diyab.net

Table of Contents

Getting Started - Introduction

This document will help you get Security Enhanced Linux up and running on a Slackware 9 installation. Before we get started please take into account that this should NOT be done on a production box and I will not be held responsible for anything resulting from such actions. This should be done on a fresh installation of Slackware 9 and nothing else.

That being said lets make certain you understand something else. PAM does not need to be installed for SELinux to work. Installing PAM with SELinux requires recompiling anywhere from a small to large amount of software. During the SELinux installation most of the core utilities will be recompiled but things like Postgresql, openldap, etc will not. So what you will need to recompile will depend on what you want to enable PAM in. However if you want PAM now or think you might want PAM in the future then now would be the time to install it since we are already going to be recompiling most of the core system utilities.

Still with me? Good. One more thing I want to clear up and that's the switching of cron daemons. The Vixie Crond that comes with SELinux has been modified to understand user roles and security contexts. Currently I do not know anything about writing code and can not code the required security changes into Dillon's Crond. However since I'm not picky about my cron daemon, as they both do the same thing, I just installed Vixie Crond and removed Dillon's Crond. If you would rather use Dillon's Crond you are free to make the appropriate changes to Dillon's Crond and submit them to the SELinux project for incorporation in the next release. Also if you do not want to use any crond you can ignore installing Vixie Crond. Now, on to the fun part.

Getting Started - What You Need

The first file you need is the latest PAM source tarball. You can get PAM from *http://www.kernel.org/pub/linux/libs/pam/*. The latest as of this writing and the version that I used is 0.77 but you can try other versions if you want. If you have success with other PAM versions please let me know so that I can post them here. A pre-requisite of PAM is cracklib. You can grab cracklib and a dictionary from my website at *http://www.diyab.net/selinux*. More specifically you can grab cracklib from *http://www.diyab.net/selinux/files/cracklib_2.7.tar.gz* and you can grab the dictionary from *http://www.diyab.net/selinux/files/wenglish_2.0.orig.tar.gz*.

The second file that you are going to need is the current SELinux tarball. You can get SELinux files from *http://www.nsa.gov/selinux/index.html*. As of the time of this writing the latest version is 2003071106, which is also the version I used. The easiest thing to do is grab the "everything in one download" tarball which should be named something like *lsm-2.4-selinux-2003071106.tgz*. This will give you an lsm patched kernel source tree, selinux, patched utilities trees, and some nice utilities from tresys.com.

The third file that you need is the iproute2 sources. You can get the latest iproute2 sources from *ftp://ftp.inr.ac.ru/ip-routing/*. As of the time of this writing the newest version is marked as 'try' and the two versions before that are marked as 'do not use' so I used the last stable version which is ss010824. If you happen to try the latest version and it seems ok please let me know.

The fourth and final file that you must have is the modified *rc.inet1* init script which uses `ip` from the iproute2 package instead of `ifconfig` and `route`. You can get the modified script from my website at *http://www.diyab.net/selinux/files/rc.inet1-ip.gz*

Optionally you may want to download the Slackware file contexts and domain policy I have written. When the policy was created everything was given its own place for a reason. For the sake of clarity and order I suggest you stick with typing each change into its respective place, however if

you are in a rush, want the easy route, or wish to keep your changes separate from the stock policy you may try these policy files. Please note that these are compiled from my notes and that I have not actually compiled them into a policy to test them. If you run into any problems using the downloaded policy please contact me and I will help you get them working. Installation of both the download policy and the manual insertion of rules are described in section 4.5 Changing the Policy. You can get the latest version of the download policy from my website at *http://www.diyab.net/selinux/files/slacknine_policy.tar.gz*. Thanks to Carsten Grohmann for this suggestion.

