Ubuntu/Debian Help Guide

Detailed set-up, and usage instructions for various Ubuntu configurations

Author:
Adam Rees

Version 0.8.1 published April 18, 2019 using LaTeX

Acknowledgements

To all those that have helped me compile this document over the years, thank you.

Adam

(First compiled December 2011)

Contents

List of Tables

List of Figures

Part I

Starting off

Chapter 1

First things first

1.1 Explanation of this Guide

This entire guide assumes that you have a fresh install of Ubuntu. This chapter (Chapter 1) contains a few packages and commands that will be useful for maintaining a server.

Rewrite this

1.2 Basic Commands

1.2.1 Sudo

Write about sudo + apt-get

1.2.2 Apt-get

1.2.3 Shutting Down and Rebooting

1.2.4 Other Commands

Below in table 1.1 are a list of commands and their uses, these are ones that have been useful when dealing with ubuntu's command line.

Command	Description
mtr [IP Address]	Pings the target and shows a tracepath
firefox &	Runs a process in the terminals background (So you can work with the terminal whilst keeping the process alive)
jobs	Shows all the background processes
fg 1	brings the task (Number 1) to the terminal
Ctrl + z	Key combination detaches from a foreground task, makes it a background task
Ctrl + d	Key combination performs logoff command in a terminal
Ctrl + c	Key combination for closing programs forcefully

Table 1.1: Table of useful commands

1.3 Useful Software

In this section the installation of some useful programs will be covered;

The command below updates the system then installs **Screen, beep, sl, festival, figlet, and cifs-utils**. To save time, the entire batch of software can be written onto one line, installed in one go, and the prompt making sure that we want to install them suppressed by using the following:

```
apt-get install screen beep sl festival figlet cifs-utils -y
```

1.3.1 Screen

Screen allows the processes to be run in a virtual terminal allowing us to close the 'real' terminal without losing the running process.

There are a few useful commands to know when using screen;

Command	Description
screen -S [Screen Name]	New screen with the name [Screen Name]
screen -d -R [Screen Name]	reattaches to a previous screen called [Screen Name]
screen -d -m	creates a new screen but does not attach to it
screen -t [Screen Name]	sets the title for the screen

Table 1.2: Table of screen specific commands

All the commands can be stacked, and running

```
screen -s test beep
```

will open a screen called test and run the command 'beep'.

There are also key shortcuts that are active in screen, there are in the table below (Table 1.3)-

Command	Description
Ctrl + a + c	Starts a new screen (within screen)
Ctrl + a + "	Lists all the active screens (within screen)
Ctrl + d	Terminates a screen
Ctrl + a + d	Detaches a screen

Table 1.3: Table of screen specific keyboard shortcuts

1.3.2 Beep

Beep is a bit of software that gives control of the internal motherboard beep with great precision (frequency, duration repetitions etc...)

To get beep to work you'll have to run the following command;

```
sudo nano /etc/modprobe.d/blacklist.conf
```

uncomment the line about the pcspkr, and restart.

In table 1.4 a list of the options and descriptions can be found.

Option	Commands
-n	starts a new beep in the same command

Table 1.4: Table of commands for beep program

More options needed here

1.3.3 sl

sl is a bit of fun that shows a 'steam locomotive' every time you accidentally type *sl* instead of *ls*. Try out the following;

```
sl -alFe
```

1.3.4 Festival

Festival is a program that will turn text into speech. Quite useful if your ubuntu installation will have external speakers and you would like some messages to be spoken.

For the brits a nice speech pack is the British 16 bit voice, you can install using;

```
1 sudo apt-get install festvox-rablpc16k --yes
```

and to configure festival to work then simply open the config file using nano;

```
1 sudo nano /etc/festival.scm
```

and append the following line;

```
(set! voice_default 'voice_rab_diphone)
```

This line tells festival to use our new voice pack.

To slow down the speech so it sounds more natural we edit the line;

```
(Parameter.set 'Audio_Command "aplay -q -c 1 -t raw -f s16 -r $SR $FILE")
```

to read;

```
(Parameter.set 'Audio_Command "aplay -q -c 1 -t raw -f s16 -r $(($SR*100/100)) $FILE")
```

where 100/100 is the speed expressed as an improper fraction, changing the fraction will change the voice speed.

Running the command is simple the following will work;

```
1 echo "Your Text to be spoken" | festival --tts
```

Where the | character is the pipe command.

1.3.5 figlet

figlet is a program that will take an input and change it into a nice ASCII art image of the word.

The usage is simple, type;

```
1 figlet WORD
```

1.3.6 cifs-utils

cifs-utils is short for *Common Internet File System utilities*. In this guide it will be used to connect to Windows or Samba shares. See section 2.1 for more information.

1.4 Nano

Nano is a built in text editor that will be used throughout this guide, its usage is simple, see below;

```
1 nano FILE
```

Depending on who owns the file you may need root privileges to open it.

There are several options when using nano, hit Ctrl+g for the help menu within nano. Remember although linux is case sensitive, the short-cuts shown in the help page are all lower-case. In table 1.5 we can see some of the more useful short-cuts to remember.

Shortcut	Description
Ctrl + x	Save and Exit
Ctrl + o	Save Only
Ctrl + w	Find a Word
Super + x	Hide the help bar

Table 1.5: Table of useful nano shortcuts

Root privileges can be obtained by using sudo, see subsection 1.2.1

The 'Super' key is the windows key on windows keyboards

1.5 Less

Less is a useful program for viewing large files. The syntax is simple;

```
1 less FILENAME
```

Once inside the less environment there are a few commands that can be used, these are shown in table 1.6;

Command	Description
v	Edit the current file
UP	moves up a line
DOWN	moves down a line
: + e FILENAME	Examine another file
: + n	Examine the next file
: + p	Examine the previous file
?	Display help

Table 1.6: Table of key-commands for LESS command

1.6 Tail

A useful command is **tail** which will output the last 10 (by default) lines of a file. Using the following command the last 10 lines of the file called FILENAME can be seen

```
1 tail FILENAME
```

However if the last 25 lines of this file needs to be seen we can simply use:

```
1 tail -n 25 FILENAME
```

This is a useful tool for looking at files that are constantly being written to, for example custom log files that monitor a hard drive state. When combined with the watch function, see section 1.10 we can constantly monitor the file without running the command repeatedly. In table 1.7 we can see some of the available commands for tail. In order to look at multiple log files simultaneously another command can be used called multitail, see section 1.7.

Option	Description
-n	Number of lines to print
-f	Follow the file as more is printed

Table 1.7: Options available for the tail function

1.7 Multitail

Multitail is a command that allows watching of multiple log files simultaneously.

1.8 Grep

Grep *(Global Regular Expression Print)* is a useful program for finding lines in a text file that contain a certain phrase or characters. The syntax is quite simple;

```
COMMAND | grep STRING

OR

grep STRING FILENAME
```

where as mentioned previously | is the pipe character.
There are several options defined for grep, these are as follows in table 1.8

Option	Description
-i	case insensitive search
-w	search for words not part-strings
-A n	display n number of lines after string
-B n	display n number of lines before string
-C n	display n/2 number of lines before and after string
-r	recursive search through all files and subdirectories of the current directory
-v	show lines with do NOT contain the string
-c	count the number of matches

Table 1.8: Options for grep command

1.8.1 Using the -C option

Lets say we want to show the line before and after our matched expression, we would type;

```
grep –C 2 STRING FILENAME
```

1.8.2 Using the -r option

This is used for scanning all the files in a directory for the string. First navigate to the directory and then type;

```
grep –r STRING *
```

1.9 Find

The find command is self explanitory, it finds files...

However its use is a little more complicated, below is the basic syntax for finding a file called *FILE.EXT*.

```
find / -name 'FILE.EXT'
```

Where the command will attempt to find the file anywhere on the computer (starting at the root directory '/'). In table 1.9 some of the options can be found.

Option	Description
-name	Performs a case sensitive search for the filename
-iname	Performs a case insensitive search
-size	Only lists files of a certain size (e.g. +100M files larger than 100 Mb)
-atime	Only shows files found with a specified access time in days
-perm	Shows files with certain permissions
-exec	Execute the following command on the files found
-type	Find files of type (Either f for file or d for directory)
-user	Find files owned by a user

Table 1.9: Table showing find command options

Find is a powerful function, below are some of the more advanced examples

1.9.1 Find files in current directory with 777 permisions and change to 666

```
find . -type f -perm 777 -print -exec chmod 666 {} \ ;
```

1.9.2 Find all files modified in the last 30 mins

```
find / -cmin -30
```

1.9.3 Find files that are newer than another file, in the current directory

```
find . -newer 'FILENAME.EXT'
```

1.10 watch

Watch is a function that will run a command repeatedly until stopped by the user. It defaults to running the command every two seconds, and displaying the output from that command. The syntax is simple and can be seen below:

```
watch COMMAND TO BE RUN
```

In table 1.10 we can see the options associated with the watch command.

Option	Description
-n	Interval in seconds to run command
-d	Highlight the differences

Table 1.10: Table showing options for the watch command

More options

for example to run the tail command ever 5 minutes we would type:

```
watch -n 300 tail FILENAME
```

1.11 wget

'wget' is a command line program for downloading files from the internet. It is simple to use, and to download a single file to the current working directory we simply type;

```
1 wget www.fileaddress/file
```

Using this a progress bar will show up, telling you the percentage completed, download speed and time remaining.

We can send the file to a specific output using the flag **-O**, an example is shown below;

```
1 wget –O myfile www.fileaddress/file
```

There are some other options to be used with wget and these are shown in table 1.11.

Option	Description
-O	Set Output file
–limit-rate	Limit download speed
-c	Resume an incomplete download
-b	Download in background
–spider	Test a download link
-i	Download multiple links (See Subsection 1.11.1)

Table 1.11: Table of wget options

1.11.1 Multiple Downloads

Lets say we have a file that contains the following;

```
www.mysite1.com/file
www.mysite1.com/file2
www.mysite2.com/file
www.mysite2.com/file2
```

We can pass this file to wget, and tell it to download the four files using the following syntax;

```
1 wget -i FILENAME
```

Where FILENAME is the name of the file shown above.

1.12 Tar

Tar is a program that is used for archiving and compressing folder/files. There are two components an archive and a compression. Just to clarify, tar does the archiving, and gzip does the compression but tar is the command does both.

First we create a tar archive of our directory *big_folder*. the v flag gives a verbose output, and the f gives the filename.

```
1 tar cvf little_folder.tar big_folder
```

Now this wont really save on space so we can compress it using;

```
1 gzip little_folder.tar
```

This can be done in one step using the **z** flag;

```
1 tar cvfz little_folder.tar.gz big_folder
```

This will archive and compress *big_folder* into *little_folder.tar.gz* the options are listed and described in table 1.12;

The archive can be extracted by using:

```
1 tar xvfz little_folder.tar.gz
```

We can even view to files in the archive without extracting them using;

```
1 tar tvfz little_folder.tar.gz
```

If needed you can extract a single file or directory from the archive by using;

```
1 tar xvfz little_folder.tar /path/to/dir/
```

for a directory, and

```
1 tar xvfz little_folder.tar.gz /path/to/file
```

for a single file.

If you want to add another file to the archive then we can use the following provided that the archive is not compressed (i.e. only a .tar NOT .tar.gz)

```
1 tar rvf little_folder.tar added_file/directory
```

If you have compressed the archive then run the following command first before adding a file to the archive;

```
1 gzip -d little_folder.tar.gz
```

Finally before creating the archive you may wish to know what size it will be after compressing/creating, linux can estimate this for you and print the result in kb using the command below;

```
1 tar -cf - /directory/to/archive/ | wc -c
```

the pipe wc -c will print the size in kb using any of the options discussed in table 1.12

Option	Description
c	Create a tar archive
x	Extract a tar archive
t	View the contents of a tar archive
r	Add a file to a tar archive (Only uncompressed)
v	Verbose (Outputs all the files archived)
f	Filename to follow (After options are passed)
z	Compress the archive through gzip

Table 1.12: Tar command table

1.12.1 Multi-Processor Gzip

The Gzip command that comes natively bundled with linux, is single threaded. This means that compression is slower than it could be.

1.13 Stop any process

Using the command line we can very easily kill a process that is maybe taking up too much room, first run;

```
1 ps aux | grep PROCESS
```

Now this will return a list of all the running processes that have PROCESS in the name along with the PID of that process. Using that PID we can either end the process with;

```
kill -s 15 PID
```

or we can send a SIGKILL to force it to end using;

```
kill -s 9 PID
```

Alternatively 'htop' is a good alternative giving a gui within the command line. See subsection 1.13.1

1.13.1 htop

Expand this section

1.14 Cronjob - Schedule a task

Crontab allows scheduling of tasks, be it a scheduled restart, an automatic update, or a custom script.

