
Admin-HowTo

yet another admin HowTo for allday use

fluffi

Release 0.4

Copyright © 2004, 2005, 2006 fluffi

1. Setting up ssh

Installing the packages:

```
root@donkey:~> apt-get install ssh
```

Allow remote X11 logins, edit /etc/ssh/sshd_config:

```
X11Forwarding yes
```

2. Package administration with sudo

Use a normal user to update/install packages, to reduce root logins.

```
root@donkey:~> apt-get install sudo
root@donkey:~> visudo
```

```
# Cmnd alias specification
Cmnd_Alias APTGET = /usr/bin/apt-get

# User privilege specification
fluffi ALL = NOPASSWD : APTGET
```

Now apt-get can be called without becoming root:

```
fluffi@donkey:~> sudo apt-get update
```

3. Using apt-proxy to save bandwidth

```
fluffi@donkey:~> sudo apt-get install apt-proxy
```

```
root@donkey:~> sudo apt-get install apt-proxy
```

If need be the defaults can be changed in /etc/apt-proxy/apt-proxy-v2.conf The

order of the backends does matter.

```
[debian]
;; The main Debian archive

;; Backend servers, in order of preference
backends =
    http://ftp.de.debian.org/debian
    http://ftp2.de.debian.org/debian

[debian-non-US]
;; Debian debian-non-US archive
;; timeout will be the global value
backends =
    http://ftp.de.debian.org/debian-non-US
    http://ftp2.de.debian.org/debian-non-US

[security]
;; Debian security archive
backends =
    http://security.debian.org/debian-security
    http://ftp2.de.debian.org/debian-security
    http://ftp.de.debian.org/debian-security
```

This proxy can now be used from multiple hosts via entries in the `/etc/apt/sources`

```
# flu: added apt-proxy sources debian/debian-non-US and security
deb http://apt-proxy:9999/debian sarge main contrib non-free
deb http://apt-proxy:9999/debian-non-US sarge/non-US main contrib non-free
deb http://apt-proxy:9999/security sarge/updates main contrib non-free
```

4. Using apt

Choose your default release and set appropriate in `/etc/apt/apt.conf`.

```
APT::Default-Release "sarge";
```

The pin your packages in `/etc/apt/preferences`.

```
Package: *
Pin: release o=Debian,a=sarge
Pin-Priority: 900

Package: *
Pin: release o=Debian,a=testing
Pin-Priority: 300

Package: *
Pin: release o=Debian,a=unstable
Pin-Priority: 100
```

And add the testing/unstable resources to the `/etc/apt/sources`

```
# flu: added testing/unstable sources debian/debian-non-US and security
deb http://apt-proxy:9999/debian testing main contrib non-free
deb http://apt-proxy:9999/debian unstable main contrib non-free
```

```
fluffi@donkey:~> sudo apt-get update
```

5. the kernel

5.1. installing the kernel source

The Linux kernel under Debian

Give a normal user access to the (kernel) sources:

```
root@donkey:~> adduser fluffi src
```

Install the sources and make a kernel image:

```
fluffi@donkey:~> sudo apt-get install kernel-tree-2.6.8
fluffi@donkey:~> cd /usr/src
fluffi@donkey:src> tar xvfj kernel-source-2.6.8.tar.bz2
fluffi@donkey:src> ln -s kernel-source-2.6.8 linux
fluffi@donkey:src> cd linux
```

for linux-tree-2.6.16-14 use:

```
fluffi@donkey:~> sudo apt-get install linux-tree-2.6.16
fluffi@donkey:~> cd /usr/src
fluffi@donkey:src> tar xvfj linux-source-2.6.16.tar.bz2
fluffi@donkey:src> ln -s linux-source-2.6.16 linux
fluffi@donkey:src> cd linux
```

If a formerly created config exists, the `.config` can be used as a base configuration.

```
fluffi@donkey:linux> cp /boot/config-2.6.8-flavour .config
```