Installing PAM

PAM has one pre-requisite that Slackware does not meet by default and that is the presence of cracklib. Start by untarring the *wenglish_2.0.orig.tar.gz* file and `cd` to the *wenglish-2.0* directory. Now copy `linux.words` to `/usr/share/dict/cracklib_dict` and `cd` back out. Next untar the *cracklib_2.7.tar.gz* file and `cd` into the `cracklib,2.7` directory. Using your favorite editor open up the *Makefile* and edit the DICTPATH variable so that it reflects the location of the dictionary we just installed as */usr/share/dict/cracklib_dict* and then build and install cracklib like so:

```
make all
make install
```

The rest of PAM is quite straightforward to install. The only decision you really have to make is if you want PAM to read either */etc/pam.conf*, */etc/pam.d* or both. By default PAM will only read one or the other so if you want both you have to specify the configure option `--enable-read-both-confs`. The default configuration file that comes with PAM is a pam.conf file, but the SELinux install tries to use */etc/pam.d*. Please note that you are not required to have the */etc/pam.d* directory and that you can use the */etc/pam.conf* if you wish as I will give you enough information to do either one.

Once you have decided which style of configuration you want to use run `./configure` with or without the relevant options. Once `configure` has completed just do make and make install. Now if you have decided to use the */etc/pam.conf* style configuration then you need to copy the *pam.conf* from the conf directory in the root of the PAM source tree to the */etc* directory. If you decided to use the */etc/pam.d* style configuration then you need to edit the *pam.conf* and break it down into service configurations. Each service file is saved in the */etc/pam.d* directory under the name of the PAM service. So for example the PAM configuration for the login service will be */etc/pam.d/login*. Each configuration file contains information about what modules and management groups PAM will use for that service. In the *pam.conf* style configuration each service is denoted by putting the PAM servicename as the first column for each configuration line, but since the servicename is denoted by the filename in the */etc/pam.d* style configuration you have to omit the servicename from the service configuration. It looks like this:

```
su      auth        required    pam_unix.so
```
(su configuration from pam.conf)

```
auth        required    pam_unix.so
```
(su configuration from /etc/pam.d/su)

You can clearly see that the only difference is that first column, so remember to remove that column when copying the service definitions from the *pam.conf*. Once you have your PAM configuration setup you are done with PAM for the moment.

SELinux Setup – Installing iproute2

Untar the iproute2 tarball and `cd` to the iproute2 directory. Using your favorite editor change line 2 in the *Makefile* to read.

```
KERNEL_INCLUDE=/usr/src/lsm-2.4/include
```

Now type `make` to compile `ip` and `tc`. Once the compile has finished copy `ip` and `tc` into the */sbin* directory and then copy the iproute2 configs into the */etc* directory.

```
cp ip/ip /sbin/ip
cp tc/tc /sbin/tc
cp -ax etc/iproute2 /etc
```

Now backup your original *rc.inet1* script and replace it with the modified script you downloaded earlier.

```
cd /etc/rc.d
cp rc.inet1 rc.bak.inet1
chmod 644 rc.bak.inet1
gzip -dc <path to>/rc.inet1-ip.gz > rc.inet1
chmod 755 rc.inet1
```

Now you need to edit the *rc.inet1* script and change the addresses to match your network setup. Please do not use `netconfig` with this script because it will most likely break the script. Also please note that I'm still working on this script and I have no idea if dhcp works or not. If you test dchp with this please let me know the outcome of it. If dhcp does not work I would be more than happy to help you get it working so feel free to contact me.

SELinux Setup – Installing SELinux

For the actual SELinux installation follow the SELinux README starting at Step-by-Step Building and Installing. When you come to step 1 ignore what step 1 says and replace it with:

```
su (if not root)
cd ../lsm-2.4
make bzImage
cp arch/i386/boot/bzImage /boot/vmlinuz-2.4.21-selinux
cp System.map /boot/System.map-2.4.21-selinux
ln -s /boot/System.map-2.4.21-selinux /boot/System.map
make modules_install
cd ../selinux
```

Now go to Step 2 in the README and continue until you reach step 8.