To open crontab, the cronjob editor simply type the following;

```
crontab -e
```

If it is the first time you have run this command then it will ask you which editor to use, the best one for new users is *nano* which is explained in section 1.4. Simply type the corresponding number of the editor and press Enter.

You can specify what to run and when, looking at the blurb at the top of the file explains how to set a job to run at a specified time, for example to get the **beep** command to run at 12:45 on the 20 August type;

```
45 12 20 8 * beep
```

From left to right, `Minutes‣Hours‣Day of Month‣Month‣Day of week‣Command`

If you want it to run every day then type;

```
45 12 * * * beep
```

Or every Monday (Days of the week are 0-6, as computers start counting at 0)

```
45 12 * * 0 beep
```

There are a few special commands that let you specify certain times like after a reboot or hourly, these are;

```
@reboot - Run once at startup
@yearly - Run once a year at midnight of Jan 1 (@annually)
@weekly - Run once a week
@daily - Run once a day
@midnight - Same as daily
@hourly - Run once an hour
```

1.14.1 Suppress Crontab email

Sometimes certain crontab jobs will email you if something goes wrong, or just every time something is executed... This can quickly fill your inbox, see the messages chapter (Chapter 3) for more. To suppress this simply append the following to the end of the conjob:

```
2>&1 > /dev/null
```

For example, the beep job from earlier would become,

```
45 12 * * 0 beep 2>&1 > /dev/null
```

Although this particular job doesn't output any mail so doesn't need suppressing.

1.15 Aliases

These are really useful little time savers;

simply take the massively long command that you regularly run i.e. starting the minecraft server, as in section 25;

```
java -Xmx1024m -Xms1024m -jar minecraft_server.jar nogui
```

and replace with an easy to remember short command i.e.

```
mine_start
```

to do this the alias command is used.

To start lets list all the aliases known to the system by typing;

```
alias
```

Now lets create a new alias, type the following;

```
alias mine_start='java -Xmx1024m -Xms1024m -jar minecraft_server.jar
    nogui'
```

Now by running *mine_start* we can start the minecraft server.

One word of warning, alias' are only valid for the session, so next time you reboot your server you'll lose them, **unless** you add it to your .bashrc file. Do this by running the following;

```
sudo nano ~/.bashrc
```

and entering your alias as you did in the command line. To create a alias for all users then add it to the */etc/bashrc* file.

Chapter 2

Advanced

http://superuser.com/questions/437330/how-do-you-add-a-certificate-authority-ca-to-ubuntu Add a CA certificate to the certificate store

In this section you will find all the topics that I have not really needed until later on, or when setting up more advanced servers, i.e. those with RAID, multiple hard drives etc.

2.1 Mounting Windows/Samba Shares

There is sometimes a need to mount an external windows-compatible share on our server. In this section we can see how to do that;

2.1.1 Temporary Mounting

We are going to use *smbclient*, and then *mount* for this. First things first we are going to check if we can see the external server share;

```
1 smbclient -L //SERVERNAME/FOLDER
```

If you are successful you should see a list of the shares available to you.

Now lets actually connect to our share;

```
1 smbclient //SERVERNAME/SHARE -U username
```

You should be prompted for the password, simply enter your password and then, (fingers crossed) you should be presented with the smb environment where you can do various things.

The easiest way to connect to a share using our server is to use the mount command, we type the following;

```
1 sudo mount -t cifs //SERVERNAME/SHARE /MOUNT/POINT -o username=USERNAME,
    noexec
```

This will mount the share as if it were a usb drive or similar. You can then interact with it using the ubuntu file system at $/MOUNT/POINT$

2.1.2 Permanent Mounting

Once you are happy with mounting the share, then we can add it to the fstab and get it to mount automatically, simply add the following line to your fstab file ($/etc/fstab$)

```
//SERVERNAME/SHARE /MOUNT/POINT cifs credentials=CREDFILE 0 0
```

Where CREDFILE is the path to a file containing the username and password in the form;

```
username=USERNAME
password=PASSWORD
```

For a more in-depth explanation of fstab see section 8.1.2.

2.2 Monitoring Computer Physical Properties

So if you have a mission critical server, or just an linux box sat under your desk, you may want to know just how hot it is inside the case. We can do this with the sensors package.

Simply install using:

```
1 sudo apt-get install lm-sensors
```

After this is finished we have to run the set up tool:

```
1 sudo sensors-detect
```

There will be a lot of questions asked, its usually fine to just hit enter to all of them (you may be there a while).

Now once this has been completed, you need to run:

```
1 sudo service kmod start
```

In order to read the changes to the kernel modules, or you could just restart the system!

To read the output from the sensors found you can simply type:

```
1 sensors
```

and it will give you an instantaneous reading of all the sensors available. If you would like to constantly poll the system for its temperatures you can use the watch function as mentioned in section 1.10, chapter 1. The command for watching the sensors output every two seconds would be:

```
1 watch sensors
```


2.3 Symbolic Links and bind mounts

Symbolic links and bind mounts are similar to a short-cut in Windows. From the command line it appears as if it is a file/folder and can be interacted as if the file/folder is located where you have put it. For example, I have a file here;

/ » scripts

but for simplicity I want to link a folder in my home directory to this folder so I can get to it quicker... So I want it here;

home » adam » scripts

2.3.1 Symbolic Links

Symbolic links come in two flavours, soft or hard.[1]

Soft Symbolic Links

Soft links reference the path to another file, behaving as if they are a sign post to the os as to which file to open. To create a soft symbolic link between *orig_file* and *link_file* we type the following;

```
1 ln -s /path/to/orig_file /path/to/link_file
```

[1] For more information I read this site: `http://linuxgazette.net/105/pitcher.html`

Hard Symbolic Links

Hard links reference the actual data, not the path, and so they will share the same permissions and data as the actual file. However hard links have to exist on the same file system as the original file, whereas soft links do not. To create a hard link between the files mentioned in the previous example we use;

```
1 ln /path/to/orig_file /path/to/link_file
```

2.3.2 Bind mounts

Now bind mounts behave in a similar way to the symbolic links, however there are some advantages/disadvantages to both. Read up and decide on which you need. There is an easy way to think of bind mounts as mounting a directory instead of a hard drive (See section 8.1. To create a bind mount we simply type;

```
1 mount --bind /path/to/orig_file path/to/link_file
```

This works but will the link be deleted on system restart. To get a persistent link we can add the following to */etc/fstab*;

```
1 /path/to/orig_file    /path/to/link_file    none    bind    0 0
```

2.4 Software RAID - *INCOMPLETE*

2.5 Unattended Upgrades

Ubuntu servers come packaged with a package based installation/updating system. As wonderful as this is, it may be tedious to type the update commands into the command line interface just to update the system. So there is another package for automatically updating these packages. *Uattended Upgrades* does precisely this. To install this simply type the following:

```
1 sudo apt-get install unattended-upgrades
```

We need to then configure the system to perform the upgrades, use the following commands:

```
1 sudo nano /etc/apt/apt.conf.d/50unattended-upgrades
```

Then uncomment the upgrades that you would like the system to install. *(Comments are put into this file by using '//'*

2.5.1 Notifications

If you have set up *ssmtp* or another Mail Delivery Agent *(MTA)*, like in section 3.4. Then an additional package can be used to send you notifications by email. Simply install the package using the following:

```
1 sudo apt-get install apticron
```

Then edit the configuration file using:

```
1 sudo nano /etc/apticron/apticron.conf
```

Now set the email address at the top.

```
EMAIL="your_email_address"
```

2.6 SysRq Key

The SysRq key, normally located at the top of your keyboard, on the same key as Print Screen. Now if you are on your brand new linux computer and it completely locks up, you can use the SysRq key to sort yourself out. We can 'un-stick' our computer by holding Alt+SysRq then pressing each key in order for a second or two, with a break in-between keys.

In order;

Key	Action
r	Keyboard into **r**aw mode
e	Gracefully **e**nd all processes
i	Kill all processes **i**mmediately
u	Flu**s**h data to disk
s	Remo**u**nts all file systems as read only
b	Re**b**oots your computer

Table 2.1: Table of SysRq keys and meaning

2.7 Automatic log on to command line interface

Important - It is wise to read the security chapter first and understand the ethos described in that chapter. Automatic log on is very convenient, and if your server is for your home then I am sure it will be fine. However bear in mind that once your account is logged in automatically then there is no password authentication stopping anyone from using your server. Although the sudo password will not be given there are plenty of things that can be achieved with out sudo privileges that you would not want happening to your server... *Be Warned*

After disabling the graphical interface we simply edit the configuration file for the first command line interface instance. The config file is stored here */etc/init/tty*1.*conf*. In the last line of the file where the command *'exec'* is called we add the following;

```
1 -a user 1
```

So that the line reads;

```
1 exec /sbin/getty -a [user1] -8 38400 tty1
```

And this will log the user in automatically to the command line interface (cli)

2.8 Disabling Graphical Display for Servers

Now I'm not a big fan of removing the graphical log in because every now and then I like to hook the server up to a monitor and check everything is OK... However to improve the servers performance then disabling it will save a few bytes of RAM... and some CPU!

Run the following commands to disable the graphical display.

```
1 sudo nano /etc/default/grub
```

Uncomment the line

$$GRUB\_TERMINAL = console$$

Then run;

```
1 sudo update-grub
```

2.9 Disable the Case power button

Sometimes you want to disable the power button, just because it is within reach of naughty fingers...
Navigate to;

/etc/acpi

And run the following;

```
1 sudo mv powerbtn.sh powerbtn.sh.orig
2 sudo ln -s /bin/false /etc/acpi/powerbtn.sh
```

When your power button is pressed it calls the */etc/acpi/powerbtn.sh* script. By moving the script and creating a symbolic link to the */bin/false* file, we tell ubuntu to do nothing when the button is pressed.

See Section 2.3 for more information on symbolic links.

2.10 De-fragmentation of Hard Drive

Linux has a few tools built in to de-fragment (defrag) the hard drive, but first we should see if it is needed. Windows users will know that occasionally you will need to defrag as your hard drive fills up, as a linux user it is claimed that you will not need to defrag at all, but it always better to check and run it if need be. Run the following as super user to see how fragmented our hard drive is:

```
1 e4defrag -c /
```

This command is for a ext4 file system. Once the command has completed it will show you how fragmented the hard drive is by giving a fragmentation score. Using the legend at the bottom of the output, determine if you would benefit from a defrag procedure. If so run the following as root:

```
1 efdefrag /
```

For some background information visit this site: ***http://jsmylinux.no-ip.org/performance/using-e4defrag/***

Chapter 3

Messages

In this chapter all the messaging systems built into ubuntu will be detailed.

3.1 System Internal Messages

There is a way to send messages to anyone logged in, this can be the contents of a file, or simply what you type. See below for the different options;

```
1 echo "Your text here" | wall
```

This command takes any standard output, for example echo, and then pipes it through the wall command to broadcast to all users.

```
1 write USER tty1
```

Where as this command will connect to a user and display your messages to them. It will start an environment where anything you type will be available to view by the other user. However you do need to know where that other user will be connected we can find out all the required information by typing the following command;

```
1 w USERNAME
```

3.2 System Mail

Sometimes you'll log into one of the TTY's and see that you have mail just after you log in. Lets read it;

Install mailutils;

```
1 sudo apt-get install mailutils --yes
```

Then to read the mail type;

```
1 mail
```

Now to read mail, simply hit Enter. Deleting the mail simply involves typing d and enter.
To delete all the mail in the inbox, you can simply type;

```
1 delete 1-115
```

Where it will delete mails 1 through 115.

For more help please read: ***http://mailutils.org/manual/html_node/Reading-Mail.html***

3.2.1 Stopping Automatic Mail

Most of the mail that you'll get is outputs from cronjobs, to stop this simply append the following to the line in crontab;

```
>/dev/null 2>&1
```

So an example crontab command with suppressed output would be;

```
1 @reboot ~/scripts/ent >/dev/null 2>&1
```

3.3 Sending an internal mail

To send an email to another user (or to yourself from a script) we simply use the following command;

```
1 echo "Some Mail body" | mail -s "A Subject" [USERNAME]
```

Or if there is a lot of text to write you can do it interactively by using;

```
1 mail -s "Some Subject" [USERNAME]
```

Then type your message in the whitespace below. when you are finished press Ctrl + D.

3.4 Email to another computer

So we can configure the server to email us if anything is wrong, or if some event happens. To do this we need to install **ssmtp** which is a *Mail Transfer Agent* (MTA).

```
1 sudo apt-get update
2 sudo apt-get install ssmtp
```

Once it has finished, we now have to configure it. Type the following:

```
1 sudo nano /etc/ssmtp/ssmtp.conf
```

Make the config file look like the one in Appendix C.

