The Red Book

Linux

suck-o.com tutorials/howtos

Index

Topic	Page
Managing processes on Linux	3 - 5
Scan your network with Nmap	6 - 7
Debian: Enable 3D rendering Nvidia Cards	8 - 9
Bastile Hardening Tool Guide	10 - 16
Debian: wireless zepto 3215W	17 - 18
Debian: Setting up a DNS server	19 - 23
Debian: Setting up a DHCP server	24 - 26
Optimize your mysql server	27
Remote MySQL	28 - 29
Basic iptables setup	30 - 32
LPIC-1	33 - 40
Generate strong password in Linux	41

Managing Processes On Linux

Original author bad_brain of suck-o.com
removing the original author will let your penis shrink.

In this little tutorial I will explain how to manage processes on Linux, how to start and stop them, how to end a process if the desktop is frozen, and of course how to get info about running processes.

---Start a process---

Different from Windows an executable don't need a special file extension (like .exe) on Linux/Unix, so the extension can be anything or even no extension at all. But it is a good practice to give Bash scripts a .sh extension for example, so executables can be easier identified, especially on a multi-user system and/or if colors are not enabled in the terminal (executables are usually displayed in a yellow-ish color).
So, how to start a process? Simple, you just have to enter the name of the executable, if you are not in the same directory as the executable you have to use the full or relative path like:

Code:
```
/usr/local/myprocess
```

If you are in the same directory as the executable you don't need to enter the path, but you have to add ./ before the executable name, like:

Code:
```
./myprocess
```

Why ./ ? The intention behind this is to make sure a user REALLY wants to run an executable, if no ./ would be needed it would be easy to trick a user, so it is a security feature.

But of course there is a way to start executables just by entering the name, for this the executable must be located inside one of the paths where the shell interpreter looks for them.
The paths are stored in the $PATH variable:

Code:
```
serv:~# echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/bin/X11
```

So if you place the executable named "myprocess" in /usr/local/bin for example you can start it with:

Code:
```
myprocess
```

Ok, all the above not only applies to processes, it applies generally to all executables, so let's get a little more specific:
First, let's start your process again, asuming it's in $PATH:

Code:
```
myprocess
```

Ok, runs fine, BUT the screen is blocked now because the process runs in the foreground. Of course you can open a 2nd terminal now, but this is not really comfy.
So, let's start the process again, but this time as background process:

Code:
```
myprocess &
```

Simple, isn't it? Adding a & does the trick!
Hm, wait, when I close the terminal the process ends too....

Ok, also no problem, let's start the process again as background process that will keep running even when you log out:
Code:
```
nohup myprocess &
```

"nohup" means "no hangup", so the process will not stop when the user that started it logs out.
Another problem that might appear is that the process prints its output on the screen, if the output is kinda useful and you want to log it for later checks you can easily write it to a file:
Code:
```
myprocess > /mylogfile &
```

But if the output is kinda useless you can send it to "nowhereland":
Code:
```
myprocess > /dev/null &
```

---Stop a process---

Well, how to stop a process kinda depends on you, I am not talking about violently kill a process now, I mean how to stop it regularly. The best is of course when you have implemented this option in your code already or you use an external start/stop script (this applies to permanent processes like servers for example). But a process of course also simply end by itself when it is done with its job (like a backup script for example).

---To stop a process that "hangs"---

If there is an error in the code for example it can happen that the process doesn't end like it should, this can lead to various problems from a high load on a server system to freezing the complete desktop on a desktop system.
What you NEVER should do is to press the "reset" button, this is really only the LAST option because it can lead to a loss of the data that was processed at that moment or it can even corrupt a whole database.
Let's start with the most simple case when you run a process in the shell and it hangs so the terminal is blocked, pressing CTRL+C should end the process.
Another way is the "kill" command, if the terminal is not blocked and/or you can open a 2nd one you can stop the process with:
Code:
```
kill <PID>
```

How do I know the PID of the process?

There are different ways, you can use the "top" command or "ps -A", if you know the name of the process you can also use the "pidof" command:
Code:
```
pidof myprocess
```

Notice that there are different kinds of the "kill" command, or better said the "kill" command can send different signals to the process. The default signal is 15 which "asks the process to please stop" (so "kill 725" equals "kill -15 725"), this way the process shuts down regularly and there is no danger of a data loss. If this doesn't work you can be a little more brutal by sending the 9 signal:
Code:
```
kill -9 725
```

This ends the processe with the PID 725 without taking care of the data that is maybe processed by that process at the moment, so this can result in a data loss (of that process of course, NOT of your whole system).
Now let's imagine you started your process on your desktop system and "uh-oh...desktop frozen!"...still no need to press "reset", even if your mouse and keyboard seem to have no effect.
By pressing CTRL+Alt+F1 you have a good chance to switch into textmode where you can end the process that caused the hang with the methods mentioned above.

---Getting info about processes---

As mentioned about (process PID) there are different ways to get informations about processes, the most basic one is:

Code:
```
ps -A
```

You can see the PID, in how many terminals the process is visible/used (TTY), for how long the process is running, and of course the process name.
If you want to have more detailed info "ps aux" is a very useful command, it display information like the process owner, PID, CPU and RAM usage, etc., a very valuable info is the actual command that started the process.
If you want to have info in realtime the "top" command is very useful, it also displays CPU and RAM usage, so it is very good to monitor the system if you suspect a process to be a resource hog.
Both ps and top have numberous options, so I recommend to check the manual pages.

---Be nice!---

Um, what? No joke, you can tell a process to be nice! Ok, actually you can give a process a priority, which means a processe with higher priority gets time (better said CPU runtime) to finish first before the process with lower priority is allowed to use the resources.
And the command for this is "nice". the highest priority is -20, the lowest priority is 19.
So let's start out process with the highest priority:

Code:
```
nice -20 myprocess
```

It is better not to be "too nice" to processes, and I wouldn't use a higher priority than -10 for a regular process, because higher priorities are only used for REALLY important kernel processes for example that are always more important than a service or a script.
The "nice" command is only used to start a new process, if the process is already running and you want to give it a different priority you will have to use the "renice" command.
So let's give our already running process (with the PID 725) a different priority (-5) because -20 was a little too much:

Code:
```
renice -5 725
```

Notice that you have to use the PID for renice, and not the process name.

Scan your network with Nmap

Hello guys.

Today I'll be quickly going through one of the best features in the network scanner Nmap.

So, what is Nmap?

It's one of the most popular network scanners used by hackers, system administrators and loads of more people. It quickly scans the network by sending out raw IP packets to determinate what hosts are available, what services' they are offering and what type of filter or firewall they are running and what OS they are using and a lot more such as device type, mac adress and reverse DNS names.

The output from Nmap is a table with all the host and what they are running, this is a off course dependent on which parameters you include in your syntax. The state of the service is either: open, filtered, unfiltered and closed.