SELinux Setup – Switching cronds

In step 8, compilation and installation of modified utilities, the Vixie cron daemon will be installed. Like the other utilities the Vixie cron daemon has been modified to work with selinux security contexts and user roles. Slackware installs Dillon's cron daemon instead of Vixie cron so you will need to remove the currently installed cron before building and installing the modified utilities. To do this login or `su` to root and run `/sbin/pkgtool`. Select the menu option labeled

```
Remove     Remove packages that are currently installed
```

then scroll down until you see the crond package which should read something like this

```
dcron-2.3.3-i386-4          dcron (Dillon's Cron daemon)
```

Highlight the entry, press space to select the package for removal and then press enter to remove it. Once the package has been removed make sure that nothing was left behind in the */var/spool/cron* directory. If there are any files or directories in */var/spool/cron* remove them. Also make certain that the following directories exist:

/etc/cron.d
/etc/cron.hourly
/etc/cron.daily
/etc/cron.weekly
/etc/cron.monthly

Now go ahead and build and install the utilities.

```
cd utils
make
su (if not root)
make install
cd ..
```

Next you have to change two lines in */etc/rc.d/rc.M* using your favorite editor.

change line 139 to

```
# Start crond ( Vixie crond):
```

and change line 143 to

```
/usr/sbin/crond
```

To finish off the cron switch you need to create your crontab. Using your favorite editor put the following text into */etc/crontab*.

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
HOME=/

47 * * * * root /usr/bin/run-parts /etc/cron.hourly
```

```
40 4 * * * root /usr/bin/run-parts /etc/cron.daily
30 4 * * 0 root /usr/bin/run-parts /etc/cron.weekly
20 4 1 * * root /usr/bin/run-parts /etc/cron.monthly
```

SELinux Setup – Openssh Linking Fix

The openssh package included in the userland utilities attempts to link to *libbsd.a*. Slackware installs the bsd compatibility library as libbsd-compat.a instead so all you need to do create a link so that openssh will be able to find the library. Create your link like so:

```
cd /usr/lib
ln -s /usr/lib/libbsd-compat.a /usr/lib/libbsd.a
```

And that's it!

SELinux Setup – Fixing su, newrole and run_init PAM Configs

If you are using the */etc/pam.conf* style PAM configuration please skip down to part A, otherwise if you are using the */etc/pam.d* style PAM configuration please skip down to part B.

Part A - /etc/pam.conf

Since the `make install` of `newrole` and `run_init` will not modify the PAM configuration unless you are using the */etc/pam.d* style of configuration you will have to type everything in by hand. Using your favorite editor copy the following text into your */etc/pam.conf*.

```
#
# The PAM configuration for SELinux newrole
#
newrole     auth            required       pam_unix.so
newrole     account         required       pam_unix.so
newrole     session         required       pam_unix.so
#
# The PAM configuration for SELinux run_init
#
run_init    auth            required       pam_unix.so
run_init     account        required       pam_unix.so
run_init     session        required       pam_unix.so
```

Part B - /etc/pam.d

Since the `make install` of `newrole` and `run_init` will create service configurations for both programs you should now have two new files in */etc/pam.d*, one being named *newrole* and one named *run_init*. Each file should be identical with a three line comment at the top labeling the file as the `su` configuration and four module configuration lines below that. For the sake of good administration you should change the comment to denote either SELinux newrole or SELinux run_init but it is not required. Then just remove the configuration line for the *pam_wheel.so* module from each file. You should end up with two files like this.