Now to send an email we have to type a command followed by text using the following format: *Note the whitespace between Subject and Hello World*

```
1 ssmtp recipient_email_address
```

```
To: recipient_email_address
From: your_email_address
Subject: subject

Hello World
```

Chapter 4

Server Security

In this chapter some thoughts on general basic server security will be given, for security concerning specific programs see the appropriate chapters.

4.1 Physical Security

This section will concern itself with everything except the operating system and installed software that runs under it.

4.1.1 Boot Order

First when we installed ubuntu, we told our BIOS to boot from a CD or USB. Now if this is the default option all a hacker has to do is insert a live USB or CD and access all your files... Not cool...

Set the default (and only, if possible) boot media to the hard drive.

4.1.2 BIOS password

There is some debate about how effective these are... Yes its true that you can overcome the password by shorting a jumper on the motherboard or taking out the CMOS battery, but by setting one you'll force an attacker to go through another step to access the BIOS/Computer, possibly putting them off.

So in short, it doesn't hurt to set one.

4.1.3 Case security

To overcome the point that anyone can bypass the password with physical access to the motherboard, we can prevent this. Most tower cases come with two hoops on the back that line up, put a padlock in them. That's what those hoops are there for.

4.2 Software Hardening

This section concerns itself with the software that runs under the OS, and the OS itself.

4.2.1 Un-necessary software

If you are running a server that is likely to come under attack (See: any computer that is connected to the internet) do not run anything that doesn't need to be running. i.e. Don't run an apache instance if you are not going to use it, it could be an access point for an attacker to gain control of your system.

4.2.2 root access

It has been said in this document a few times that you shouldn't really enable the root user, and only use the sudo function instead.

4.2.3 Fail2Ban

`fail2ban.org/wiki/index.php/Main_Page` `www.linux-magazine.com/Online/Features/Intrusion-Detectio`
`linuxaria.com/howto/how-to-protect-apache-with-fail2ban`

Fail2Ban is a brilliant piece of software which will monitor your log files looking for suspicious behaviour, then ban those that exhibit it. For example with ssh if someone tries ten different usernames with ten different passwords, fail2ban will recognise this and ban the ip adress before they can correctly guess their way in. The same happens with apache and people looking for *phpmyadmin* etc.

To install type

```
1 sudo apt-get install fail2ban
```

Now we need to set up which services are going to be run, and configure them. Navigate to */etc/fail2ban* then create a new file called *jail.local*. This will allow any changes we make to be persistent between updates.

Open up *jail.conf* and look at the services there, to enable them, write the service name and *enabled* = *true* in the jail.local file, as below:

```
[ssh]
enabled = true
```

Any other configuration that you may need to do can also be included here, i.e. port numbers, or a different amount of allow retries:

```
[ssh]
enabled = true
port = 500
maxretry = 2
```

In appendix D a working *jail.local* file can be seen.

The keen readers will notice that there is an entry in the example *jail.local* that does not correspond with any entry in *jail.conf*. This is a custom entry that corresponds to a new rule that has been placed in the *filter.d* directory. This rule is written in regex form and has been included in appendix E, this rule looks for anyone trying to access a well known apache exploit for example running the phpmyadmin setup script.

4.3 Monitoring - INCOMPLETE

`http://www.binarytides.com/linux-commands-monitor-network/`

```
1 sudo apt-get install iftop nethogs bmon vnstat speedometer
```

4.3.1 iftop

```
1 sudo iftop -n -i eth0
```

4.3.2 nethogs

```
1 sudo nethogs
```

Interface Name	Short
Loopback Interface	lo
eth0	Ethernet Interface 1
eth1	Ethernet Interface 2
virbr0	Virtual Bridge
wlan0	Wi-Fi LAN 0

Table 4.1: List of interfaces shown in linux

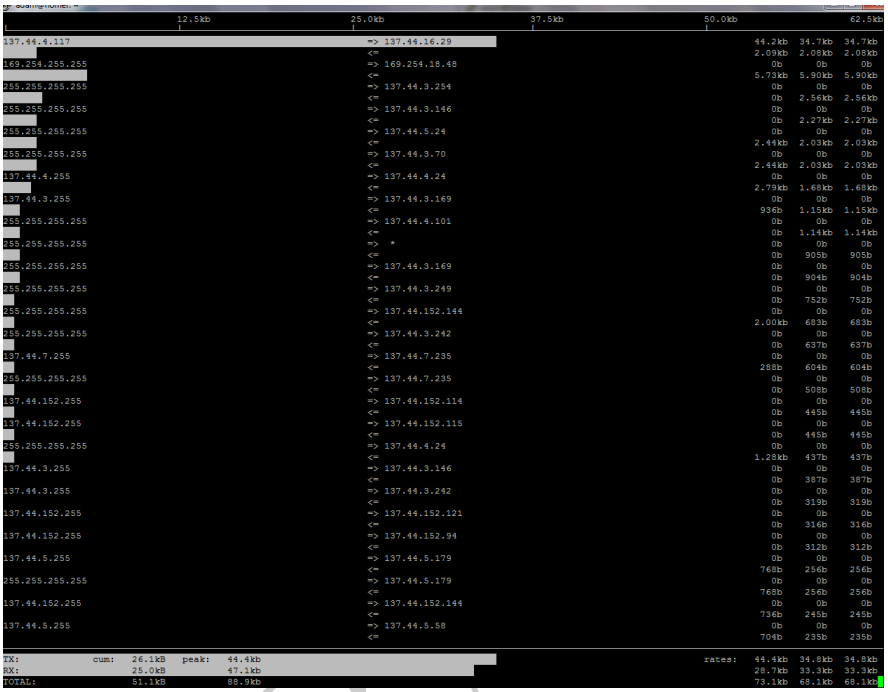

Figure 4.1: iftop terminal output

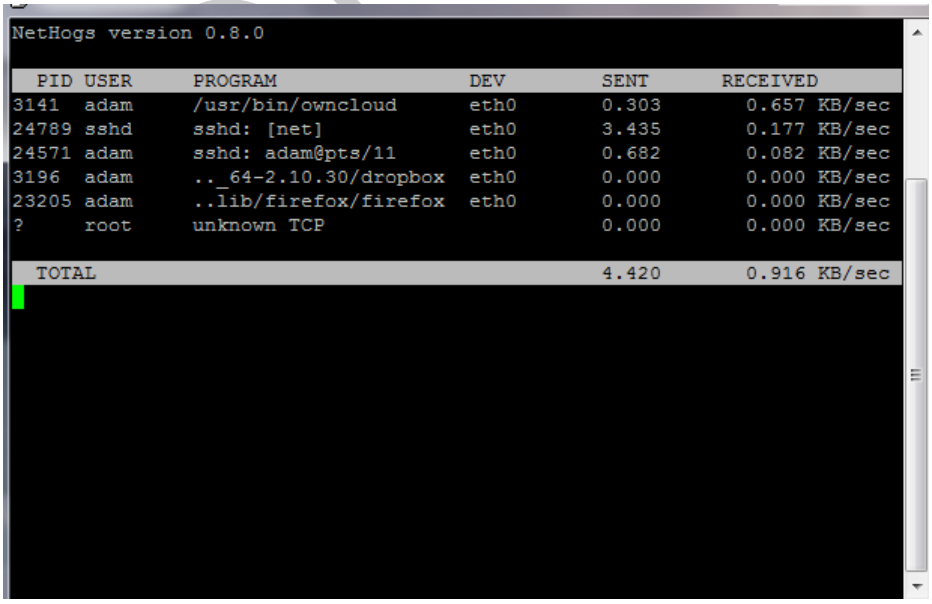

Figure 4.2: nethogs terminal output

4.3.3 bmon

```
1 bmon
```

bmon is used to display graphically the total bps (bytes per second) across all your devices, aswell as in RX (Receiving) and TX (Transmitting) graphs. See figure 4.3 for an example of the output, where the interfaces are shown in table4.1.

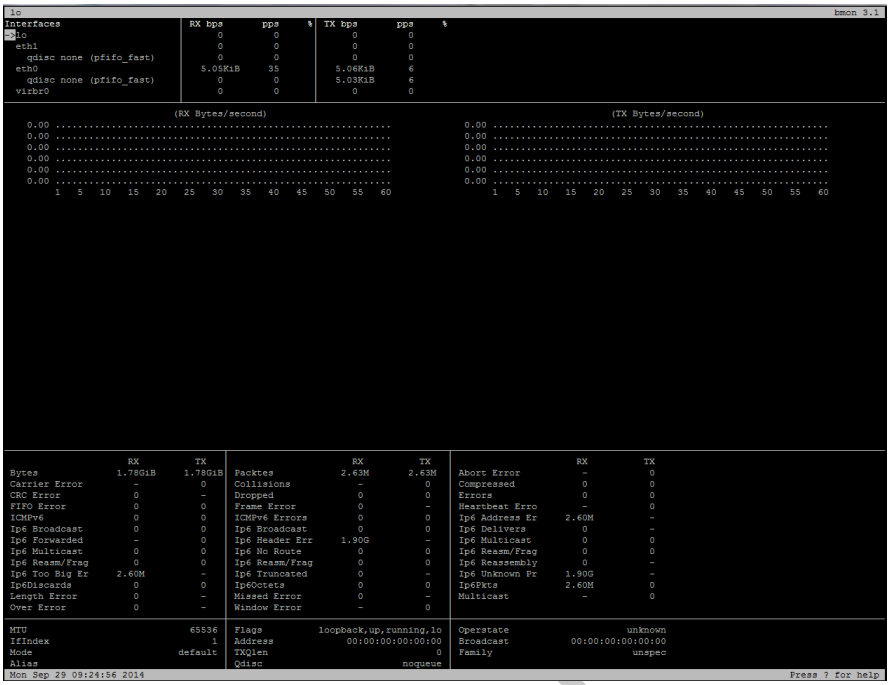

Figure 4.3: Bmon terminal output

4.3.4 vnstat

```
1 sudo vnstat
```

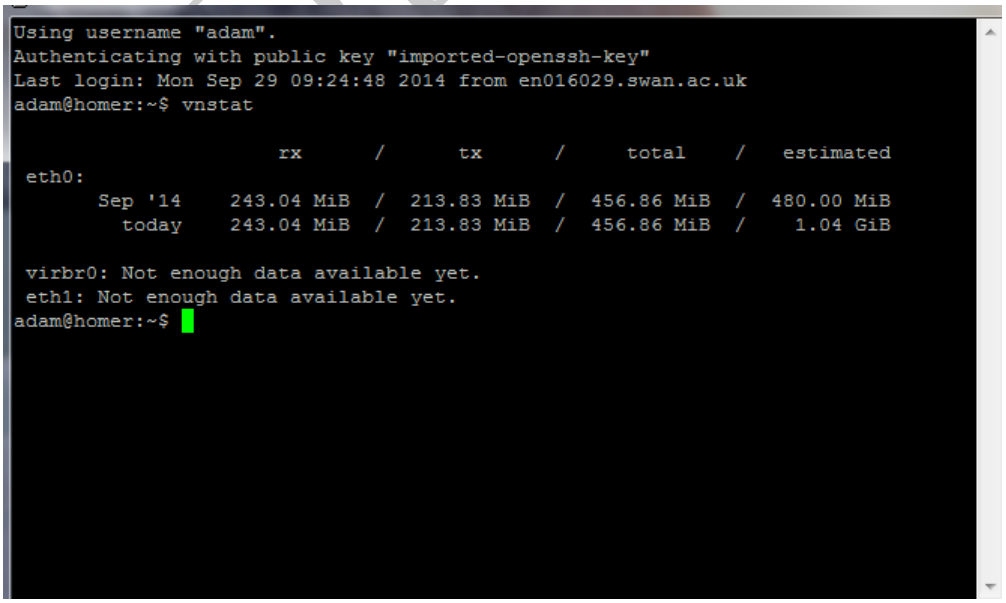

Figure 4.4: vnstat terminal output

4.3.5 speedometer

```
1 sudo speedometer -r eth0 -teth0
```

Figure 4.5: speedometer terminal output

4.4 Backup

Backing up is an important aspect of computer security, if a computer is compromised the best way to overcome it is to start again, see Part VII. If this happens, you'll need a backup of all your data to restore. In this section we see how to generally backup data.

For MySQL we have covered backing up the database into a file in subsection 16.0.3.