Open means that the machine is listening for packets on that port, Filtered means that a filter or a firewall is preventing Nmap from scanning that port, and Nmap couldn't really tell whether the port is open or closed.
Unfiltered means that a port is active and responds to Nmap but Nmap can really tell whether the port is opened or closed.

Nmap has tons of features, but I'll be showing the one I like the best.

Let's get going by installing Nmap.

Some of you will be able to simply install the program using apt-get.

The string is: **sudo apt-get install nmap**

After the install, Nmap should be ready to scan.

Make sure you're connected to a network.

What we will do is simply scan all the computers in a specific IP range to determinate running services and ports.

Why would we want to scan our network?
I suppose everyone should keep an eye on their network, whether it's your sister running a malicious trojan or if your company's server is running any unnecessary dangerous software or maybe a hacker trying to determinate how he will attack a certain victim. Either way, I'm sure you will find a use for this program.

What you need to know:

The IP range in you LAN. The simplest way to obtain this is possibly to check with your DHCP server. Or if you are truly 1337, you might be able to obtain it using your subnet.

Let's get started:

nmap 192.168.1.* -oG nmap-scan.txt

What this will do is scan every computer inside this local IP range and save down the info using the -oG parameter to a certain nmap-scan file. One of the reasons why I like to save the scan down is simply because it's a lot simpler to read.

Then simply check what nmap obtained from your network:

cat nmap-scan.txt | grep open

Since we can filter out any unnecessary information such as closed ports, we'll use the grep command to search for a specific string.

So, read your output file and make sure no one is bleeding any malware on your network!

DEBIAN: enable 3D rendering Nvidia Cards

Ok so I have done this tutorial mainly for Lenny users as it has taken me best part of a day to get this working and your system needs this for 3D games and Compiz to run.

K now i have started with a fresh install and have only installed the basic start up files and the XFCE Desktop. First of all the sources.list need include "contrib non-free" if the aren't already. Now open a shell as root user and follow these commands.

Code:

```
update-pciids
apt-get install module-assistant nvidia-kernel-source
m-a prepare
m-a a-i nvidia
apt-get install nvidia-glx
depmod -a
modprobe nvidia

# This is only necessary if you start to use nvidia for the first time:

dpkg-reconfigure xserver-xorg
```

now we check the /etc/X11/xorg.conf and search for the section "Module" and check for

Code:

```
Load "glx"
```

and make sure it is present. Now check that there is no reference to "dri" or "GLCore" and if there are comment them out with "#" before saving.

In the "Device" section for your video card, change the driver (normally nv or vesa) to nvidia. If the driver is not defined, add the line

Code:

```
Driver "nvidia"
```

Now log out and log back onto your system and sart another shell as root and type

Code:

```
glxinfo|grep rendering
```

If you see

Code:

```
bash: glxinfo: command not found
```

then we need to install the mesa-utils package which we can do with apt

Code:

```
apt-get install mesa-utils
```

Now if we try

Code:

```
glxinfo|grep rendering
```

we should see

Code:

```
glxinfo|grep rendering
direct rendering: Yes
```

I hope this saves people the amount of time it took me :XD

Bastile Hardening Tool Guide

This is other guide I made, more than a guide is an article with descriptions of the options it tries to modify from your systems, I made it because although I don't use it much, I always forget what those options try to do even while reading the default descriptions in the tool.

Hope you find it useful, at least to know what you could change in your system to make it a little more secure, if not, feel free to discard it, flame it or delete it Wink

Contents

1. Introduction
2. What's Bastille
3. Bastille flags
4. Hardening Walk-through
5. Conclusion

Introduction

Security is a most, we all know that, or do we? There are a lot of systems that are not secured in any way all over the net, we know this because we read security reports updated daily, and you know this because there's a chance that you've been owned/rooted/pwnd once or more times ^^ so every effort you take to make your system secure can help you to avoid loosing all your hair, but we know, securing a system can take a lot time and if you're the lazy-type admin will take longer, but fear not, others might to help you, even more, others might try to educate you Smile

This is an article that offers you information about one easy way to apply harder security to your Linux system. It's important to say that by reading this article is assumed that you have a basic-intermediate knowledge about how Linux works, however, I try to made the article as easy to follow as I could for the beginners (given the fact I'm a beginner too) Smile

From now on, I'll be referring to "hardening" as the process to implementing rules to activate or deactivate a feature in order to make your system's security more stronger.

At must note before you go crazy changing things, Bastille, as any other automatic tool to do things, can do a lot of good but also a lot of damage if used improperly, so, read carefully, contrast what you see here with other sources, and more importantly, test it in a non-production system, something that you can screw up without lose your job or important files.**What's Bastille?**

This is a tool, or more appropriate, a set of scripts that helps you (yes lazy admin, I'm talking to you!) to set a stronger security on a system. Bastille can be used over most common Linux distributions, HP-UX and Mac OSX, however, this article is just based over Linux, so understand that not all the settings posted here are equal in different systems.

Bastille works by showing a set of questions, all regarding to a feature that could be secured by using one of the automated scripts that Bastille is based on. What I like of Bastille is that not only is easy to follow and have pretty good things to enforce security but it's designed to educate the user by offering explanations (although sometimes not so useful explanations), and the security implications, on every setting that will be changed.

To this point you might be a little interested but keep wondering what if the settings fails or what if you don't feel good about the changes you've made? If you're asking this questions then you will most that pleased to know that Bastille does come with a very good "undo changes" feature, so, if you are not happy with the changes you can run ReverseBastille.

Bastille also offers you a way to secure more than one system based in just one configuration file that can be called using the -i flag.

On the downside, Bastille isn't capable of securing "all" your system, by this I mean that although Bastille can take care over different services, there are others that you'll have to secure by your self, so, Bastille's work is to help you to secure "Linux" but not to secure your MySQL server or your PureFTPD Server, that's your work!

Bastille's Flags

You should see the options and arguments that can be used with Bastille by issuing Bastille -h, but since I know you're lazy I will post them here Smile

Code:

```
root@root [~]# Bastille -h
Usage: Bastille [ [ -h | -b [--<os version> ] | -c | -r | -x | -a [--<os version>] | -l ] -i <alternate config> ]
        -b : use a saved config file to apply changes
             directly to system
        -c : use the Curses (non-X11) GUI
        -h : this help
        -i : input alternate configuration file
        -r : revert Bastille changes to original file versions (pre-Bastille)
        -l : list the standard config file(s) (if any) that matches the last
             run config
        --os version : ask all questions for the given operating system
             version.  e.g. --os HP-UX11.11
        -x : use the Perl/Tk (X11) GUI
        --assess / -a : run Bastille in assessment mode, generating a report and displaying it in a browser
        --assessnobrowser : run Bastille in assessment mode, generating a report with no browser
```

The flags are self explanatory, however I'll explain the --assess flag. Basically, it will launch a browser with a report, in this report you'll see a set of questions that are intended to let you know what security rules you should think about in order to make your system more secure. The explanations of every question isn't the best but will help you more than nothing Wink

Hardening Walk-through

In this part I cover the questions you'll be asked while running Bastille and the security goal of each, although you'll find explanations of every change you're about to make, I hope this will gives you a better idea about what to expect of the tool.