```
#
# PAM configuration for SELinux newrole/run_init
```

```
#
auth            required      pam_unix.so
account         required      pam_unix.so
session         required      pam_unix.so
```

The only difference between the two files now should be the comment and that's only if you changed them to denote which service it is for.

The default configuration for the `su` service requires anyone using `su` to be in the group wheel. This is accomplished with the *pam_wheel.so* module. You can do one of three things.

1. Leave the configuration as is and make certain that anyone who wishes to use `su` is part of the wheel group.
2. Change the group the *pam_wheel.so* module checks for with the group=XXXX module option.
3. Remove the line containing the *pam_wheel.so* definition.

Which way you decide to address the situation is up to you, I just wanted to make you aware so that you do not think something is broke when you try and `su` from a non wheel account and it does not work.

SELinux Setup - Fixing /bin/sh

The init rc scripts, like any proper shell script, use the first line to declare the executing shell for the script. The shell defined in the init scripts is */bin/sh*. On Slackware */bin/sh* is a link to */bin/bash* and not an actual executable. The problem with this is that when the filesystem is labeled */bin/sh* gets labeled as bin_t instead of shell_exec_t. The result is that when the init runs commands from the rc scripts it runs things in the wrong domain. The easiest fix for this is to remove the link and create a copy of bash as */bin/sh*. This will not only fix the init scripts but any other scripts that use */bin/sh* as well. The policy change needed is defined in the next section so all you need to do now is remove the link, copy bash to sh and reset the permissions like so

```
cd /bin
rm /bin/sh
cp /bin/bash /bin/sh
```

Now go to step 9 in the README and continue until you reach step 14.

SELinux Setup – Changing the Policy

You should now be at step 14. Ignore step 14 in the README and replace it with this:

```
cd setfiles
make
su (if not already root)
make install
cd ..

cd /etc/security/selinux/src/policy
```

If you plan on manually typing each policy change you may continue at part A. If you have the download policy and wish to use that please continue at part B.

Part A – Manual policy changes

Now using your favorite editor (such as vi, emacs, pico, etc.) make the following changes.

In *domains/program/initrc.te* at line 26 add

```
# Read link files in /etc/rc.d
allow initrc_t initrc_exec_t:lnk_file { read };
```

In *domains/program/fsadm.te* at line 101 add

```
# This is for running over a serial console
allow fsadm_t devtty_t:chr_file { read write };
```

In *domains/program/fsadm.te* at line 107 add

```
# Slackware boot scripts run fsck and need this
allow fsadm_t shell_exec_t:file { execute read };
```

In *domains/program/netutils.te* at line 13 add

```
# Create a type for iproute2 confs
type netutils_conf_t, file_type, sysadmfile;
```

In *domains/program/netutils.te* at line 50 add

```
# Allow reading of iproute2 confs
allow netutils_t netutils_conf_t:file { rw_file_perms };
allow netutils_t netutils_conf_t:dir { search getattr };
```

In *domains/program/newrole.te* at line 75 add

```
# Newrole needs to read /proc/self on slackware 9
allow newrole_t proc_t:lnk_file { read };
```

In *domains/program/ssh.te* at line 9 add

```
# Create a type for sshd privsep directory
type sshd_privsep_dir_t, file_type, sysadmfile;
```

In *domains/program/ssh.te* at line 35 add

```
# Access the privsep directory
allow sshd_t sshd_privsep_dir_t:dir { getattr search };
```

In *file_contexts/types.fc* at line 98 add

```
/bin/sh                --    system_u:object_r:shell_exec_t
```

In *file_contexts/types.fc* at line 100 add

```
/bin/ksh              --    system_u:object_r:shell_exec_t
```

In *file_contexts/types.fc* at line 224 add

```
/usr/i386-slackware-linux/lib.*\.so.*     system_u:object_r:shlib_t
```

In *file_contexts/program/initrc.fc* change the line that reads

```
/etc/rc\.d/rc        system_u:object_r:initrc_exec_t
```

to

```
/etc/rc\.d/.*     system_u:object_r:initrc_exec_t
```

and if you want you can remove the two lines that read

```
/etc/rc\.d/rc\.sysinit     system_u:object_r:initrc_exec_t
/etc/rc\.d/rc\.local      system_u:object_r:initrc_exec_t
```

as the first line that you changed encompasses those two files as well.

In *file_contexts/program/netutils.fc* at line 6 add

```
/etc/iproute2(/.*)?     system_u:object_r:netutils_conf_t
```

In *file_contexts/program/ssh.fc* at line 9 add

```
/var/empty     system_u:object_r:sshd_privsep_dir_t
```

Part B – Download policy

Installation of the download policy is simple. The policy tarball contains two files, the file contexts changes *slacknine.fc* and the domain policy changes *slacknine.te*. Untar the policy files and copy each file to the proper place in the policy source tree like so