4.5 Monitoring Logs

Linux systems log everything that happens, the location of these logs can be found in */var/log/∗*. Table 4.2 shows the log names and what they contain:

log name	log description
messages	General log messages
boot	System boot log
debug	Debugging log messages
auth.log	User login and authentication logs
deamon.log	Running services such as squid, ntpd, and others log
dmesg	Linux kernel ring buffer log
dpkg.log	All binary package logs (including installations)
faillog	User failed login log (Binary File)
karn.log	Kernel log file
lpr.log	Printer log file
mail	All mail server message log files
mysql	MySQL server log file
user.log	All userlevel logs
xorg.0.log	X.org log
apache2/	Apache web server log files directory
lighttpd/	Lighttpd web server log files directory
fsck	fsck command log
apport.log	Application crash report + log file

Table 4.2: Log Name and Description Table

Chapter 5

SSH Server with Public-Private key Authentication

5.1 Installation

SSH allows you to connect to your server from a remote computer, just think if your server sits in the cupboard under the stairs and does not have a monitor or even a keyboard this would be handy. To install the SSH-server on the server, run the following in the terminal;

```
1 sudo apt-get install openssh-server
```

That's it, the server will work out of the box, just make sure that you take a look at the section on securing the ssh server (Section 5.5)

5.2 Display a banner on login

To make your OpenSSH server display the contents of the

/etc/issue.net

file as a pre-login banner, simply add or modify the line:

```
1 Banner /etc/issue.net
```

In the file

/etc/ssh/sshd_config

Rather than editing the banner file, which displays before logging into the ssh connection, you can change the MOTD (Message of the Day) which displays after logging in.

Now you can change the default message, or simply add a message to the end of the file. To add another message, then simply run the command;

```
1 sudo nano /etc/motd.tail
```

The file will be empty by default, and add anything you like, from a simple message to a full blown ASCII art masterpiece. Once completed, exit nano and next time you log in the default MOTD will be shown with your addition at the end.

The default information is derived from some scripts that run in order and are located here;

/etc/update − motd.d/

simply cd to the directory and then ls to show the scripts.

In order to disable one of the scripts from running, then simply change the permissions of the script so that it is no longer executable. Run the command;

```
sudo chmod -x [FILE NAME HERE]
```

You can add your own scripts by simply adding them to this directory and giving the prefixed number that will put it order.

If you decide to enable a script that you disabled earlier then simply run the command;

```
sudo chmod +x [FILE NAME HERE]
```

5.3 Accessing the SSH shell remotely

Windows;

Now on the main computer that you will access the server from install Putty. See Section 5.7. Putty will let us remotely access the command line.

Ubuntu/Linux;

This is slightly simpler, create a script and run it in terminal (Don't forget to enable execution). An example of the script is below;

```
#!/bin/bash

slogin -i [Path to ssh key] user@a.server.com
```

5.4 SSH public-private keys

SSH keys allow authentication between two hosts without the need of a password. SSH key authentication uses two keys a private key and a public key. The SSH connection will need to be secured as anyone that can guess your password will be able to connect to your computer if the port is open to the internet. One option would be not to forward your ports through your routers firewall, however sometimes you may be away from home and want to access the server. Private/Public keys are the way forward.

To generate the keys, from a terminal prompt enter:

```
ssh-keygen -t dsa
```

This will generate the keys using a DSA authentication identity of the user. During the process you will be prompted for a password. Simply hit Enter when prompted to create the key.

By default the public key is saved in the file

/.ssh/id_dsa.pub

and

/.ssh/id_dsa

is the private key. Now copy the *id_dsa* file to the client, then append *id_dsa.pub* to;

/.ssh/authorized_keys

by running the following command;

```
cat id_dsa.pub >> .ssh/authorized_keys
```

However if the computer is a fresh installation, then you can simply change the name of the *id_dsa.pub* file to *authroized_keys*.

Now open the config file, running the command;

```
sudo nano /etc/ssh/sshd_config
```

Change the following settings;

```
ChallengeResponseAuthentication no
PasswordAuthentication no
UsePAM no
```

Reload the ssh server,

```
sudo service ssh restart
```

As long as the private key is copied to all the clients wishing to connect to the ssh server, you should now be able to SSH to the host without being prompted for a password.

5.5 Further Hardening

This step is very vital if your server has access to the internet, after I installed ssh on my server and configured it as above, I had many attempted attacks on my server. Nothing major, some attempted cracking of passwords with incorrect user names. But since I had the public-private keys enabled and passwords disabled there was no way they were getting in. However I was not happy about my access logs filling up with failed connections. since changing the ssh port from the standard 22 to another port I have not had a single attempted connection apart from my own. F.Y.I. the log containing the connection attempts can be found in */var/log/auth.log*

After any changes to the config file, the ssh server must be restarted by running:

```
sudo service ssh restart
```


Port Number

To change the port simply edit the file

$$/etc/ssh/sshd\_config$$

By typing the following;

```
sudo nano /etc/ssh/sshd_config
```

and changing the line

$$Port\ 22$$

to another port number.

Multiple port numbers can be defined in this file by simply adding more lines similar to the one above, i.e.

```
Port 22
Port 222
Port 2222
Port 22222
```

If connecting to the server with a linux box, and use a common port across multiple servers the port can be set on the connecting computer so that the $-p$ flag does not need to be used. This can be set in the $/etc/ssh/ssh_config$ file. **Take special note of the file name**

Protocol 2

Make sure that the line in your config file reads;

```
Protocol 2
```

This forces the server to only use ssh protocol 2 and not protocol 1 which is obsolete.

Limit Users

This is a good way of restricting the users that are allowed to ssh into the server, simply add the following line to your config file;

```
AllowUsers USER USER USER ...
```

Or deny certain users;

```
DenyUsers root USER USER ...
```

The same can be done for groups;

```
AllowGroups GROUP
DenyGroups GROUP
```

The order that openssh processes these directives is as follows;
`DenyUsers‣AllowUsers‣DenyGroups‣AllowGroups`

Idle Logout/Login

This is very useful for closing unused ssh connections, simply add the following two lines to the end of your config file;

```
ClientAliveInterval 180
ClientAliveCountMax 0
```

This tells the server to log out any clients that have been inactive for the last half an hour.

Disable .rhosts

This is set as default, but worth a check;

```
IgnoreRhosts yes
```

Disable root login

This is somewhat a polarising point, but better safe than sorry, don't let the root login through ssh;

```
PermitrootLogin no
```

Updates

Just make sure that you constantly update your ssh server instance, use *apt-get*.

5.6 Running Scripts after Login

There is a file that is executed upon login by a user over ssh, it can be found here;

$$/etc/ssh/sshrc$$

If the file exists, then it will be executed as a script upon login.

Alternatively, you can edit each individual users $/.ssh/rc$, this file executes if it exists and the user logs in through ssh.

5.7 Using Putty (On Windows)

Using Putty is easy, assuming that you downloaded and installed it to your Windows machine. We will have two computers in action now, the 'remote' ubuntu server, and the 'local' windows machine.

Now lets create a new session, and save it for use later...

Type the host name or IP address of your remote ubuntu server, and then enter the port. Now lets give the session a name (I like to use the name of the computer) so that we can refer to it later. For now lets click on save! See the screen-shot in figure 5.1;

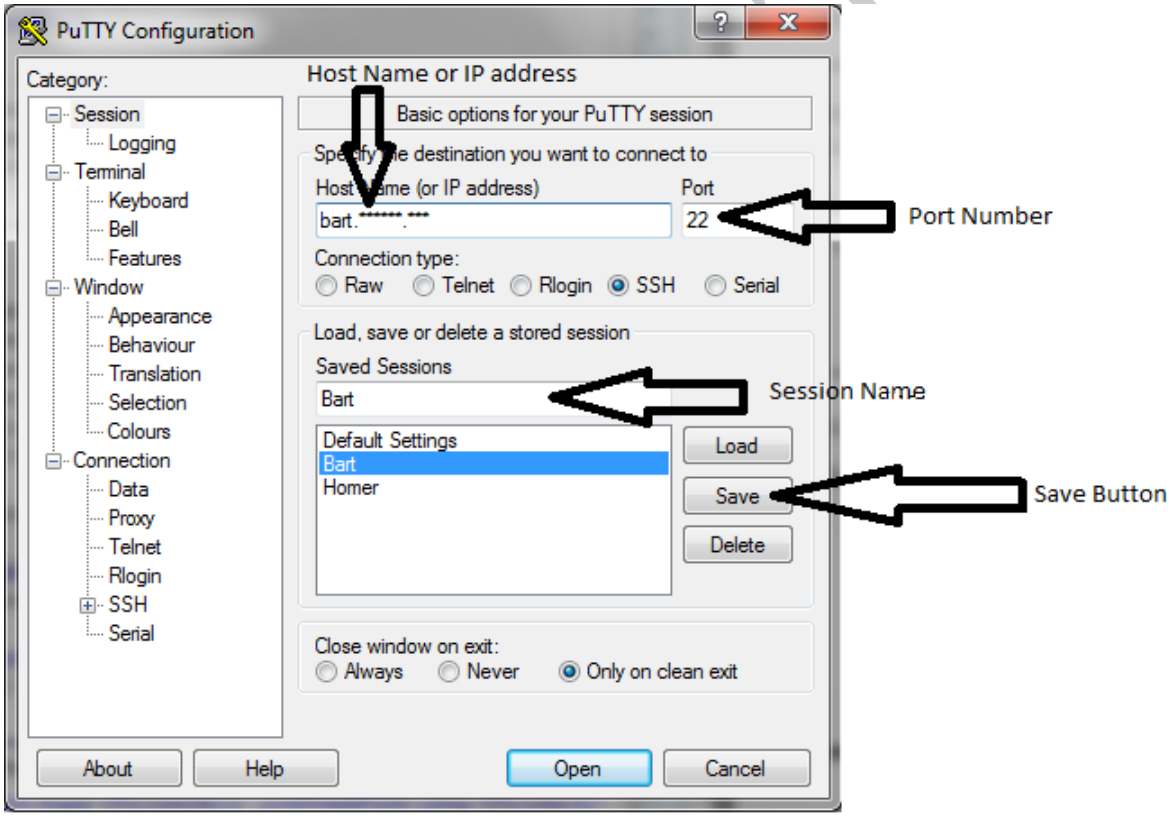

Figure 5.1: Screen-shot of the first page of Putty's interface

Now if you followed the method in Section 5.4, you need to sort out getting your key from your remote machine onto your local machine. (USB Stick, scp, ftp...)

We then tell putty to use this key when connecting to your remote server. Go to the **Auth** page, Connection > SSH > Auth. Then give putty the path to your key. (Note you will have to open it in PuttyGen and convert it to the correct format, by saving it as a private key.)

Now go to **Data**, Connection > Data, and fill in the *auto-login username*.

5.8 Forwarding X11 to Windows

If you need to run a GUI program over ssh then there is a way to do it;

On the remote machine, change a line in the sshd_config file to read;

```
1 X11Forwarding yes
```

Now on the local (windows) machine, you need to download and install *Xming*. Once installed it doesn't need any more configuration.

Back to Putty, we need to set it up to handle X11 forwarding. Go to Connection » SSH » X11, then check the Enable X11 option, and type into X display location *localhost:0*.

Now simply connect to the server and run a GUI program such as xclock to test it is working.

Chapter 6

Virtual Machines - *INCOMPLETE*

Virtual Machines are very useful for separating running processes from the core OS.

There are two types, hardware and virtual. This guide will show the steps required to set up a virtual (software based) machine on a pre existing system. We will assume that the host system will be a command line only system with no GUI.

6.1 Software Virtualisation

Part II

Hardware

Chapter 7

Server Naming Scheme

Servers are usually named haphazardly by picking a name such as Simpsons characters, or Family Guy characters. However I came across a guide on the web [1].

The basic idea is that there are two types of DNS records for each server, an 'A' record, and a 'CNAME' record.

7.1 A Records

The 'A' record is made by choosing a random name from a unique word list, such as the one in appendix F, and using that as a subdomain of your purchased domain. For example the random word is *vampire*, so the A record would be

```
vampire.example.com : 192.168.1.1
```

7.2 CNAME Records

Now the CNAME record can be used for server administrators to know what each server does at a glance, following the system below in table 7.1, an using the abbreviations in tables 7.2 & 7.3.

Purpose	Environment	Location	Domain

Table 7.1: CNAME System

So for example our vampire server from earlier would have a record like:

```
ssh01.prd.cdf.example.com
```

So from the beginning, *ssh* states its a SSH server, *prd* states its a production server, and *cdf* states its located in Cardiff, UK. If you have servers in different countries you can add the two letter country code in front of the three letter location. I use the UN locode list for location symbols, **UN LOCODE Database**.