Doesn't matter if you use Bastille with the curses based interface or TK's, the questions will be the same on a Linux system.

Bastille is divided by sections, questions and answers, the answers are of tree types: Yes, No and user input, at the moment you want to return to a previous questions just click the Back button or press Esc. For easy to read purpose, I'll be using this format:

Section: FilePermissions
Feature: Ping Usage
Question: Something
Explanation: Something
Recommended Answer: Yes|No|User Input

Let's get start:

To understand the first set of questions you need to understand what's SUID. Is a bit that gives to "common users" the permit to access a the file or a feature that should be run by root. This bit exists for convenience but it can be for obvious reasons a security problem.

Section: FilePermissions
Feature: SUID Mount/Unmount
Question: Would you like to disable SUID status for Mount/Unmount?
Explanation: If SUID is set, normal users can mount and unmount devices (such as flash drives, cdroms and so on). In desktop systems when "you" are the only one that have access to such system then you can use this bit. Even when if you don't set it you still mount things by changing to root using su, but if so, if KDE try to mount some usb drive, will ask you for root's password which, if you're the only user, isn't necessary.
Recommended Answers: If you're the only one with access to the system then Answer No, but if the system is used by several people then answer Yes.

Section: FilePermissions
Feature: Ping Usage
Question: Would you like to disable SUID status for Ping?
Explanation: By default, the SUID bit is set to ping so only root can use this tool and you should leave it that way.
Recommended Answer: If you're the only one with access to the system then you can safely remove the SUID of Ping by answering No to this one.

Section: FilePermissions
Feature: BSD r-tools
Question: Would you like to disable the r-tools?
Explanation: The BSD r-tools are things like rsh, rlogin, rdist and such. The problem with this tools is that everything is passed in clear text, that means that if someone is sniffing the LAN where you're at in the same time you're using rlogin to connect to some remote system, then that "someone" can be able to read the traffic you're sending, thus, gaining your user and password details to connect to the remote system.
Recommended Answer: For obvious reasons the answer should be "No", don't worry about loosing this tools since there are others that you can implement more secure than this ones.

Section: AccountSecurity
Feature: BSD r-tools
Question: Should Bastille disable clear-text r-protocols that use IP-based authentication?
Explanation: The difference of this one with the later is that this question is referred to r-tool's services that can be using to access, remotely, to your system (i.e: using rlogin to access to your system from other machine).
Recomended Answer: Again, everything send using r-tools are in clear text so the answer to this one should be "No" as the other.

Section: AccountSecurity
Feature: Password Aging.
Question: Would you like to enforce password aging?
Explanation: The idea with this feature is to change the default time that a password should be used before the system prompts the user to change it. By default, this is set as 99.999 days so a user can use a password that amount of time without having to change it, however, in large environment or systems where security is a most, admins set a lower value forcing the password to be changed frequently which is a good approach. Bastille will set the time to 60 days but you can change that by editing the file /etc/login.defs
Recommended Answer: It depends on the system you're working at, if it's a system that has many users that come and goes then it's good to answer "Yes" in here, that way you can reduce the compromising passwords.

Section: AccountSecurity
Feature: TTY's access.
Question: Should we disallow root login on tty's 1-6?
Explanation: This setting (if answered Yes) will disallow the root login at any tty (virtual terminals), by doing this, to use the root account you'll have to login first as regular user in any tty and then issue the su command. For many users this could be a pain actually but it isn't a bad idea at all since if root's password is compromised, the person

who know that password won't be able to login directly as root, he'll have to know the password of a normal user account too.
Recommended Answer: "Yes" Smile

Section: BootSecurity
Feature: Grub's Password.
Question: Would you like to password-protect the grub prompt?
Explanation: By enabling this, you'll have to enter a password in order to grub can be loaded and used. This is a way to prevent people who might have "physical" access to the system to gain root access by using the grub (i.e: setting the flag "Single" to the kernel line end to boot directly as root Wink)
Recommended Answer: This setting might gives you lot of troubles if you have a multiboot system, if so, you should follow Bastille's advice and answer "No" to this one, however, if you don't have multiboot systems then you can safely can answer Yes to this.

Section: BootSecurity
Feature: Ctrl+Alt+Del reboot.
Question: Would you like to disable CTRL-ALT-DELETE rebooting?
Explanation: A system can be rebooted by using ctrl+alt+del, this will send a kill signal to gracefully end services, clean tmp and then reboot. Someone then though that by disabling this an attacker couldn't reboot the system but he can by shutting the computer's power supply, then, since a hard reboot could actually damage your system (because things aren't killed "gracefully"), is better to let the attacker to reboot the system by using ctrl+alt+del Razz
Recommended Answer: Based on the fact that even disabling this the system can be reboot is better to answer "No" to this Wink

Section: Inetd
Feature: TCP Wrappers and xinetd
Question: Would you like to set a default-deny on TCP Wrappers and xinetd?
Explanation: inetd and its follower xinetd are called superservers, this can be on control of other services like ssh, ftp, email and the like, the idea behind this superservers is first to have a control center for other servers and second to execute those servers under the permissions of inetd and xinetd. What this question does is to set a rule that will deny external access to all the servers running under inetd or xinetd, by external access I mean anything that it is not localhost.
Recommended Answer: This one I'll leave it to you, you could answer "Yes" here and then make changes to the server files under /etc/inetd.d or /etc/xinetd.d in order to allow access to the servers your system might use (i.e: if you have a ftp server and such), answering yes will be good because with all denied you only have to worry by allowing access to really needed servers, thus avoiding security breaks by forgetting to deny access to some service.

Section: Inetd
Feature: Telnet
Question: Should Bastille ensure the telnet service does not run on this system?
Explanation: Telnet is old and insecure by default, every piece of traffic going under telnet will be in clear text, although even today telnet might have some use (think of smpt spoofing), this particular question is related to "telnet as a service", meaning that we're talking about allowing that someone from another machine could use telnet to access your system.
Recommended Answer: Obviously the answer here is "Yes". For those that are concerned, this won't disable your telnet clients.

Section: Inetd
Feature: FTP
Question: Should Bastille ensure inetd's FTP service does not run on this system?
Explanation: FTP is similar to telnet in the way that it's insecure by default, everything send or received using ftp goes through the network in clear text, if you are in the need to have an active ftp server, then you should try with a better solution like sftp.
Recommended Answer: You should answer "Yes" in here and use a better solution than the old FTP, however it's up to you the final decision of this answer.

Section: Inetd

Feature: Authorized Use message
Question: Would you lite to display "Authorized Use" messages at login time?
Explanation: What this feature does is to edite the /etc/issue file and configure /etc/login.defs to call /etc/issue before every login showing a message to let users trying to connect what kind of system their are trying to access. Bastille will create a very standard message, you should change the contents of the /etc/issue file in order to adapt it better to your system specification.
Recommended Answer: Answer what you want with this, it's said that this could help you if some day you're prosecuting some attacker.