5.2. configuring and compiling the new kernel

We configure the kernel via text menu because there is no X11 available yet. If configured already we can use the more comfortable `make xconfig` command. To use the `menuconfig` we need the `ncurses` package.

```
fluffi@donkey:linux> sudo apt-get install libncurses5-dev fakeroot
fluffi@donkey:linux> sudo apt-get install kernel-package module-assistant
fluffi@donkey:linux> make menuconfig
fluffi@donkey:linux> make-kpkg clean
```

When done with the configuration: ready, steady, go:

```
fluffi@donkey:linux> fakeroot make-kpkg --append-to-version "-14-686-donkey" \
--revision 2.6.16 --initrd kernel-image
```

5.3. Using the Debian module assistant

```
fluffi@donkey:~> sudo apt-get install module-assistant
fluffi@donkey:~> module-assistant -kvers-list 2.6.16-14-686-donkey prepare
```

5.4. configuring and compiling the NVidia kernel modules

To download the nvidia kernel sources debian uses the tool 'wget' so we should install it. It's a useful little proggy anyway ;)

```
fluffi@donkey:~> sudo apt-get install module-assistant
fluffi@donkey:~> sudo apt-get install nvidia-kernel-source
```

```
fluffi@donkey:~> cd /usr/src
fluffi@donkey:src> tar xvfz nvidia-kernel-src.tar.gz
fluffi@donkey:src> cd /usr/src/linux
fluffi@donkey:linux> fakeroot make-kpkg --append-to-version "-14-686-donkey"
--revision 2.6.16 modules_image
```

5.5. configuring and compiling the madWiFi kernel modules

The madWiFi kernel sources are available as debian packages.

```
# flu: added madWiFi
deb ftp://debian.marlow.dk/ sid madwifi
```

```
fluffi@donkey:~> sudo apt-get update
fluffi@donkey:~> sudo apt-get install madwifi-source madwifi-tools
```

```
root@donkey:~> cd /usr/src/linux
root@donkey:linux> fakeroot make-kpkg --append-to-version "-14-donkey-686"
--revision 2.6.16 modules_image
```

5.6. installing new kernel and modules

We have to add a line to the lilo configuration file: /etc/lilo.conf.

```
fluffi@donkey:~> sudo apt-get install module-init-tools initramfs-tools initrd-tools udev
root@donkey:~> cd /usr/src
root@donkey:src> dpkg -i *.deb
```

If running lilo we add the following to `/etc/lilo.conf` and rerun lilo.

```
# flu: added a new lilo section for 2.6.16-14 kernel
image=/boot/vmlinuz-2.6.16-14
    initrd=/boot/initrd-2.6.16-14.img
    label=2.6.16-14
    read-only
```

```
root@donkey:src> lilo
```

6. Apache 2

ToDo: Apache return codes: 404, 200, 304
ToDo: Quiet your Debian Apache....apache talks too much.

6.1. installing apache

```
fluffi@donkey:~> sudo apt-get install apache2-mpm-worker
```

6.2. enabling per-directory configuration files

we can allow the per-directory authentication in the main configuration file `/etc/apache2/apache2.conf`

```
# flu: enabled .htaccess authentication in /var/www/eressea

Directory "/var/www/eressea"
    AllowOverride AuthConfig
/Directory
```

next we create a passwordfile for the area:

```
root@donkey:~> htpasswd2 -c /etc/apache2/epasswd user
```

now we create a `.htaccess` file in `/var/www/eressea`

```
AuthType Basic
AuthName "Restricted Files"
AuthUserFile /etc/apache2/epasswd
Require user parteiname
```

6.3. enabling cgi-bin

simply add a symbolic link in the mods-enabled directory...

```
root@donkey:~> cd /etc/apache2
root@donkey:~> cd mods-enabled
```

```
root@donkey:~> ln -s /etc/apache2/mods-available/cgi.load .
```