Abbreviation	Description
dev	Development
tst	Testing
stg	Staging
prd	Production

Table 7.2: Environment Abbreviations

[1] `https://www.mnxsolutions.com/devops/a-proper-server-naming-scheme.html`

Abbreviation	Description
app	Application Server
sql	Database Server
ftp	FTP Server
mta	Mail Server
dns	Name Server
cfg	Configuration Management Server
mon	Monitoring Server
prx	Proxy/Load Balancing
ssh	SSH Jump/Bastion
sto	Storage Server
vcs	Version Control Server
vmm	Virtual Machine Manager
web	Web Server

Table 7.3: Purpose Abbreviations

Chapter 8

Hard-drives

8.1 Mounting partitions/other hard drives

First list all the drives on the computer;

```
1 sudo fdisk -l
```

create a directory somewhere easy, and then mount the partition.
We do this by simply typing;

```
1 sudo mkdir /media/newhd
2 sudo mount /dev/[HDD Name] [SOME PLACE (~/newhd)]
```

where [HDD Name] is the name of the hard drive shown in fdisk, and newhd is the name you want to give it in your Ubuntu file system.
You can now view the files by navigating to the mounted disk, this is found in;

/newhd

8.1.1 Unmounting

Unmounting is quite easy, to unmount the hard drive mounted previously we type;

```
1 sudo umount /dev/[HDD Name]
```

8.1.2 Permanent Mounting

To automatically mount the drive on start-up (Or 'permanent' mounting) we put an entry into the fstab file.

```
1 sudo nano /etc/fstab
```

Now we input an entry into the file using the following protocol;

File System	Mount Point	Type	Options	Dump	Pass

Table 8.1: Fstab protocol

Comments are put in using a '#' key, and it is a good idea to enter a comment saying which drive you are adding.

File System

In this bit you add the drive that you want to mount. For example:

```
/dev/sdb1
```

Or even:

```
UUID:e20c237f-c9cd-4b6e-8c73-5eb8c2f3e85a
```

to find the UUID of a drive, we simply type:

```
1 sudo blkid
```

Mount Point

Now we specify where to mount to drive, for example:

```
/media/hd1
```

Make sure the folder exists first!

Type

This is simply the filesystem type, for example ext3

Options

You can safely leave this as defaults, or google for other options.

Dump & Pass

If you are adding a second drive which will be used for storage, the you can safely just put:

```
0 0
```

8.2 Hard drive settings

Ok, so we've installed the hard drive, and mounted it, we've even made a bash script (see section 12.1) to automatically mount the drives. But now we don't want the hard drives constantly spinning away, wasting power or their precious life time. That's where we need *hpdarm* lets look at what it does;

```
1 sudo hdparm -S 120 /dev/sdb
```

This command will set the spin down time of the 'sdb' hard disk to 10 mins.

```
1 sudo hdparm -C /dev/sdb
```

This command will output the state of the drive, i.e. active/idle, or standby

```
1 sudo hdparm -M 254 /dev/sdb
```

This command sets the AAM (Automatic Acoustic Management) value to 254 (Nosiest/Fastest). The possible values are;

- 0 - AAM off
- 128 - Normal (Manufacturers Recommended)

- 254 - Fastest

```
sudo hdparm -y /dev/sdb
```

This command suspends the current drive putting it into standby mode. **IMPORTANT:** Don't use -Y as this will put the hard drive into off mode which require a restart of the computer.

8.3 (Re)Formatting hard drives

So we have a new hard drive or found an old one and want to reformat it for linux. We are going to use two tools **fdisk** and **mkfs**. First things first we need to check the hard drive is not mounted. (See Section 8.1 for help).

Now lets list the hard drives on our computer, using;

```
sudo fdisk -l
```

So lets say the hard drive we want to format is listed as *sdb*, we now need to start fdisk and point it at the right drive;

```
sudo fdisk /dev/sdb
```

You will now be put into an environment where we can do a lot of damage to the hard drive, but don't let that put you off, its all reversible! (Except recovery of files/folders) Lets delete all the partitions on the drive, lets say there is only one on this example drive, but just repeat the process for all the partitions, type;

```
d 1
```

and then type Enter to delete that partition.

After deleting the partitions, we need to create a new one, simply type;

```
n
```

At the prompt, you need to type p for primary, and then Enter, pressing Enter for the next two questions, letting it get on with its thing.

Now we have a blank drive with one partition, so we need to type w and then Enter to exit fdisk and save the changes.

With all this done we can create the file system that we are going to use. Do some research and find the best for you, but in this example we are going to use ext3;

```
sudo mkfs.ext3 /dev/sdb1
```

Where **sdb1** is the new partition that we created. Now wait for it to finish and then you can use the hard drive as normal. (See Section 8.1 for advice on how to mount and use your hard drive)

8.4 Hard Drive Recovery

If you have a corrupt hard drive, we may be able to recover the data from within Linux. This works with a hard drive from multiple operating systems including Windows, Mac, and Linux itself. We need to install some software to help us, run the following command as sudo:

```
apt-get install --no-install-recommends smartmontools
apt-get install testdisk
```

Part III

User Management

Chapter 9

Creating a new user

9.1 Interactive

This is the easiest way to add an individual user, simply type the following and answer the following prompts;

```
1 adduser USERNAME
```

9.2 Non-Interactive

Lets begin by creating a new user;

```
1 sudo useradd USERNAME
```

Now this will create a user, but without a home directory or password, and with a not very useful shell... Lets change that!

```
1 sudo useradd --shell /bin/bash --home-dir /home/DIRECTORY USERNAME
2 sudo passwd USERNAME
```

So this will create a user with a home directory, and the second command will give you the option of giving them a password.

We can change any of the default options, or view them, see table 9.1 for the options you can change, or type the following to view the default options;

```
1 sudo useradd -D
```

Command	Description
-D	See list of defaults
–home-dir [HOMEDIRECTORY]	Set home directory
–shell [SHELL]	Set the default shell to be used
–expiredate [EXPIREDATE]	Set expiration date of the account

Table 9.1: Table of selected commands for the useradd command

9.3 Bulk User Creation

Ok lets say that there is a lot of users to be added, we can simply use a text file to specify the usernames and passwords etc, then tell linux to create the users.

Firsts things first lets create the text file according to table 9.2;

```
user1:password:1008:1000:User 1:/home/user1:/bin/bash
user2:password:1009:1000:User 2:/home/user2:/bin/bash
user3:password:1010:1000:User 3:/home/user3:/bin/bash
user4:password:1011:1000:User 4:/home/user4:/bin/bash
user5:password:1012:1000:User 5:/home/user5:/bin/bash
```

So to break it down;

username	password	user id	group id	Comment	home directory	shell
user1	password	1008	1000	User 1	/home/user1	/bin/bash

Table 9.2: Table showing the construction of text file for user creation

After creating the file (Let's say its called userfile) we simply run

```
1 sudo newusers USERFILE
```

Chapter 10

Multiple User Security

With multiple users its hard to see what their doing, so you can set their permissions to what ever you want, but what about users with sudo permissions?

10.1 Sudo logging

By default the system logs sudo uses to */var/log/auth.log* but what if we want a dedicated log just for sudo uses? Simply edit the sudo file;

```
1 sudo nano /etc/sudoers
```

Then add the following line;

```
Defaults        logfile=/var/log/sudo.log
```

You can then read the sudo log by simply using;

```
1 sudo cat /var/log/sudo.log
```

10.2 File Permissions

File permissions are pretty straight forward, every file/folder/drive is considered as a file and has permission as follows;

$$OWNER/GROUP/OTHER$$

Setting the file permissions can be done in two ways, either using letters (wrx) or numbers. For ease of use the number method will be covered here.

To show the permissions of files in a directory, navigate to that directory and type;

```
1 ls -l
```

You will be presented with the permissions of those files, the permissions will be presented as a string of letters, these correlate to table 10.1. A typical string for a file may look like;

```
-rwxr-xr-x
```

Where this would mean that the file is not a directory (the first hyphen), the owner can read, write and execute the file, the owners group can only read and execute the file, and all others can read and execute the file.

Now for a directory the permissions vary slightly, taking the last example a directory with the same permissions would look like;

`drwxr-xr-x`

File permissions can be changed using the following command;

```
chmod PERMISSIONS FILENAME
```

To set the permissions we enter a three digit number in the order mentioned earlier (owner/-group/other). The numbers correlate to table 10.1;

System	Read	Write	Execute
Number Based	4	2	1
Letter Based	r	w	x

Table 10.1: File permissions in the two accepted formats used by Ubuntu

So for our test file, we want the owner to be able to do all three; $4+2+1=7$, the group to be able to read and execute; $2+1=3$, and others to just execute; 1. The permissions will be set as follows;

```
chmod 731 testfile
```

Now confirm the change by running;

```
ls -l
```

10.2.1 Permissionless files

A big security flaw is files with no owner or group, as anyone can write and read or execute these files.

Run the following as sudo;

```
sudo find / -xdev \( -nouser -o -nogroup \) -print
```

It will take a while, but it will list all the files with no user or group.

Chapter 11

User Management

So as a server there may be more than one user able to connect to it, and do various things... This chapter will show you how to manage your users.

First things first there are usually only two users when you install a linux system (apart from the software/daemon usernames) and these are **root** and **you**. The you user is the one that you created upon install, and the root user is by default the one you use when using 'sudo'. The root user does not come with a password as by default it is locked... (See section 11.4).

11.1 Listing all the human users of a linux box

It's a simple bit of code, but long;

```
1 cat /etc/passwd | grep /bin/bash
```

Why not add it to your bashrc file as an alias (See section 1.15). The line you need to add is something like;

```
1 alias listusers='cat /etc/passwd | grep /bin/bash'
```

11.2 Forcing password policies

So lets say we run a really secure system where the users have to change their passwords every x number of days... Lets say in 30 days, bob will have to change his password, for this example...

```
1 sudo chage -M 30 bob
```

This sets the password for bob to expire in 30 days.

Now we want to be nice to bob, so lets warn him a week before that his password will expire;

```
1 sudo chage -M 30 -W 7 bob
```

So this means that seven days before the password expires he will have the chance to change his password. After 30 days the account will lock because there will be no password on it.

11.2.1 Change password on first/next login

So lets say you've created a new user, but have given them an easy to remember password *(password1)*...? We need them to change it the first time they log in, luckily this is one lines worth of command;

```
1 sudo chage -d 0 USER
```

11.3 Adding Users to groups

If you need to create a group then use the following;

```
1 sudo groupadd GROUPNAME
```

Verify it exists with;

```
1 grep GROUPNAME /etc/group
```

Now if the user exists, we can use the following to add the user to an existing group;

```
1 sudo usermod -a -G GROUPNAME EXISTINGUSER
```

However if the user does not exist then use the following to create the user and add them to groups;

```
1 sudo useradd -G GROUPNAME NEWUSER
```

11.3.1 Difference between Primary and Secondary Groups

Most of the differences come from file creation, these are explained below...

Primary Groups

When a user creates a new file, the group that the file belongs to is the same as the users primary group. This means that any other members of that group can view the file.

Secondary Groups

When a user creates a file it does not belong to the secondary groups of the user, however any files that do belong to the secondary group can be viewed. See the section of file permissions for more detail (Subsection 10.2).

11.4 Locking/Unlocking Accounts

If you want to set a users account to locked, then all you need to do is run the following;

```
1 sudo passwd -l USERNAME
```

To unlock then simply use;

```
1 sudo passwd -u USERNAME
```

11.5 Changing User Details

There's a comment field in the output of the password file, its shown when you run the command in section 11.1.

Anyway a good thing to include here is the user details such as;

Full Name	Telephone Number	Room Number/Location	Other

We change this by running the code below;

```
1 sudo usermod --comment 'NAME,PHONE,ROOM,OTHER' USERNAME
```

This will then change the comment section in the user details.

11.6 Remotely log-off other users

First we need to see who is logged in type the following;

```
1 w
```

This will list all the users that are currently logged in. Now we can either logout users by their username, or where they are logged in.

11.6.1 By User-name

This is the easy one, simply type;

```
1 sudo pkill -KILL -u USER
```

This will kill the session of USER (Hence the options *-KILL* and *-u*).

11.6.2 By controlling terminal

So looking at my output from w;

```
bob@server:~$ w
 15:41:28 up  4:39,  3 users,  load average: 0.00, 0.01, 0.05
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
bob      tty1                      15:39    4:39m  0.30s  0.15s -bash
bob      pts/0    en016029.swan.ac 15:38    3:24   0.15s  0.00s w
```

We can see that I am logged in to tty1. Lets force a log out;

```
1 sudo pkill -KILL -t tty1
```

Now this forces a logout to anyone connected to tty1.