Section: Inetd
Feature: Authorized Permit
Question: Who is responsible for granting authorization to use this machine?
Explanation: The answer you place here will be added to the later message to show users who they may contact in order to obtain permit to access to the system.
Recommended Answer: This answer is "input type based", meaning that you can type whatever you want (i.e: System Admin: uid0@something.com)

Section: ConfigureMiscPAM
Feature: System usage resource limits
Question: Would you like to put limits on system resource usage?
Explanation: PAM is a module that let's you to define rules over specifics services or features to grant or deny access. In this case, this setting could help you to reduce DoS attacks, won't stop them (use a firewall to that), it just can reduce them by adding rules like the amount of process an user can run at anytime. (Remember that under unix environments, every server runs under a virtual user account)
Recommended Answer: In short, answer "Yes" to this question, but don't relay "only" on it to avoid DoS attacks.

Section: ConfigureMiscPAM
Feature: Users accounts console access
Question: Should we restrict console access to a small group of user accounts?
Explanation: Is common that users that have login access over a console have some privileges at the same time (like binary file access), this setting however what offers is a way to allow console logins only to those accounts you trust.
Recommended Answer: You should answer "Yes" to this.

If you answered "Yes" to the previous questions, then you'll be presented with the following question:

Section: ConfigureMiscPAM
Feature: Accounts settings
Question: Which accounts should be able to login at console?
Explanation: Here you'll be able to type the account names (remember, the ones you trust) to give those accounts login access.
Recommended Answer: This answer is "input type based", just write the account names you trust, by default, only root will be granted, but you can add more of course, just set a space between the account's names (i.e: root user1 user2).

Section: Logging
Feature: Additional Logging.
Question: Would you like to add additional logging?
Explanation: This setting (if answered Yes) will give you the possibility to log on a remote system, also, will configure tty7 and tty8 to show those logs.
Recommended Answer: Logging is useful to know what might happen with the things that you "don't see" so you could answer Yes to this one.

Section: Logging
Feature: Remote Logging.
Question: Do you have a remote Logging host?
Explanation: Here you'll set the IP address of the remote system where logs should be placed.

Recommended Answer: This answer is "input based", here you'll need to type the IP address of the server to hold the logs.

Section: Logging
Feature: Process accounting
Question: Would you like to set up process accounting?
Explanation: This will enable the logs daemon to hold information about what process was executed, when and by who.
Recommended Answer: This kind of information is useful, mostly after a problem happen, however this will make your system's processor to work a lot so I leave the answer to you.

Section: MiscellaneousDaemons
Feature: GPM
Question: Would you like to disable GPM?
Explanation: GPM is a service that lets you use the mouse on text based consoles (use a mouse without using the X server).
Recommended Answer: Frankly I don't know why this question is asked at all, I don't personally see a security problem with this one as long as you have it answering only to localhost, anyway, if you like to use the mouse over the console then answer "No" to this.

Section: Sendmail
Feature: Daemon mode
Question: Do you want to stop sendmail from running in daemon mode?
Explanation: Daemon mode means that this service will be always listening in order to process emails (be it by sending them or by receiving them). If you don't have a mail server, then you don't need this service to be running in daemon mode, you could actually configure it to let cron or such to execute it every few minutes to process mail queue or let Bastille to configure it for you.
Recommended Answer: If you have a mail server then answer "No" to this and find more resources on how to protect it.

Section: Apache
Feature: Apache web server
Question: Would you like to deactivate the Apache web server?
Explanation: This setting is self explanatory, answering Yes will deactivate the apache web server, in case you need it you'll have to enable it manually (just run it using chkconfig or some init.d script)
Recommended Answer: Only answer "No" to this one if you use this server (even locally).

Section: Apache
Feature: Bind Apache
Question: Would you like to bind the Web server to listen only to the localhost?
Explanation: "Bind" is a way to create a link between a service and a specific IP address or domain, this setting will bind the apache web server to the localhost (127.0.0.1 addr) so the web server will only listen and process request coming from the localhost.
Recommended Answer: This setting is good if you'll only use the server locally, for instance, if you use it to develop and have a public server in other system. Answer according to your needs.

Section: Firewall
Feature: Packet filtering script
Question: Would you like to run the packet filtering script?
Explanation: If you answer Yes to this one, you'll be asked several things in order to define a set of rules that should be interpreted by ipchains or iptables (like ip masquerade, pre-routing, post-routing and so on...). Bastille have only support to make scripts compatible with 2.2 and 2.4 kernels, so if you use a 2.6 kernel you might have to change or add specific settings after Bastille create the script.
Recommended Answer: Answer this according to your needs, I personally don't use it since I prefer to build the script myself or use other tools like shorewall.

And we're done!

After this, you'll be prompted with a message that will ask you to confirm your changes, to this point, none of the changes have been applied so you can either exit without saving, go back and change the configuration or confirm that you're done.

After confirming, you'll be asked to save the changes, this will create a file in /var/log/Bastille called "last.config". This file could be used to apply the same configurations to other machines with the same system and then called like:

Code:

```
root@root [~]# bastille -i last.config
```

Still at this point, even when you save the configuration no changes have been made so you'll see a final message to ask you if you want the changes to be applied.

Remember that if you applied the changes and aren't happy with the result you could revert everything by using:

Code:

```
root@root [~]# RevertBastille
```

Conclusion

If you have reached until this point then I have to thank you for spend the time to read this, I hope you enjoyed. Feel free to check the main site of Bastille at: http://bastille-linux.sourceforge.net/index.html

As a final note, if you're looking for more in depth security then you could read about projects that use hardening at kernel level like GRSecurity (http://www.grsecurity.net/) or look if your distribution have a hardened version which commonly include kernel patches, also, you might want to check projects like Engarde Linux (http://www.engardelinux.org/), AppArmor (http://www.novell.com/linux/security/apparmor/) and OpenBSD(http://www.openbsd.org/)

Debian: wireless zepto 3215W

This short howto will describe how you can get the wireless card on a zepto 3215W to work with Debian/Lenny. Will of course work with other iwl3945 and iwl4965 cards.

If you are running Lenny, you will have to add an entry in the sources.list file for aptitude.

So do

Code:

```
nano /etc/apt/sources.list
```

and add

Code:

```
deb http://ftp.se.debian.org/debian/ etch main contrib non-free
deb-src http://ftp.se.debian.org/debian/ etc main contrib non-free
```

save with CTRL + O and exit with CTRL + X.

The last line isn't really needed, but I thought it might as well be there for future purposes.

Anyway, start with downloading the firmware package

Code:

```
apt-get install firmware-iwlwifi
```

Then all you have to do is to modprobe the right driver for your card. if it's the iwl 3945 you do

Code:

```
modprobe iwl3945
```

and if it's the iwl4965 you do

Code:

```
modprobe iwl4965
```

There, your card should be working now. Now when you restart, you would have to modprobe it again, so to avoid this you have to add it to your modules file.