```
cd <directory containing slacknine_policy.tar.gz>
tar -xzvf slacknine_policy.tar.gz
cp slacknine.fc /etc/security/selinux/src/policy/file_contexts/program
cp slacknine.te /etc/security/selinux/src/policy/domains/program
```

Now we need to recompile the policy to make sure that there are no errors in your changes and also to install the new policy. Make sure you are still in the */etc/security/selinux/src/policy* directory by typing `pwd` and `cd` to there if you are not. Once you are in the policy directory run

```
make install
```

If there are no errors you should see a line that reads

```
/usr/local/selinux/bin/checkpolicy:   policy configuration loaded
```

Otherwise you might have a problem somewhere. Go back and check to make sure you did not spell anything incorrectly if you got an error. Now you need to initialize the file labels, so run

```
  make reset
```

Once that finishes you have a few changes to make in the rc scripts. We'll call this step 14 1/2 since we're done with step 14 but not moving to step 15 yet. Now using your favorite editor make the following changes.

In */etc/rc.d/rc.6* comment out lines 113 and 114 to match this

```
#echo "Saving random seed from /dev/urandom in /etc/random-seed."
#dd if=/dev/urandom of=/etc/random-seed count=1 bs=512 2> /dev/null
```

In */etc/rc.d/rc.M*

comment out line 18 to match this

```
# /bin/setterm -blank 15
```

comment out lines 78 and 79 to match this

```
# chmod 755 / 2> /dev/null
# chmod 1777 /tmp /var/tmp
```

comment out lines 82, 83, 84 and 85 to match this

```
#if [ -x /sbin/ldconfig ]; then
#  echo "Updating shared library links:  /sbin/ldconfig"
#  /sbin/ldconfig
#fi
```

In */etc/rc.d/rc.modules* comment out lines 26 through 41 to match this

```
# If /usr is mounted and we have 'find', we can try to take a shortcut:
#if [ -x /usr/bin/find -a -e /lib/modules/$RELEASE/modules.dep \
#     -a /lib/modules/$RELEASE/modules.dep -nt /etc/modules.conf ]; then
#  NEWMODS="`/usr/bin/find /lib/modules/$RELEASE -type f -newer /lib/modules/$R$
#  # Only rebuild dependencies if new module(s) are found:
#  if [ ! "" = "$NEWMODS" ]; then
#    echo "New kernel modules have been found in /lib/modules/$RELEASE/:"
#    echo "$NEWMODS"
#    echo "Updating module dependencies for Linux $RELEASE:"
#    /sbin/depmod -a
#  else
#    echo "Module dependencies up to date (no new kernel modules found)."
#  fi
```

SELinux Setup – Lilo Configuration

Now you should be at step 15. Ignore what step 15 in the README says and instead just add the following section into your */etc/lilo.conf* and make sure it is the default.

```
image = /boot/vmlinuz-2.4.21-selinux
   root = <replace with your boot device>
   label = selinux
```

```
read-only
```

If you are not sure what your boot device is you should be able to copy the root line from one of the other boot definitions. To make sure you have this boot entry set as the default you should either have this entry as the first entry in the lilo.conf or you should have specified the `default=` parameter in the global section of the *lilo.conf*. Now run `/sbin/lilo` to reinstall lilo and then continue with step 16 of the README.

What now?

By now you should have a basic SELinux box running on Slackware 9. If you do not or if you have other questions/problems please feel free to email me and I will try to help you. As you learn SELinux a little more you may find that you need to change the policy here and there or do other things. I am trying to keep track of all such changes regarding Slackware so if you have information you would like to share I'd be more than happy to post it on my site with all of my other SELinux Slackware information. Be sure to join the NSA SELinux mailing list which you can find on the NSA website at *www.nsa.gov/selinux/index.html*. Have fun!

Timothy Wood - diyab@diyab.net
http://www.diyab.net/selinux/index.html