Part IV

Bash Scripting

Chapter 12

Basics of Bash

12.1 Bash Scripts

These scripts are really simple to set up, simply open a blank file and type the following;

```
1 #!/bin/bash
2
3 [Your set of commands here]
```

Now set the file to executable by running the command below to change permissions;

```
1 sudo chmod +x [FILE NAME]
```

12.1.1 Output

The messages from a custom script can be output to the to a system log (syslog) using;

```
1 logger
```

Usage is simple, to write a line to the *syslog* file we use;

```
1 logger A Message
```

There are other options use;

```
1 man logger
```

to see them.

Chapter 13

Window Control with Bash

In appendix B an example of a bash script that can manipulate the positioning of a window can be seen. This example opens a file every five minutes in the same place every time.

In order to use this script or any of the techniques for setting the window size/position a piece of software called wmctrl is required, run the following as root:

```
1 apt-get install wmctrl
```

Part V

Standard Server Applications

Chapter 14

FTP Server

This is a good way of getting files to your machine without using Dropbox, which does not work unless you are logged in...

vsftpd is an FTP daemon available in Ubuntu. It is easy to install, set up, and maintain. Run the following command:

```
1 sudo apt-get install vsftpd
```

Now there are two ways of configuring the ftp server, Anonymous or User Authenticated.

Anonymous is good when your server is only accessible from within your LAN, this is the default setting when the server is installed. A FTP user is created and the default upload location is

/home/ftp

However I prefer the User Authenticated version in which the user logs in with their own user name on the server and uploads to their home directory. Both installation procedures are shown in this section.

14.1 User Authenticated

To enable the server to authenticate the FTP users before allowing access, then open

/etc/vsftpd.conf

using nano.

Change the following;

```
1 local_enable=YES
2 write_enable=YES
```

Restart the FTP server by running

```
1 sudo /etc/init.d/vsftpd restart
```

The configuration to this point will allow users of the server to log in and upload files, however to secure the FTP server further we can restrict users to their home directories by uncommenting the line;

```
1 chroot_local_user=YES
```

Or even restrict certain users to their home directory by uncommenting;

```
1 chroot_list_enable=YES
2 chroot_list_file=/etc/vsftpd.chroot_list
```

Running the command;

```
sudo nano /etc/vsftpd.chroot_list
```

Will create the file which restricts users to their home directory, simply add usernames one per line.

Also

/etc/ftpusers

is a list of users that are disallowed FTP access.

The server can be further secured with a SSL certificate, read this web page; Certificates[1]

To enable secure FTP (FTPS) then add the following line to the bottom of

/etc/vsftpd.conf

```
ssl_enable=Yes
```

Notice the key and certificate related options;

```
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
```

14.2 Anonymous

This is the default configuration, and the uploaded files are available in

/home/ftp

However to change the default directory to

/srv/ftp

for example then simply change the FTP users home directory;

```
sudo mkdir /srv/ftp
sudo usermod -d /srv/ftp ftp
```

Then restart the FTP service;

```
sudo /etc/init.d/vsftpd restart
```


[1]For thos who can not click - https://help.ubuntu.com/10.04/serverguide/certificates-and-security.html

Chapter 15

Apache

15.1 Installation

Simply run the command;

```
1 sudo apt-get install apache2
```

The apache config files may need changing, they are located here;

/etc/apache2/..

Opening the *apache2.conf* file, we can see how the documents in the directory relate to each other.

15.2 Initial Set up

Now to just serve one website we can simply add website (http) files to the */var/www* folder.

15.3 Virtual Hosts

If we want our server to host more than one site, we need to enable virtual hosts. This can be accomplished in two ways, either;

- Each site gets its own IP address, therefore an ethernet card/port for each site
- Each site gets its own domain name, with one shared IP

The easiest and cheapest way to set up multiple hosts is the second option.

15.3.1 Setting up multiple hosts

So each site needs its own directory within */var/www/*, for example;

/var/www/example.com

Also the directory needs the permissions **755**, see Subsection 10.2 for how to set this.
Once the directory has been created, we need to tell apache about it;
Copy the default configuration file, renaming it with your site name using the following command;

```
1 sudo cp /etc/apache2/sites-available/default /etc/apache2/sites-available
    /EXAMPLE
```

Now open the config file using nano (See section 1.4)
Editing this file we need to change the following;

- Add:

  ```
  ServerName EXAMPLE.COM
  ```

- Change:

  ```
  DocumentRoot /var/www
  ```

  To:

  ```
  DocumentRoot /var/www/EXAMPLE
  ```

Now with these changes completed, we simply need to activate the site by using;

```
1 sudo a2ensite EXAMPLE.COM
```

Where EXAMPLE.COM is the address of the site you set in *ServerName*
Now just restart apache using;

```
1 sudo service apache2 restart
```

Rinse and Repeat for additional hosts.

15.4 Hardening

15.4.1 Right Users

Install apache2 under a different user and group to any other service. Simply check the */etc/apache2/envvars* file for the user and group that apache runs under.

15.4.2 ServerTokens

When you have an error with your website, apache attempts to help you by stating the version of apache and the OS that it is running on, we don't really want this because it will help a potential attacker, change the following in the *security* file in the */etc/apache2/conf.d* directory;

```
ServerTokens OS
```

To

```
ServerTokens Prod
```

Also comment the line that says;

```
ServerSignature On
```

and uncomment the line that says;

```
ServerSignature Off
```

15.4.3 SSL

```
https://help.ubuntu.com/community/forum/server/apache2/SSL
   https://www.digitalocean.com/community/tutorials/how-to-set-up-multiple-ssl-certificates-or
```

Chapter 16

MySql

Simply run the command;

```
1 sudo apt-get install mysql-server
```

And remember the password and username specified during installation.
Most programs that require the use of MySql will create and manage their own databases. **IM-**

PORTANT Do not run MySql under the same user and group as apache, as an attack on one, could leave the other vulnerable.

`http://www.cyberciti.biz/faq/mysql-command-to-show-list-of-databases-on-server/`
`http://www.bios.niu.edu/johns/bioinform/mysql_commands.htm`

16.0.1 MySQL basic commands

Here are a list of basic commands for MySQL. Note the semicolon at the end of each command.

Command	Description
SHOW DATABASES;	Lists all databases in MySQL
USE [NAME];	Switches to specified database
quit;	Quit from MySQL

Table 16.1: Table of Basic commands for MySQL

16.0.2 Reset mysql root password

Sometime we forget passwords, so it is important we know how to change them, here is a six part solution to change the password of the mysql root user.

Stop MySQL

```
1 sudo service mysql stop
```

Start in safe mode

```
1 sudo mysql_safe --skip-grant-tables &
```

Then change database

```
use mysql;
```

Change Password

Change the password using the following, and putting a new password instead of newpasswordhere below.

```
update user set password=PASSWORD ("newpasswordhere") where User = 'root
    ';
```

flush privileges & quit

```
flush privileges;
quit;
```

Restart MySQL

```
sudo service mysql restart
```

16.0.3 Backup and Restore MySQL databases

`http://webcheatsheet.com/sql/mysql_backup_restore.php`

In this section we will see how to backup and restore the mysql databases.

Backing up

Before logging in to mysql, run the following commands:

```
mysqldump -u [USERNAME] -p [DATABASE NAME] > [BACKUPFILE.sql]
```

Don't put the square brackets in the command, and replace where necessary, you will be asked for your password after running the command.

You can backup multiple databases at once by separating them with a space.

This command will lock out users when it is running, unless you have a very large database, you shouldn't need to worry.

Restoring a Backup

There are two ways to restore, one to add a backup database into a new install (effectively creating a new database with the old data), and one to overwrite an existing database.

To simply add the backup database to mysql where it didn't exist before run the following:

```
mysql -u [USERNAME] -p [DATABASE NAME] < [BACKUPFILE.sql]
```

However if the database does exist but you wish to overwrite the data, simply type:

```
mysqlimport -u [USERNAME] -p [DATABASE NAME] [BACKUPFILE.sql]
```

Chapter 17

phpBB server

install mysql and apache, sections 16 and 15.

Now do the following;

```
1 sudo apt-get install phpbb3
```

you'll be prompted for the mysql root password, not the root users password.

```
1 sudo ln -s /usr/share/phpbb3/www /var/www/phpbb
```

If you get a message about php environment not having support for a database run this;

```
1 sudo /etc/init.d/apache2 restart
```

Now the server should be running, go to an internet browser and go to the address;

```
1 http://{THE SERVER IP}/phpbb
```

The server now needs configuring from the graphical admin control panel using the default user and password given below;

```
username = admin
password = admin
```

Don't forget to change the password and enable the board! (Both in the admin c.p.)

Chapter 18

Samba

18.1 Installation

So we need to install some software before we can use samba to share our files, run the following from the terminal.

```
1 sudo apt-get install samba samba-common-bin
```

Need to finish this

Chapter 19

Mail Server

Figure 19.1 shows the process by which mail is sent and received, and all the different parts involved. This is just a basic schematic and doesn't show the web interface for collecting mails, which is simply an apache server. Not shown in the picture is a wrapper software that interfaces between clamAV/Spam Assassin and Postfix

The red line indicates an outgoing message from the client, and the black line indicates an incoming message. As can be seen all interactions between the mail server and the outside world are handled by postfix, and interactions between the server and client are handled by dovecot. ClamAV and Spamassassin are simply screening tools used by the server to keep the client safe. There are other tools available also these are listed in table 19.1

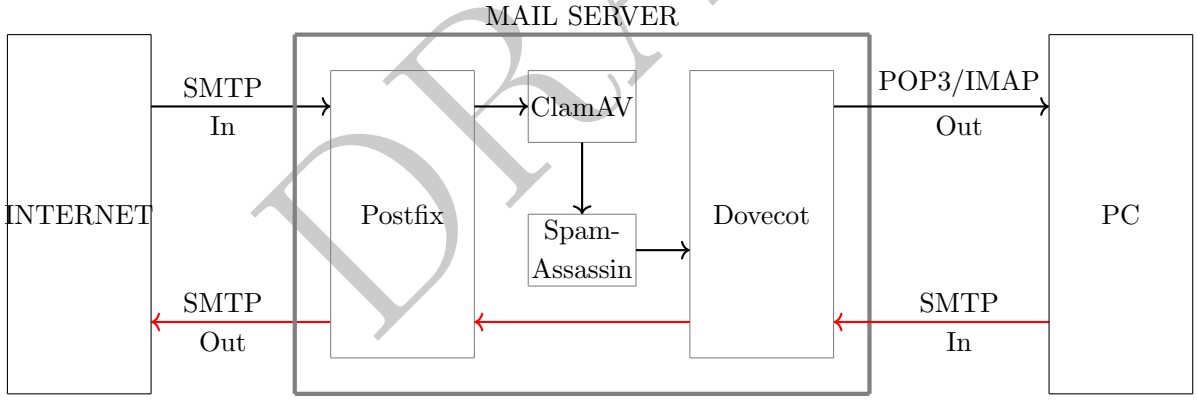

Figure 19.1: Schematic showing mail server interactions

Service	Description
Mailman	Mailing list management program
Scrollout	Anti Spam Software

Table 19.1: Table of other services available to run with a mail server

Resources

Postfix Installation
Postfix tools/commands
Postfix filtering software wrapper Scrollout Anti Spam

Chapter 20

Open Project

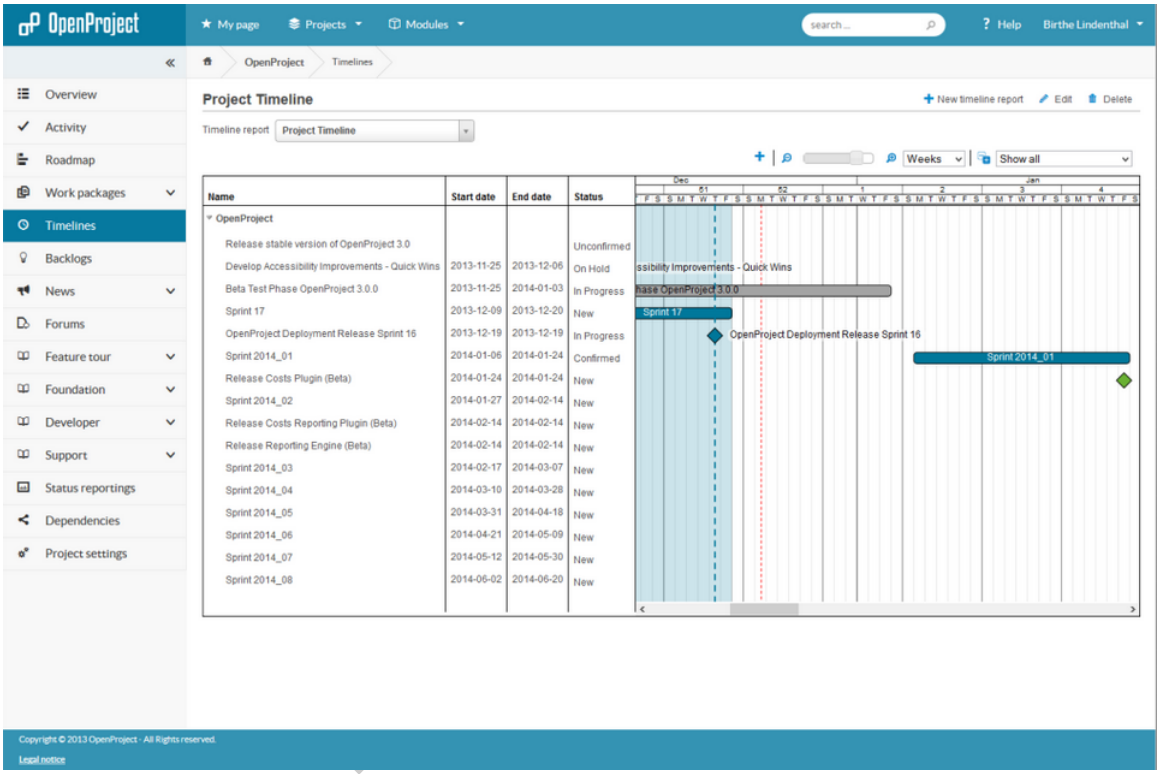

Figure 20.1: OpenProject Web interface

Chapter 21

Media Goblin

Media Goblin is an open source alternative to content sites such as youtube or flicker.