Code:

```
nano /etc/modules
```

if it's the iwl 3945 you add

Code:

```
iwl3945
```

and if it's the iwl4965 you add

Code:

```
iwl4965
```

save with CTRL + O and exit with CTRL + X.

There, that's all. Enjoy your wireless! :)

Debian: Setting up a DNS server

This tutorial will cover the basics of setting up a DNS server in a Linux/Debian enviornment (works with Ubuntu just as well). The examples will be based on my own configuration.

What is DNS?

DNS stands for "Domain Name System" (server) and is a service that translates domain names to IP addresses and back. Everytime you write google.com in your browser, the request is sent to your main DNS server and translated. The answer, which contains the IP address, is then sent back to your computer so that you can reach your destination. If your main DNS server can't resolve the query (request), it will send it to one of it's forwarding DNS servers to attempt to solve the request. If the next server can't solve it, it will send it to the next one, and so on and so forth.

Installing the DNS server

In this tutorial I will use bind9. Therefore, invoke the following command:
(Don't forget you have to be root to do this, sudo su)

> **Quote:**
> apt-get install bind9

Configuring the server

First of all, you have to edit the config file named "named.conf.local". There is a file called "named.conf", but since it's already pre-configured in most cases, you can leave it be. if it, however, is not configured, you will have to edit that one instead (named.conf that is).

Anyway, here we go....

If the named.conf is already configured, it should look something like this:

> **Quote:**
>
> // prime the server with knowledge of the root servers
> zone "." {
> type hint;
> file "/etc/bind/db.root";
> };
>
> // be authoritative for the localhost forward and reverse zones, and for
> // broadcast zones as per RFC 1912
>
> zone "localhost" {
> type master;
> file "/etc/bind/db.local";
> };
>
> zone "127.in-addr.arpa" {
> type master;
> file "/etc/bind/db.127";
> };

```
zone "0.in-addr.arpa" { type master; file "/etc/bind/db.0";
};
zone "255.in-addr.arpa" {
type master;
file "/etc/bind/db.255";
};

include "/etc/bind/named.conf.local";
```

If it doesn't look like this, then do editing in the named.conf from here on. If it however looks like this, then you should edit named.conf.local, easy to follow ey?

Anyway, now you have to edit the file called named.conf.local:

Quote:

nano /etc/bind/named.conf.local

Quote:

```
zone "teresa" {
type master;
file "/etc/bind/zones/teresa.db";
};

zone "0.0.168.192.in-addr.arpa" {
type master;
file "/etc/bind/zones/rev.0.168.192.in-addr.arpa";
};
```

zone: This is the zone that the server will refer to for the domains, basically put in your domain name where it says "teresa" (teresa is my server). The second zone is for the reverse lookup. There is a lot to read about zones. This tutorial will not cover much about zones, but if you want to know more then you can refer to the following page:

Quote:

http://www.bind9.net/manual/bind/9.3.2/Bv9ARM.ch01.html#id2546579

type: It's either master or slave. The master is the first DNS that will be used, and in case it breaks down or "something" happens for some reason, then the slave server would still be there to server it. We will only create the master server in this tutorial though.

file: zone definition file. This is where all the zone information is held, with all the domain info, addresses and so on.

0.0.168.192.in-addr.arpa: is the reverse address, and in this case my network address is 192.168.0.0, which means that I should write 0.0.168.192 as the zone name for the reverse lookup.

Now, save the file with "CTRL + O" and then exit to bash with "CTRL + X". Now you have to edit the file called named.conf.options in the same folder, and to do so, invoke the following command:

Quote:

nano /etc/bind/named.conf.options

Quote:

```
options {
directory "/var/cache/bind";

// If there is a firewall between you and nameservers you want
// to talk to, you might need to uncomment the query-source
// directive below. Previous versions of BIND always asked
// questions using port 53, but BIND 8.1 and later use an unprivileged
// port by default.

//query-source address * port 53;

// If your ISP provided one or more IP addresses for stable
// nameservers, you probably want to use them as forwarders.
// Uncomment the following block, and insert the addresses replacing
// the all-0's placeholder.

forwarders {
193.111.152.2;
};

auth-nxdomain no; # conform to RFC1035
listen-on-v6 { any; };
};
```

Note that the line "query-source address * port 53;" is commented in this config, which means that if you have a firewall on the server, you will have to uncomment it since it will use random ports without it.

forwarders: a DNS server that you forward requests to, for example if you setup this DNS as your main home DNS server, you would want a forwarding DNS to your ISP for example, so that if your DNS can't solve a DNS request (like google.com), it will send it to the next server, which is the forwarding address.

There, now save the file and exit to bash. Now it's time to create the zone definition files.

Quote:

```
mkdir /etc/bind/zones
nano /etc/bind/zones/teresa.db
```

create the directory and edit the file.

Quote:

```
teresa. IN SOA ns1.teresa. admin.teresa. (
// Do not modify the following lines!
2006081401
28800
3600
604800
```

```
38400
)

// Replace the following line as necessary:
// ns1 = DNS Server name
// mta = mail server name
// example.com = domain name
teresa. IN NS ns1.teresa.
teresa. IN MX 10 mta.teresa.

// Replace the IP address with the right IP addresses.
www IN A 192.168.0.6
mta IN A 192.168.0.6
ns1 IN A 192.168.0.6
```

ns1: Stands for "name server 1", which is the DNS server.

mta: Stands for mail transfer agent, which basically is another name for SMTPD (Simple Mail Transfer protocol Deamon).

www: world wide web third-level domain name. Yeah you should know what that is and what it's used for.

MX: The MX record stands for "mail exchange" and is a special domain record for routing mail.

A: Stands for "Address Record", and is a record that simply returns a 32-bit IPv4 address.

SOA: Contains information about the DNS zone, the primary ma,e server, email of the domain administrator, the domain serial number, and timers that are meant for zone refreshing.

If you want to read more about the different record types, then go here:
http://en.wikipedia.org/wiki/A_record#A

Quote:

www IN A 192.168.0.6

can basically be described as follows:

Quote:

domain IN record address

now, when you are done, save the file and exit to bash. Now you have to edit the create and edit the file.

Quote:

nano /etc/bind/zones/rev.0.168.192.in-addr.arpa

Quote:

```
@ IN SOA ns1.teresa. admin.teresa. (
2006081401;
```

```
28800;
604800;
604800;
86400
)

IN NS ns1.teresa.
1 IN PTR teresa
```

PTR: Points towards the hostname, and is often used in reverse lookups. as in, you ask for an IP's hostname, instead of the other way around.

Now, all you have to do is save the file and exit to bash, and restart bind to make the new changes load.

Quote:

```
/etc/init.d/bind9 restart
```

Slow resolutions?

