LINUX KERNEL & IPTABLES
COMPILE & INSTALLATION PROCEDURE
By Mark E. Donaldson

This procedure describes and details a "from scratch" compilation and installation of the latest Linux kernel version with iptables (to include POM for the 2.4.x kernel or POM-ng for the 2.6.x kernel if desired). As of this date, the latest kernel versions stand at 2.4.25 and 2.6.5, and the latest version of iptables stands at 1.2.9. We will be using the 2.4.25 kernel for the example in this procedure.

1. Download the most recent versions of the following software into the /usr/src directory:

   - **Linux kernel version 2.4.x or 2.6.x from http://www.kernel.org.**
   - **iptables version 1.2.9 from http://www.netfilter.org.**
   - **iptables patch-o-matic from http://www.netfilter.org.**

2. Delete symbolic link to the current kernel source tree, and optionally, remove the current kernel source tree:

   - **rm /usr/src/linux**
   - **rm –Rf /usr/src/linux-2.4.x (optional if needed)**
   - **rm –Rf /usr/src/linux-2.6.x (optional if needed)**

3. Unpack the software downloaded in Step 1 in the /usr/src directory:

   - **cd /usr/src**
   - **bunzip2 linux-2.4.25.tar.bz2**
   - **tar –xvf linux-2.4.25.tar**
   - **bunzip2 iptables-1.2.9.tar.bz2**
   - **tar –xvf iptables-1.2.9.tar**
   - **bunzip2 patch-o-matic.tar.bz2**
   - **tar –xvf patch-o-matic.tar**

4. Create symbolic links to the Linux kernel and iptables source trees:

   - **ln –s /usr/src/linux-2.4.25 /usr/src/linux**
   - **ln –s /usr/src/iptables-1.2.9 /usr/src/iptables**

5. (Optional) Copy the old kernel configuration file to new source directory. With a new kernel version, I recommend using a new default .config since there might be incompatibility issues between the two kernel versions:

   - **cp /boot/vmlinuz.config /usr/src/linux**

6. (Optional) Apply the iptables patch (patch-o-matic). You have three choices: pending (modules slated to be added to the kernel), base (extensions to the kernel), or extra (more extensions).

   - **cd /usr/src/patch-o-matic**
   - **KERNEL_DIR=/usr/src/linux ./runme pending**
   - **(optional) KERNEL_DIR=/usr/src/linux ./runme base**
   - **(optional) KERNEL_DIR=/usr/src/linux ./runme extra**

7. Prepare the kernel source code. Move to the kernel source directory and clean up any existing .o files and old dependencies:

- **cd /usr/src/linux**
- **make mrproper**

8. Configure the new 2.4.25 Kernel. The options available are “make config”, “make menuconfig”, “make xconfig”, and “make gconfig”. To use “xconfig”, the “qt-devel” package must be installed on the system. To use “gconfig”, the “GTK+ 2.0-devel”, “glib2”, and “libglade 2.0” packages must be installed on the system. We will use “make xconfig”:

   - **make xconfig**

   To set the configuration options for iptables, enter the “network options” area of the configuration file. Select all desired options as “modules”. This may also be a good time to enable support for Ipv6, so take the plunge and enable the “Ipv6 protocol (EXPERIMENTAL)”. As we have not yet installed the iptables code into the kernel code, we do not need to be concerned with iptables configuration options on this step. Once complete, save the configuration and exit from the configuration tool.

9. Make (compile) the new 2.4.25 kernel:

   - **make dep**
   - **make bzImage**
   - **make modules**
   - **make modules_install**

10. Make (compile) and install iptables 1.2.9:

    1. **cd /usr/src/iptables**
    2. **make BINDIR=/usr/sbin LIBDIR=/usr/lib MANDIR=/usr/share/man KERNEL_DIR=/usr/src/linux >& iptables-make.log**
    3. **make install BINDIR=/usr/sbin LIBDIR=/usr/lib MANDIR=/usr/share/man KERNEL_DIR=/usr/src/linux >& iptables-install.log**
    4. **(optional for developers) make install-devel**
    5. **/sbin/ldconfig**

11. Copy the new 2.4.25 kernel image to the boot directory, and the new iptables 1.2.9 binaries to their proper directories. I suggest you use the following script to guard against frivolous error:

```
#!/bin/sh
#########################################################################
# copy_kernel.sh
# copies new kernel and support files to boot partition
#########################################################################
# declare a trusted path
PATH=/sbin:/usr/sbin:/usr/local/sbin:/bin:/usr/bin:/usr/local/bin
export PATH

# set script variable name
SCRIPT_NAME="copy_kernel.sh"

#######################
# VARIABLES
#######################
VERSION="-2.4.25"
OLD_VERSION="-2.4.20"
SOURCE_DIR="/usr/src/linux/"
```

```
BOOT_DIR="/boot/"
IMAGE="/usr/src/linux/arch/i386/boot/bzImage"
MAP="System.map"
DOT_CONFIG=".config"
CONFIG="vmlinuz.config"
BUILD_IPTABLES="1"
IPT_VERSION="1.2.9"

#######################
# BEGIN
#######################
echo "Running $SCRIPT_NAME"
echo "Renaming $BOOT_DIR/$MAP to $BOOT_DIR/$MAP.$OLD_VERSION"
mv $BOOT_DIR/$MAP $BOOT_DIR/$MAP.$OLD_VERSION

echo "Renaming $BOOT_DIR/$CONFIG to $BOOT_DIR/$CONFIG.$OLD_VERSION"
mv $BOOT_DIR/$CONFIG $BOOT_DIR/$CONFIG.$OLD_VERSION

echo "Copying $SOURCE_DIR/$MAP to $BOOT_DIR/$MAP.$VERSION"
cp $SOURCE_DIR/$MAP $BOOT_DIR/$MAP.$VERSION

echo "Copying $SOURCE_DIR/$CONFIG to $BOOT_DIR/$CONFIG.$VERSION"
cp $SOURCE_DIR/$CONFIG $BOOT_DIR/$CONFIG.$VERSION

echo "Linking $BOOT_DIR/$MAP.$VERSION to $BOOT_DIR/$MAP"
ln -fs $BOOT_DIR/$MAP.$VERSION $BOOT_DIR/$MAP

echo "Copying new kernel image $IMAGE to boot directory as $BOOT_DIR/vmlinuz$VERSION"
cp $IMAGE $BOOT_DIR/vmlinuz$VERSION

if [ "$BUILD_IPTABLES" = "1" ];
then
        echo "Copying new iptables version $IPT_VERSION executables to /usr/local/sbin directory"
        cp /usr/local/sbin/iptables /usr/sbin
        cp /usr/local/sbin/ip6tables /usr/sbin
        cp /usr/local/sbin/iptables-save /usr/sbin
        cp /usr/local/sbin/iptables-restore /usr/sbin
        cp /usr/local/sbin/ip6tables-save /usr/sbin
        cp /usr/local/sbin/ip6tables-restore /usr/sbin
fi

echo "Completed Running $SCRIPT_NAME"
#######################
# END
#######################
```

12. Update bootloader configuration. Add a section to the lilo.conf (or grub.conf) file to boot the new kernel. We want to leave the old kernel in a bootable condition in case there are problems with the new kernel:

    image=/boot/vmlinuz-2.4.25
    label=/linux-2.4.25
    root=/dev/hda1
    read-only

13. Install the boot loader: **/sbin/lilo**

14. Reboot the machine and test the configuration.