Chapter 22

Zone Minder

```
1 sudo apt-get install tasksel
2
3 sudo tasksel
```

In the prompt select LAMP server, this will install Apache (Chapter 15), MySQL (Chapter 16), and PHP (Sometimes installed by default).

Now choose a password for the MySQL server.

Now update/upgrade, then install zoneminder:

```
1 sudo apt-get install zoneminder
```

Now we must add a delay so that MySQL can start before zoneminder:

```
1 sudo nano /etc/init.d/zoneminder
```

then add the following above $zmfix -a$:

```
sleep 15
```

Now fix a problem in the config file:

```
1 sudo nano /etc/zm/zm.conf
```

and add the following:

```
ZM_VERSION=1.26.5
```

Now link apache to zoneminder:

```
1 ln -s /etc/zm/apache.conf /etc/apache2/conf-enabled/zoneminder.conf
```

restart apache

```
1 sudo service apache2 restart
```

Now we need to install Cambozola

```
1 sudo wget www.zoneminder.com/sites/zoneminder.com/downloads/cambozola.jar
    -O /usr/share/zoneminder/cambozola.jar
```

Chapter 23

Partkeepr

http://www.untacomei.com/wp/centos-6-3-how-to-setup-a-partkeepr-and-phpmyadmin/
http://technologyrealm.blogspot.co.uk/2014/03/partkeepr-installation-on-ubuntu-or.html

Part VI

Game Servers

Chapter 24

Teamspeak Server

24.1 Set-up

The following procedure is good for installing a teamspeak server. After installation, most of the configuration is done using the teamspeak client software. Don't forget to copy the access token for use on your client software as admin.

1. Download the teamspeak server file from the website

2.Unpack the file (remember to replace the file name if it is different

```
tar xfv teamspeak3-server_linux-x86-3.0.2.tar.gz
```

3. Change directory;

```
cd teamspeak3-server_linux-x86
```

4. Now run the binary by typing;

```
./ts3server_linux_x86
```

and stop it using Ctrl+c

5. The server commands are as follows;

```
./ts3server_startscript.sh start
./ts3server_startscript.sh stop
./ts3server_startscript.sh status
```

Find the internal ip address of the server by typing;

```
ip addr
```

Set up a bash file to start, stop or check status of server

Chapter 25

Minecraft Server

25.1 Updating Java

The next set of commands are run with root privileges, run the following code;

In order to ensure that the java environment is correct, run the command;

```
1 java -version
```

Do the following:

```
1 update-java-alternatives -l
```

This will list off all the various Java VMs that are installed.

```
1 # update-java-alternatives -l
2 java-6-openjdk 1061 /usr/lib/jvm/java-6-openjdk
3 java-6-sun 63 /usr/lib/jvm/java-6-sun
```

Next set the proper Java VM to use by typing the following;

```
1 update-java-alternatives -s java-6-sun
```

Now, run

```
1 java -version
```

It should show the same version information as before.

The rest of these commands are run without root privileges.

25.2 Installing Minecraft Server

Now download the server file from the minecraft website, and place it into the home directory

To start the server, you will need to use the following command:

```
1 java -Xmx1024m -Xms1024m -jar minecraft_server.jar nogui
```

Once all the text has finished, it will have created a new world file. Now force the server to save the files and stop.

```
1 save-all
```

which forces the server to save the generated map, then

```
1 stop
```

to shut the server down.

25.3 Stopping and Starting the server

To start the server, first off make sure you are in a screen session by typing

```
screen -list
```

like below:

```
mcserver@mcserver:~$ screen -list
There is a screen on:
 2434.tty1.mcserver          (01/09/2011 12:58:57 PM)          (Attached)
1 Socket in /var/run/screen/S-mcserver.
```

This indicates that you have screen session. If you the console says that no screens are running then type;

```
screen
```

to start one.

Once in the screen session, type in the command shown below.

```
java -Xmx1024m -Xms1024m -jar minecraft_server.jar nogui
```

To disconnect from the screen session, hit Ctrl+a and then the d key, this will drop back to the shell prompt where to exit simply type;

```
exit
```

to logout. The Minecraft Server will continue to run.

To reconnect with the screen session, type in

```
screen -r
```

You will be reconnected to the server and can then perform the following commands:

```
say Server is going down

save-all

stop
```

This tells the Minecraft Server to save, shutdown and exit, and tell all the users that it is doing so.

25.4 Useful commands in MC Server

Table 25.1 contains some of the useful commands for maintaining a minecraft server.

Command	Description
help or ?	shows this message
kick *player*	removes a player from the server
ban *player*	bans a player from the server
pardon *player*	pardons a banned player so that they can connect again
ban-ip *ip*	bans an IP address from the server
pardon-ip *ip*	pardons a banned IP address so that they can connect again
op *player*	turns a player into an op
deop *player*	removes op status from a player
tp *player1 player2*	moves one player to the same location as another player
give *player id* [num]	gives a player a resource
tell *player message*	sends a private message to a player
stop	gracefully stops the server
save-all	forces a server-wide level save
save-off	disables terrain saving
save-on	re-enables terrain saving
list	lists all currently connected players
say *message*	broadcasts a message to all players

Table 25.1: Useful Commands for Minecraft Server

Chapter 26

Simutrans Server - *INCOMPLETE*

26.1 Beginning

So to start we need a subversion client installed, and the build-essential package in order to compile the binaries;

```
1 sudo apt-get install subversion
2 sudo apt-get install build-essential
```

and we download the latest simutrans version from the server by running;

```
1 svn co --username anon -r 4303 svn://tron.homeunix.org/simutrans/
    simutrans/trunk
```

when prompted for a password just hit Enter.

26.2 Compiling

Now go into the trunk directory and copy the config file using the command below;

```
1 cp config.template config.default
```

Now open the new file using;

```
1 sudo nano config.default
```

and uncomment the following

$$BACKEND = posix$$

$$COLOUR\_DEPTH = 0$$

$$OSTYPE = linux$$

$$DEBUG = 3$$

$$OPTIMISE = 1$$

$$WITH\_REVISION = 1$$

Then run the;

```
1 make
```

command. Followed by the;

```
1 strip sim
```

Part VII

Uninstallation

Chapter 27

Basics

In this chapter the methods of uninstalling programs from Ubuntu.

27.1 Program Uninstallation

So there are two ways to uninstall any program such as the ones we installed in this guide.

The first;

```
1 sudo apt-get remove [PROGRAM]
```

and the second;

```
1 sudo apt-get purge [PROGRAM]
```

Now they basically do the same thing, with the exception that the purge command also removes the configuration files associated with the program, freeing up more space. However this means that you have to completely re-configure any software that you have purged if you re-install it.

27.2 Startup Removal

In the Subversion Chapter (Section **??**) we added a new script to the init.d directory and told ubuntu that we want to run that script on startup. So to start we stop the script being executed, and then remove it from the init links.

```
1 sudo chmod -x [SCRIPT]
2 sudo rm [SCRIPT]
3 sudo update-rc.d -f [SCRIPT] remove
```

After this a simple reboot of the system should stop any unwanted server processes.

27.3 Group/User Removal

In certain installations its good practice to create a group or user for the server to run as for security, however when you uninstall said program you need to get rid of the user or group, do this by using one of the following;

```
1 deluser [username]
2 delgroup [groupname]
```

27.4 Nuclear Option

If you want to really start from scratch then the best way to do this is to completely reinstall Ubuntu wiping the current installation. This should also be done each time the OS is upgraded.

Part VIII

Additional - Appendices

Appendix A

Empire Strikes Back - Beep Tune

```
#!/bin/bash

beep -l 350 -f 392 -D 100 -n -l 350 -f 392 -D 100 -n -l 350 -f 392 -D 100
    -n -l 250 -f 311.1 -D 100 -n -l 25 -f 466.2 -D 100 -n -l 350 -f 392 -
   D 100 -n -l 250 -f 311.1 -D 100 -n -l 25 -f 466.2 -D 100 -n -l 700 -f
   392 -D 100 -n -l 350 -f 587.32 -D 100 -n -l 350 -f 587.32 -D 100 -n -l
    350 -f 587.32 -D 100 -n -l 250 -f 622.26 -D 100 -n -l 25 -f 466.2 -D
   100 -n -l 350 -f 369.99 -D 100 -n -l 250 -f 311.1 -D 100 -n -l 25 -f
   466.2 -D 100 -n -l 700 -f 392 -D 100 -n -l 350 -f 784 -D 100 -n -l 250
    -f 392 -D 100 -n -l 25 -f 392 -D 100 -n -l 350 -f 784 -D 100 -n -l
   250 -f 739.98 -D 100 -n -l 25 -f 698.46 -D 100 -n -l 25 -f 659.26 -D
   100 -n -l 25 -f 622.26 -D 100 -n -l 50 -f 659.26 -D 400 -n -l 25 -f
   415.3 -D 200 -n -l 350 -f 554.36 -D 100 -n -l 250 -f 523.25 -D 100 -n
   -l 25 -f 493.88 -D 100 -n -l 25 -f 466.16 -D 100 -n -l 25 -f 440 -D
   100 -n -l 50 -f 466.16 -D 400 -n -l 25 -f 311.13 -D 200 -n -l 350 -f
   369.99 -D 100 -n -l 250 -f 311.13 -D 100 -n -l 25 -f 392 -D 100 -n -l
   350 -f 466.16 -D 100 -n -l 250 -f 392 -D 100 -n -l 25 -f 466.16 -D 100
    -n -l 700 -f 587.32 -D 100 -n -l 350 -f 784 -D 100 -n -l 250 -f 392 -
   D 100 -n -l 25 -f 392 -D 100 -n -l 350 -f 784 -D 100 -n -l 250 -f
   739.98 -D 100 -n -l 25 -f 698.46 -D 100 -n -l 25 -f 659.26 -D 100 -n -
   l 25 -f 622.26 -D 100 -n -l 50 -f 659.26 -D 400 -n -l 25 -f 415.3 -D
   200 -n -l 350 -f 554.36 -D 100 -n -l 250 -f 523.25 -D 100 -n -l 25 -f
   493.88 -D 100 -n -l 25 -f 466.16 -D 100 -n -l 25 -f 440 -D 100 -n -l
   50 -f 466.16 -D 400 -n -l 25 -f 311.13 -D 200 -n -l 350 -f 392 -D 100
   -n -l 250 -f 311.13 -D 100 -n -l 25 -f 466.16 -D 100 -n -l 300 -f
   392.00 -D 150 -n -l 250 -f 311.13 -D 100 -n -l 25 -f 466.16 -D 100 -n
   -l 700 -f 392

exit 0
```

Appendix B

Bash Window Control

```
1  #! /bin/bash
2
3  while true; do
4          #Open PDF
5          evince /home/adam/Desktop/Thesis.pdf &
6          #Grab the PID so we can close it later
7          pid="$!"
8          #Get time of opening
9          T=$(date +"%T")
10         #Echo the PID for lulz
11         echo "PID: $pid, opened at $T"
12         #Sleep for 1.5s to allow the pdf to open
13         sleep 1.5s
14         #Get the X name of the window
15         name=$(wmctrl -lp | awk -vpid="$pid" '$3==pid{print $1}')
16         #Position the window
17         wmctrl -ir "$name" -e 1,1,1,1024,1280
18         #Wait for 5 minutes
19         sleep 5m
20         #Kill the program
21         kill "$pid"
22         #Echo the time the program was last closed for sanity checking
23         T=$(date +"%T")
24         echo $T
25 done
```