Sometimes your server might resolve the requests slowly. To fix this you can try to disable IPv6 by editing the following file:

Quote:

```
nano /etc/default/bind9
```

Quote:

```
OPTIONS="-4 -u bind"
RESOLVCONF=yes
```

There, save and exit to bash...

That is all, hope you learned something. Have fun!

Debian: Setting up a DHCP server

There might come a situation when you need a DHCP server in your network to hand out the TCP/IP configuration for your workstations. This HowTo will show you how it's done in Linux/Debian (Works the same in Ubuntu).

What is a DHCP?

Every computer that is connected to a network and that uses TCP/IP, needs TCP/IP configurations, and since it's not practical and effective to be giving these out manually every time you start your computers in your network, DHCP servers exists. DHCP stands for Dynamic Host Control Protocol, and is a service that simply delivers TCP/IP configurations for computers in a network and "lends" them IP addresses.

Installing the DHCP server

First of all you have to of course install the server itself, and in Debian this can easily be done by using the apt-get command to download and install it from the Debian repository.

Code:

```
apt-get install dhcp3-server
```

Configuring the DHCP server

Then you have to configure the DHCP server to fit your needs, you have to invoke the following command to edit the config file, also take your time to discover the other files that are located in the /etc/dhcp3 folder.

Code:

```
nano /etc/dhcp3/dhcpd.conf
```

Add these settings at the top of dhcpd.conf

Code:

```
ddns-update-style none;
ddns-updates off;
deny client-updates;
one-lease-per-client false;
allow bootp;
option T150 code 150 = string;
```

ddns-update-style none : ddns or Dynamic DNS is a method for clients to tell a DNS server to change it's configurations. This option can either be interim, none or ad-hoc. I recommend using none for starters.

ddns-updates off : same as above, simply shuts it off completely
deny client-updates : If this is set, a client can not update it's configuration. It will have to either drop the leasing of the IP address or wait until the leasing time goes out.

one-lease-per-client false : if this is set to false, a client can have more then one lease.

allow bootp : bootp is commonly used by nodes that doesn't have a harddrive to get a configuration, it basically means that your computer can get the configurations during startup.

And these at the bottom, in this example they are specifik for my network, so you will have to edit the file to fit your needs.

Code:

```
subnet 10.0.0.0 netmask 255.255.255.0 {
   interface eth1;
   range 10.0.0.10 10.0.0.254;
   default-lease-time 6000;
   max-lease-time 7200;
   option subnet-mask 255.255.255.0;
   option broadcast-address 10.0.0.255;
   option routers 10.0.0.1;
   option domain-name-servers 192.168.0.1;
   option time-offset -3600;
}
```

This is the default settings for any machine that requests a configuration.

subnet 10.0.0.0: The network id

netmask 255.255.255.0: The netmask for the network, netmasks are often used to split ip ranges in smaller networks, but this one is a default one (for the wrong IP class but, meh ^^)

interface eth1: The network interface the server till be listening on.

range 10.0.0.10 10.0.0.254: The IP range it will deliver.

default-lease-time 6000: the default time a lease is held, the default time a computer can borrow an IP before the server checks if it still wants it. You should set this to a higher number, to reduce errors and traffic.

max-lease-time 7200: maximum time it can borrow an IP.

The options below are the configurations that the DHCP server will send to the client requesting the configuration.

option subnet-mask 255.255.255.0: This one was explained before ^^

option broadcast-address 10.0.0.255: The address the computer will broadcast on. Broadcast is used for different things, for example when the client wants to find another clients name, or if it wants a DHCP server to respond when it doesn't have one, it will send out a broadcast on the network.

option routers 10.0.0.1: The default router/gateway. A router is a node that divides a network in two basically, this is almost always your "broadband router". A router has 2 IP's for your information, but I wont explain that here.

option domain-name-servers 192.168.0.1: If you have a DNS server on the network you can use this option. Notice that the IP is of a totally different class then the ones given out by the DHCP. This is because that is the IP of a second router on my home network, which is the primary one that I use.

option time-offset -3600: Explains itself, note that the time offset is set in seconds.

These are simply specific configurations for some of my computers, most of them should be easy to understand now since I just explained them above. But I will explain those that I didn't

Code:

```
host Enma {
    hardware ethernet xx:xx:xx:xx:xx:xx;
    fixed-address 10.0.0.2;
}
host Teresa {
    hardware ethernet xx:xx:xx:xx:xx:xx;
    fixed-address 10.0.0.1;
    option subnet-mask 255.255.255.0;
    option broadcast-address 10.0.0.255;
    option routers 10.0.0.1;
    option domain-name-servers 192.168.0.1;
}
```

hardware ethernet xx:xx:xx:xx:xx:xx : this is the MAC address of the specified computer, this is very important for the DHCP server to know which computer is to be given the special settings.

Well, the rest, I leave to you now. Have fun, I had ^^

Optimize your mysql server

the mysql server bears a lot of optimization potential, but especially for beginners the config can be a pain in the rear. I found a real nice script that helps a lot finding optimized settings for your mysql server, here's how to get/use it:

Code:

```
#download it into /usr/bin so you can start it from #everywhere
cd /usr/bin
wget http://www.day32.com/MySQL/tuning-primer.sh

#give the proper permissions to run it
chmod 755 tuning-primer.sh
```

alright, done! you can run the script now from every location by typing *tuning-primer.sh*
keep your mysql password for the root user ready, if you don't want to enter it every time chose "yes" when it asks "Would you like me to create a ~/.my.cnf file for you?".

when running it you get a nice and colorful output, red color means there should/can be optimized something, the syntax for the optimized settings (if there is no default setting than simply have to be edited in /etc/mysql/my.cnf) can be googled easily.

for the proper calculation of memory usage "bc" have to be installed, should be part of the repositories for any package-based distro.

enjoy!

Remote MySQL

So ive got multiple computers in my possession, and now that school is out i have had some random things ive wanted to do, just to do them.

The first one was being able to remotely connect to a mysql database from one computer to another. The computer that im using to run MySQL is running Ubuntu Server edition. (its teh good shit!) The computer that im using to connect is running windoze. so here is how i did it! (this isnt the only way to do it, i just found this to be the fastest way to do it.)