6.4. enabling cgi-bin outside the /cgi-bin/ directory

```
Directory "/var/www/eressea/cgi-bin"  
    Options +ExecCGI  
/Directory  
  
AddHandler cgi-script pl
```

7. Setting up a DHCP server

Installing DHCP on a Debian system:

```
fluffi@donkey:~> sudo apt-get install dhcp3-server
```

Setting up the IP pool for the local computers /etc/dhcp3/dhcpd.conf

```
# flu: option definitions common to all supported networks...  
option domain-name "frogger.wg";  
option domain-name-servers nameserver1,nameserver2;  
  
# flu: local ethernet  
subnet 192.168.13.0 netmask 255.255.255.224 {  
    range 192.168.13.10 192.168.13.20;  
    option routers 192.168.13.1;  
}  
  
# flu: local wireless  
subnet 192.168.0.0 netmask 255.255.255.224 {  
    range 192.168.0.10 192.168.0.20;  
    option routers 192.168.0.1;  
}
```

8. Setting up a subversion server

Installing subversion on a debian system:

```
fluffi@donkey:~> sudo apt-get install subversion-tools libapache2-svn
```

Create an empty subversion repository

```
root@donkey:~> mkdir -p /var/lib/svn/newRepos  
root@donkey:~> svnadmin create /var/lib/svn/newRepos  
root@donkey:~> chown -R www-data:www-data /var/lib/svn/newRepos
```

Use the new repository with apache2: add /

etc/apache2/sites-available/newRepos. This will provide some basic authentication.

```
<Location /svn/howtos/newRepos>

    DAV svn

    SVNPath /var/lib/svn/newRepos

    AuthType Basic
    AuthName "Subversion Repository"
    AuthUserFile /etc/subversion/newRepos.htpasswd

    Require valid-user
</Location>
```

You'll have to create the `/etc/subversion/newRepos.htpasswd` and add the users you'll like to grant access to the repository. Activate the new *site*:

```
root@donkey:~> cd /etc/apache2/sites-enabled
root@donkey:~> ln -s ../sites-available/newRepos .
root@donkey:~> /etc/init.d/apache2 restart
```

Adding the new repository to websvn:

```
fluffi@donkey:~> sudo apt-get install websvn enscript
```

Moving a subversion repository

```
root@donkey:~> svnadmin dump /path/to/repository > aSvn.dump
```

9. Setting up a MySQL server

Installing MySQL on a Debian system:

```
fluffi@donkey:~> sudo apt-get install mysql-server
```

Change the empty root password...

```
fluffi@donkey:~> mysql -u root
mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('newpwd');
mysql> SET PASSWORD FOR 'root'@'host_name' = PASSWORD('newpwd');
```

or delete the anonymous accounts see (Change default privileges):

```
fluffi@donkey:~> mysql -u root
mysql> DELETE FROM mysql.user WHERE User = '';
mysql> FLUSH PRIVILEGES;
```

10. Madwifi

Installing DHCP on a Debian system:

```
fluffi@donkey:~> sudo apt-get install wireless-tools wavemon
```

11. Adding scripts at boot time

Installing custom scripts on a Debian system. e.g. myScript in all default runlevels at priority 25:

```
root@donkey:~> cp mySkript /etc/init.d/  
root@donkey:~> update-rc.d myScript defaults 25  
Adding system startup for /etc/init.d/myScript ...  
/etc/rc0.d/K25blah -> ../init.d/myScript  
/etc/rc1.d/K25blah -> ../init.d/myScript  
/etc/rc6.d/K25blah -> ../init.d/myScript  
/etc/rc2.d/S25blah -> ../init.d/myScript  
/etc/rc3.d/S25blah -> ../init.d/myScript  
/etc/rc4.d/S25blah -> ../init.d/myScript  
/etc/rc5.d/S25blah -> ../init.d/myScript
```

```
log_failure_msg FAILURE TEXT  
log_warning_msg WARNING TEXT  
log_success_msg SUCCESS TEXT
```