Appendix C

SSMPT configuration File

```
1  # Config file for sSMTP sendmail
2  #
3  # The person who gets all mail for userids < 1000
4  # Make this empty to disable rewriting.
5  #root=postmaster
6  root=MyEmailAddress@gmail.com
7
8  # The place where the mail goes. The actual machine name is required no
9  # MX records are consulted. Commonly mailhosts are named mail.domain.com
10 #mailhub=mail
11 mailhub=smtp.gmail.com:587
12
13 AuthUser=MyEmailAddress@gmail.com
14 AuthPass=MyPassword
15 UseTLS=YES
16 UseSTARTTLS=YES
17
18 # Where will the mail seem to come from?
19 #rewriteDomain=
20 rewriteDomain=gmail.com
21
22 # The full hostname
23 #hostname=MyMediaServer.home
24 hostname=MyEmailAddress@gmail.com
25
26 # Are users allowed to set their own From: address?
27 # YES - Allow the user to specify their own From: address
28 # NO - Use the system generated From: address
29 FromLineOverride=YES
```

Appendix D

Fail2Ban jail.local Example

```
1  [ssh]
2  enabled = true
3  port = 500
4  maxretry = 2
5
6  [pam-generic]
7  enabled = true
8  maxretry = 2
9
10 [ssh-ddos]
11 enabled = true
12 port = 500
13
14 [apache]
15 enabled = true
16 logpath = /var/log/apache2/access.log
17
18 [apache-noscript]
19 enabled = true
20 logpath = /var/log/apache2/access.log
21
22 [apache-overflows]
23 enabled = true
24 logpath = /var/log/apache2/access.log
25
26
27 [apache-postflood]
28
29 enabled = true
30 port = http,https
31 filter = apache-postflood
32 logpath = /var/log/apache2/access.log
33 findtime = 10
34 maxretry = 1
```

Appendix E

Fail2Ban Apache Rule

```
1 [Definition]
2
3 failregex = <HOST>.*"[A-Z]*␣/(cms|user|muieblackcat|db|cpcommerce|wp-
      login|joomla|awstatstotals|wp-content|wp-includes|pma|phpmyadmin|
      myadmin|mysql|mysqladmin|sqladmin|mypma|admin|xampp|mysqldb|pmadb|
      phpmyadmin1|phpmyadmin2).*"
4                 <HOST>.*\"␣
                      (502|500|417|416|415|414|413|412|404|405|403|401|400)
5
6 ignoreregex␣=␣.*\"GET \/(press|mailto|domestic|word).*
```

Appendix F

Unique Word List

Credit to: [tothink.com]

acrobat africa alaska albert albino album kevin scholar update
alcohol alex alpha amadeus amanda amazon legacy sleep william
america analog animal antenna antonio apollo locate square absorb
april aroma artist aspirin athlete atlas mammal stretch anagram
banana bandit banjo bikini bingo bonus mayday survive ariel
camera canada carbon casino catalog cinema modest teacher baboon
citizen cobra comet compact complex context navy today beach
credit critic crystal culture david delta night turtle betty
dialog diploma doctor domino dragon drama octavia valery bless
extra fabric final focus forum galaxy oval wonder bread
gallery global harmony hotel humor index parlor absurd bundle
japan kilo lemon liter lotus mango phoenix andy canary
melon menu meter metro mineral model plato armor choice
music object piano pirate plastic radio powder bahama cliff
report signal sport studio subject super promo beast cotton
tango taxi tempo tennis textile tokyo rachel between decide
total tourist video visa academy alfred reward bogart donor
atlanta atomic barbara bazaar brother budget round break enrico
cabaret cadet candle capsule caviar channel sailor button fame
chapter circle cobalt comrade condor crimson secure candid fire
cyclone darwin declare denver desert divide sherman chris flood
dolby domain double eagle echo eclipse size clone freddie
editor educate edward effect electra emerald song cover gate
emotion empire eternal evening exhibit expand spoon deposit glass
explore extreme ferrari forget freedom friday stick druid grid
fuji galileo genesis gravity habitat hamlet support epoxy hair
harlem helium holiday hunter ibiza iceberg think fast hippie
imagine infant isotope jackson jamaica jasmine tonight fish hydro
java jessica kitchen lazarus letter license tribune floor invest
lithium loyal lucky magenta manual marble under front join
maxwell mayor monarch monday money morning vendor gelatin karl
mother mystery native nectar nelson network vista goblin shelf
nikita nobel nobody nominal norway nothing wave griffin sincere
number october office oliver opinion option android halt sofia
order outside package pandora panther papa define hobby spend
pattern pedro pencil people phantom philips ironic imitate stella
pioneer pluto podium portal potato process ego invite sunset
proxy pupil python quality quarter quiet kiwi joseph tavern

rabbit radical radius rainbow ramirez ravioli legend shelter tobacco
raymond respect respond result resume richard lopez smile trade
river roger roman rondo sabrina salary margin stadium type
salsa sample samuel saturn savage scarlet meaning stuart value
scorpio sector serpent shampoo sharon silence morph sweet visible
simple society sonar sonata soprano sparta needle telecom watch
spider sponsor abraham action active actor nissan toga wisdom
adam address admiral adrian agenda agent ohio twin deal
airline airport alabama aladdin alarm algebra owner vega include
alibi alice alien almond alpine amber parole year prodigy
amigo ammonia analyze anatomy angel annual phrase accent ski
answer apple archive arctic arena arizona plume anvil yes
armada arnold arsenal arthur asia aspect prague arrow poncho
athena audio august austria avenue average quest bali prime
axiom aztec bagel baker balance ballad raja beatles race
ballet bambino bamboo baron basic basket rhino beyond rent
battery belgium benefit berlin bermuda bernard rubber bonanza rodent
bicycle binary biology bishop blitz block saint broken saddle
blonde bonjour boris boston bottle boxer shake buzzer season
brandy bravo brazil bridge british bronze shine carrot monkey
brown bruce bruno brush burger burma slalom cipher nebula
cabinet cactus cafe cairo calypso camel sound conan origami
campus canal cannon canoe cantina canvas stage crack prelude
canyon capital caramel caravan career cargo sting desire puzzle
carlo carol carpet cartel cartoon castle supreme drum robin
castro cecilia cement center century ceramic thomas erosion moral
chamber chance change chaos charlie charm torch father nickel
charter cheese chef chemist cherry chess trinity flame orinoco
chicago chicken chief china cigar circus unit forbid prepare
city clara classic claudia clean client venice fuel remote
climax clinic clock club cockpit coconut vital gibson rose
cola collect colombo colony color combat weather gopher moses
comedy command company concert connect consul annex ground nina
contact contour control convert copy corner dispute harris orion
corona correct cosmos couple courage cowboy null honey pretend
craft crash cricket crown cuba dallas fax info repair
dance daniel decade decimal degree delete ladder ivan rover
deliver delphi deluxe demand demo denmark lesson juice mouse
derby design detect develop diagram diamond lorenzo ship noise
diana diego diesel diet digital dilemma margo spain othello
direct disco disney distant dollar dolphin mercy star promise
donald drink driver dublin duet dynamic morris subway reply
earth east ecology economy edgar egypt neuron swim rudolf
elastic elegant element elite elvis email nitro temple nancy
empty energy engine english episode equator olga tommy orchid
escape escort ethnic europe everest evident page uncle paper
exact example exit exotic export express paul version prosper
factor falcon family fantasy fashion fiber pierre yellow rival
fiction fidel fiesta figure film filter pogo alfonso saga
finance finish finland first flag flash press appear natasha
florida flower fluid flute folio ford quick austin oregano
forest formal formula fortune forward fragile ranger balsa paprika

france frank fresh friend frozen future ribbon beauty provide
gabriel gamma garage garcia garden garlic ruby billy riviera
gemini general genetic genius germany gloria salt book sahara
gold golf gondola gong good gordon shallow bucket nice
gorilla grand granite graph green group shirt byte oberon
guide guitar guru hand happy harbor slow cave orca
harvard havana hawaii helena hello henry south clarion parent
hilton history horizon house human icon stamp conduct perform
idea igloo igor image impact import stock current plate
india indigo input insect instant iris sweden detail spray
italian jacket jacob jaguar janet jargon tictac easy strange
jazz jeep john joker jordan judo torso except summer
jumbo june jungle junior jupiter karate triton felix target
karma kayak kermit king koala korea urban flex tina
labor lady lagoon laptop laser latin verona forever tunnel
lava lecture left legal level lexicon voice gallop gentle
liberal libra lily limbo limit linda wedding ginger greek
linear lion liquid little llama lobby armani grace herbert
lobster local logic logo lola london genuine guest inch
lucas lunar machine macro madam madonna nurse heart joel
madrid maestro magic magnet magnum mailbox jet hope justice
major mama mambo manager manila marco lake ingrid maze
marina market mars martin marvin mary life james milan
master matrix maximum media medical mega love julius george
melody memo mental mentor mercury message marion shoe grille
metal meteor method mexico miami micro middle spark heroic
milk million minimum minus minute miracle mystic storm initial
mirage miranda mister mixer mobile modem never sugar joshua
modern modular moment monaco monica monitor nixon table kimono
mono monster montana morgan motel motif open tibet medusa
motor mozart multi museum mustang natural paint tower mile
neon nepal neptune nerve neutral nevada peace unicorn giant
news next ninja nirvana normal nova pinball voodoo hammer
novel nuclear numeric nylon oasis observe point young hexagon
ocean octopus olivia olympic omega opera presto alias isabel
optic optimal orange orbit organic orient quiz apropos journal
origin orlando oscar oxford oxygen ozone region avatar kinetic
pablo pacific pagoda palace pamela panama rider barcode member
pancake panda panel panic paradox pardon rufus before miller
paris parker parking parody partner passage scale bison gilbert
passive pasta pastel patent patient patriot shannon border harvest
patrol pegasus pelican penguin pepper percent side buenos husband
perfect perfume period permit person peru small caesar ivory
phone photo picasso picnic picture pigment speech chant judge
pilgrim pilot pixel pizza planet plasma stand clark leonid
plaza pocket poem poetic poker polaris store congo memphis
police politic polo polygon pony popcorn swing danube mimic
popular postage precise prefix premium present time dexter gossip
price prince printer prism private prize touch eddie hazard
product profile program project protect proton truck exile immune
public pulse puma pump pyramid queen urgent field jason
radar ralph random rapid rebel record vibrate flipper juliet

recycle reflex reform regard regular relax vortex fractal leopard
reptile reverse ricardo right ringo risk wheel game michael
ritual robert robot rocket rodeo romeo cake giraffe mimosa
royal russian safari salad salami salmon idiom gray gram
salon salute samba sandra santana sardine obscure gustav heaven
school scoop scratch screen script scroll job heavy inca
second secret section segment select seminar laura horse jerome
senator senior sensor serial service shadow light inside jump
sharp sheriff shock short shrink sierra lunch jester lima
silicon silk silver similar simon single mask july miguel
siren slang slogan smart smoke snake mike sigma mission
social soda solar solid solo sonic nadia split prefer
source soviet special speed sphere spiral newton story rio
spirit spring static status stereo stone north sulfur learn
stop street strong student style sultan opus taboo list
susan sushi suzuki switch symbol system palma ticket malta
tactic tahiti talent tarzan telex texas pearl trivial match
theory thermos tiger titanic tomato topic place unique mirror
tornado toronto torpedo totem tractor traffic polka warning nato
transit trapeze travel tribal trick trident pretty absent gregory
trilogy tripod tropic trumpet tulip tuna quota ambient gyro
turbo twist ultra uniform union uranium remark archer herman
vacuum valid vampire vanilla vatican velvet road axis hostel
ventura venus vertigo veteran victor vienna sabine bazooka invent
viking village vincent violet violin virtual scuba benny jimmy
virus vision visitor visual vitamin viva shave blast kansas
vocal vodka volcano voltage volume voyage sinatra brave sister
water weekend welcome western window winter snow buffalo farmer
wizard wolf world xray yankee yoga spell camilla food
yogurt yoyo zebra zero zigzag zipper state child fossil
zodiac zoom acid adios agatha alamo sunday clever frog
alert almanac aloha andrea anita arcade tape costume fruit
aurora avalon baby baggage balloon bank toast data geneva
basil begin biscuit blue bombay botanic toyota dinner impress
brain brenda brigade cable calibre carmen trust enjoy gizmo
cello celtic chariot chrome citrus civil user explain extend
cloud combine common cool copper coral virgo fiona famous
crater cubic cupid cycle depend door waiter float confide
dream dynasty edison edition enigma equal whiskey frame garbo
eric event evita exodus