Remotely connecting to a MySQL database may be effective if you are getting a lot of queries made by a lot of different people at one time. You will probably hardly ever need to do this, i did it just to say that i did to be fully honest. lol

so lets get to work.
You can go about this in two ways.

the first is to install mysql database and then use the MySQL login thingy that comes standard with ubuntu. i did not like this way, command line can be a pain in the rear, but this way is doable.

the second (the way i did it) was to simply install phpmyadmin and use that to run all of our queries. This makes it easier because we can use it to create databases, restrict users to certain databases and all of the other fun stuff.

so first install all of the crap youll need:

Code:

```
sudo apt-get install mysql-server
sudo apt-get install apache2
sudo apt-get install php5
sudo apt-get install php5-mysql
sudo apt-get install phpmyadmin
```

that will install all of the crap, php mysql apache phpmyadmin. pretty easy, but we arnt ready yet. you have to edit the apache config file to allow phpmyadmin to be used.
so run the following command:

Code:

```
sudo nano /etc/apache2/apache2.conf
```

now you need to add the following line somewhere inside that file:

Code:

```
Include /etc/phpmyadmin/apache.conf
```

save and exit.

now going from another computer on the network to:

(the computer's LAN ip adress that you jsut installed phpmyadmin on)/phpmyadmin

youll be asked to sign in to phpmyadmin. pretty easy stuff so far.

from withing phpmyadmin you can create a database or two. everything as you see fit.
for example i created a database named visser1.
now to allow remote acess to that database from my other computer i clicked on the tab in phpmyadmin that allows me to run MySQL queries. and i ran the following query:

Code:

```
GRANT ALL ON visser1.* TO root@"192.168.1.102' IDENTIFIED BY 'PASSWORD';
```

where:
192.168.0.102 is the LAN ip of the computer that is conecting remotely.
PASSWORD is the password
root is the user name of the remote connection.

now that we have set it all up we need to see if we can connect from the other computer.

because my other computer was running windoze i installed a program called navicat. it allows me to manage my databases and the such and it can come in handy!
so download and install it and create a new conection.
rather than 'localhost' use the LAN ip of the computer with the mysql database set up on it. type in th username and password that you set up then hit test conection. it will tell you if it worked or if it didnt. if it did then hit connect and boom. your done!

now if you were running a script from that computer and wanted to conect to the other computers database rather than typing in local host you would type in the LAN ip.

so thats it, your all set up. fun stuff.

Basic iptables setup

This guide will cover the basics of using iptables on a desktop linux computer. In the example that I show here I would like to note that all examples are done on a Ubuntu 7.10 computer.

What is iptables?

iptables is an application that allows administrators to configure the Netfilters. Basically it's a firewall application that can filter out packets on a very low level, making it very powerful.

Setting the rules

Ok so first of all you should know that iptables is located in "/sbin/iptables", as well as "iptables-save" and "iptables-restore".

Chain options

-I Insert: Insert a rule first in the table.
-A Append: Insert a rule last in the table.
-D Delete: Delete a rule.
-L List: List the tables.
-P Policy: Select a policy (chain) and either DROP or ACCEPT packets

Rule options

-j Target: What will happen if the rule matches? (ACCEPT/DROP/DENY?)
-p Protocol: What protocol will match? (ie...tcp/udp/icmp)
--dport Destination port: The destination port (22,80,21,8080)
-m Module: Selects a module, I will only use one in this guide...

The table has 3 default chains, which are the following:

* Input

* Forward

* Output

A typical set of rules (when written in bash) would look like this

```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -I INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A INPUT -i eth0
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT
```

This set of rules would do the following

1: Drop all packets by default on "INPUT" which would make the computer "Invisible" to other nodes.
2: Drop all the packets on "FORWARD"
3: Accept packets from the outside if a connection has been established.
4: Accept packets on the port 22 (SSH)
5: Accept packets from the localhost

*now, -P INPUT DROP, means that we chose the chain "INPUT" and that it should DROP all packets by default.
*-I INPUT means that we are inserting (setting a rule first in the table basically) and then loading the module (-m) "state" and setting it to "ESTABLISHED,RELATED" and then accepting the packets if it matches the rules (-j)

*-A INPUT means we are appending the chain "INPUT" (which basically means that we are placing a rule in the end of the table) and then accepting packets on port 22 with the "tcp" protocol (-p tcp --dport 22 -j ACCEPT).

you can now list your rules by using "iptables -L"

If you want to know more about how to set the chains and rules in iptables you can use the command "man iptables" which gives you the iptables manual.

Saving and restoring

A thing you should know about iptables is that they are reset when you restart the computer. There are ways to restore them though, and the applications for this is as mentioned before:

iptables-save
iptables-restore

To save the iptables you must do the following. After you have set the rules after your liking, you must write the following: "iptables-save > /iptables" which means that it saves the rules in the file "iptables" under root.

Now that your rules are saved you must be able to restore them as well...

Goto and edit "/etc/network/interfaces" by writing "nano /etc/network/interfaces"

The file should look something like this...

Code:

```
  GNU nano 2.0.6        File: /etc/network/interfaces

# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
#iface eth0 inet dhcp
```

To reset your iptables you have to use the command "iptables-restore", and by placing it in this file it will load when you boot up your computer. When placing the command in the file, it should look something like this...

Code:

```
  GNU nano 2.0.6        File: /etc/network/interfaces
```

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
pre-up iptables-restore < /iptables
auto eth0
#iface eth0 inet dhcp
```

Now just save it with "CTRL + O" and then "CTRL + X" to end nano.

There, that is all you need to know to get you going =)

LPIC-1

Some of you (me too) want to make the LPI exams, the LPI certifications are worth much more in the business world than any of the MS ones, one of the reasons is that those exams are not really easy. the needed knowledge is pretty extensive, so I wrote down what knowledge is expected for the LPIC-1, source is the official LPIC-1 exam book. _____

the exams are pretty cheap btw, about 100 bucks, on special events like Linux conventions exams are even offered for half the price. learning from books don't really makes sense, the best practice is to work on a Linux system....also learning together with others is useful, for example with your local LUG (Linux user group).
you can look for the nearest LUG here: http://www.linux.org/groups/
the LPI can be found here: http://www.lpi.org

ok, now to the content of the 1st exam, of course not ALL will be tested, the content of each test is generated randomly.

hardware and system architecture
- BIOS settings
- modem and soundcard configuration
- setup of non-IDE drives
- setup of different PCI cards
- setup of communication devices
- setup of USB hardware

Linux installation and package-management
- creation of HDD partitions
- installation of a bootmanager
- creation and installation of programs from source code
- management of shared libraries
- usage of the Debian package manager
- usage of the RedHat package manager

GNU- and Unix-commands
- working in a shell
list of important files, directories and applications:
bash
echo
env
exec
export
pwd
set
unset
~/.bash_history
~/.profile

process texts using filter programs
list of important files, directories and applications:
cat
cut
expand
fmt
head
hexdump
join
nl

sed
sort
split
tac
tail
tr
unexpand
uniq
wc

basics of file management
list of important files, directories and applications:
cp
find
mkdir
mv
ls
rm
rmdir
touch
File Globbing

usage of streams and pipes
list of important files, directories and applications:
tee
xargs
<
<<
>
>>
|
(back ticks)

starting, monitoring and ending of processes
list of important files, directories and applications:
bg
fg
jobs
kill
nohup
ps
top
killall

processing of textfiles via regular expressions
list of important files, directories and applications:
grep
sed

basics of data processing using the vi editor

creation of partitions and file systems
list of important files, directories and applications:
fdisk
mkfs
mkswap

ensuring data system integrity
list of important files, directories and applications:
du
df

fsck
e2fsck
mke2fs
debugfs
dumpe2fs
tune2fs

mounting and unmounting of file systems
list of important files, directories and applications:
/etc/fstab
mount
unmount

management of disk quotas
list of important files, directories and applications:
quota
edquota
repquota
quotaon

file access control via file permissions
list of important files, directories and applications:
chmod
umask
chattr

management of file owner settings
list of important files, directories and applications:
chmod
chown
chgrp

creation and editing of hard- and symlinks
list of important files, directories and applications:
ln

finding system files and placement of files in the right place
[i] list of important files, directories and applications:
find
locate
slocate
updatedb
whereis
which
/etc/updatedb.conf

installation and configuration of X11
list of important files, directories and applications:
xorgcfg
xorgconfig
/etc/X11/xorg.conf
XF86Setup
xf86config
xvidtune
/etc/X11/XF86Config
.Xresources

setup of a display manager
list of important files, directories and applications:
/etc/inittab
/etc/X11/xdm/*

/etc/X11/kdm/*
/etc/X11/gdm/*

installation and customization of a window manager environment
list of important files, directories and applications:
.xinitrc
.Xdefaults
xhost
environment variable DISPLAY

management of the kernel and kernel modules
list of important files, directories and applications:
/lib/modules/kernel-version/modules.dep
/etc/modules.conf
/etc/modprobe.conf
depmod
insmod
lsmod
rmmod
modinfo
modprobe
uname

configuration, creation and installation of customized kernels and kernel modules
list of important files, directories and applications:
/usr/src/linux/*
/usr/src/linux/.config
/lib/modules/kernel-version/*
/boot/*
make
make targets

booting, initialization, shutdown and runlevels
list of important files, directories and applications:
/var/log/messages
/etc/modules.conf
/etc/modeprobe.conf
dmesg
LILO
GRUB

changing runlevels, shutdown and reboot of the system
list of important files, directories and applications:
/etc/inittab
shutdown
init

management of printers and printer queues
list of important files, directories and applications:
CUPS-configuration files, -tools and utilities
/etc/printcap
lpc
lpq
lprm
lp

printing files
list of important files, directories and applications:
a2ps
lpr
lpq

installation and configuration of local and network printers
list of important files, directories and applications:
CUPS-configuration files, -tools and utilities
/etc/printcap
/var/spool/cups/
/var/spool/lpd/*/
lpd

using and managing local system documentation
list of important files, directories and applications:
CUPS-configuration files, -tools and utilities
MANPATH
man
apropos
whatis

finding Linux documentation on the internet

notification of users about system-related events
list of important files, directories and applications:
/etc/issue
/etc/issue.net
/etc/motd

using and customizing the shell environment
list of important files, directories and applications:
~/.bash_profile
~/.bash_login
~/.profile
~/.bashrc
~/.bash_logout
~/.inputrc
function
export
env
set
lists
seq
unset

writing and customizing simple scripts
list of important files, directories and applications:
for
while
test
chmod

administrative operations
list of important files, directories and applications:
/etc/passwd
/etc/shadow
/etc/group
/etc/gshadow
chage
gpasswd
groupadd
groupdel
groupmod
passwd
useradd

userdel
usermod

optimization of user environments and global environment variables
list of important files, directories and applications:
/etc/profile
/etc/skel
env
export
set
unset

using and configuring logfiles
list of important files, directories and applications:
/etc/syslog.conf
/var/log/*
logrotate
tail -f

automatizing administrative operations by planned jobs
list of important files, directories and applications:
/etc/anacrontab
/etc/at.deny
/etc/at.allow
/etc/crontab
/etc/cron.allow
/etc/cron.deny
/var/spool/cron/*
at
atq
atrm
crontab

designing of effective data backup strategies
list of important files, directories and applications:
cpio
dd
dump
restore
tar

administrate the system time
list of important files, directories and applications:
/usr/share/zoneinfo
/etc/timezone
/etc/localtime
/etc/ntp.conf
/etc/ntp.drift
date
hwclock
ntpd
ntpdate

TCP/IP basics
list of important files, directories and applications:
/etc/services
ftp
telnet
host
ping
dig

traceroute
whois

TCP/IP configuration and troubleshooting
list of important files, directories and applications:
/etc/HOSTNAME or /etc/hostname /etc/hosts
/etc/networks
/etc/host.conf
/etc/resolv.conf
/etc/nsswitch.conf
ifconfig
ifup and ifdown
route
dhcpcd
dhclient
pump
host
hostname
domainname
dnsdomainname
netstat
ping
traceroute
tcpdump

configuring Linux as PPP client
list of important files, directories and applications:
/etc/ppp/options.*
/etc/ppp/peers/*
/etc/wvdial.conf
/etc/ppp/ip-up
/etc/ppp/ip-down
wvdial
pppd

configuration and administration of xinetd, inetd and related services
list of important files, directories and applications:
/etc/hosts.allow
/etc/hosts.deny
/etc/services
/etc/xinetd.conf
/etc/xinetd.d/
/etc/xinetd.log
/etc/inetd.conf

running and basic configuration af a MTA
list of important files, directories and applications:
config files of Postfix, Qmail, Exim and Sendmail
/etc/mail/*
~/.forward
commands of the sendmail emulation layer
newaliases

running and basic configuration of Apache
list of important files, directories and applications:
/etc/apache2
httpd.conf
apache2ctl
apachectl
httpd

administration of NFS- and SAMBA daemons
list of important files, directories and applications:
/etc/exports
/etc/fstab
/etc/smb.conf
mount
unmount

setup and configuration of basic DNS services
list of important files, directories and applications:
/etc/hosts
/etc/resolv.conf
/etc/nsswitch.conf
/etc/named.conf
named

setup of the secure shell (OpenSSH)
list of important files, directories and applications:
/etc/hosts.allow
/etc/hosts.deny
/etc/nologin
/etc/ssh/sshd_config
/etc/ssh_known_hosts
/etc/sshrc
sshd
ssh-keygen

security-related administrative operations
list of important files, directories and applications:
/proc/net/ip_*
find
iptables
passwd
socket
nmap
netstat

basic security of systems
list of important files, directories and applications:
/etc/xinetd.d/*
/etc/xinetd.conf
/etc/inet.d/*
/etc/inetd.conf
/etc/nologin
/etc/passwd
/etc/shadow
/etc/syslog.conf

security on userlevel
list of important files, directories and applications:
quota
usermod
ulimit

Generate strong password in Linux

I just learned of a neat little program to generate strong passwords.

Debian, or other distrobution.

sudo apt-get install mkpasswd

To use type:

mkpasswd --char=10

This will generate a 10 char random password. Good luck remembering it. Wink

You can also try:

mkpasswd -s "insertwordhere"

This will take the characters from the entered word, and create a password from those char.

Ubuntu uses a different program in its repositories for some reason, but its called makepasswd, to install it s
type:

sudo apt-get install makepasswd

The program is used exactly the same way, except of course, you replace mkpasswd with makepasswd.

Now enjoy your new secure passwords and rid of your 'abc123'/'cathat' passwords today