Global Information Assurance Certification Paper

Secure Installation of

Linux Mandrake 8.0

For @Home Internet Use

GIAC Level Two Securing Unix
GCUX Practical Assignment
Version 1.7
Gene Feliciano Jr.

CONTENTS

Introduction

Linux has exploded in popularity since its kernel was posted to USENET in 1991 by then University student Linus Torvald. Today it is a viable alternative solution for many home users who want to use something other than Microsoft based operating systems with their high speed DSL and cable connection. This paper's objective is to guide the novice cable modem user who wishes to "try something different" though a complete installation of Mandrakes' distribution of Linux version 8.0. The only thing assumed is that the reader has some knowledge of a text editor under Linux. The editor I prefer is vi, not because it the best but because it is available on all flavors of Unix. A quick tutorial is provided as Appendix A.

The example machine I'm using for the install:

Processor PIII 750mhz
BIOS Award 8/16/2001
384 Megs SDRAM
Removable 10 GIG Maxtor IDE Drive
3dx Voodoo 3 video card
40 Speed ATAPI CDROM
IOMEGA ZIP Drive (100megs)
One Realtek RTL8029 Ethernet LAN Card

Before starting the installation, think about what functions the system is going to perform for you. For instance, are you going to write code with it or just surf the web, chat with friends or play online games, student activities or home office etc. In many cases what you plan on using your system for will decide on the level of security you will need. For example:

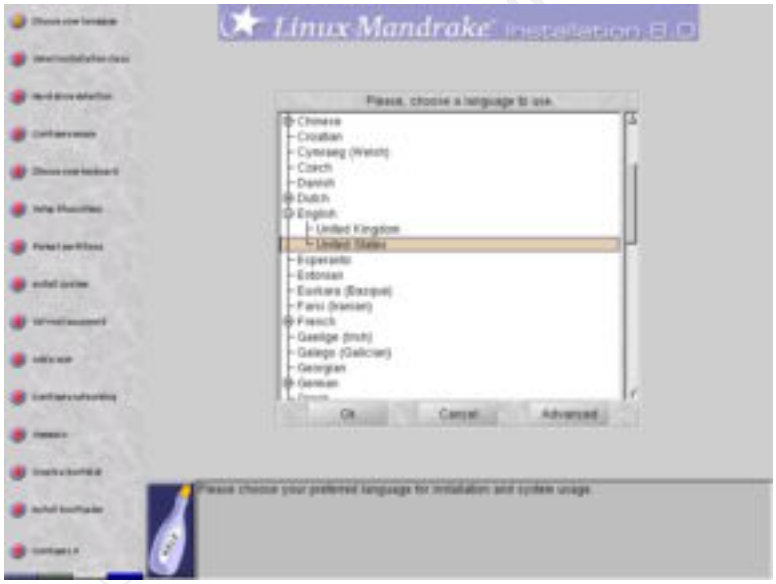
Function:

Home workstation
C, C++ and Perl development and course work
Web Browsing
Email (client)
FTP (Client)
SSH from work (Server)
USENET (client)
Word processing

Once the purpose of the machine is done its time to put this aside for now you have a bases for planning the install and the security requirements for the system. This where balancing risk and paranoia becomes an art form. Always remember, the more you lock down a system the less "user friendly" it will become.

INSTALLATION

1. Create boot disk using images or use boot disk that comes with OS. If your Cdrom drive allows booting a boot disk is not required, just place the first CD into the reader and reboot.
2. Set a bios (basic input/output system) password for the system. On home systems setting a bios password can be accomplished by accessing your system bios during boot up usually by using one or more key(s) like “ctrl-del”, del, F5 or F10. A menu system will popup list many hardware setting and a setting for bios passwords. Each PC is different but the password setting is standard on most chip sets.
3. Disconnect machine from the network. You have a choice here. You can disconnect your system from the LAN (100% safe) or install your system behind a firewall/gateway machine (not as safe). Security professionals from many organizations including SANS, Usenix, and Foundstone recommended for the truly paranoid to deploy a sniffer/IDS system on the network segment installations are being done from. Once again this is truly a matter of choice. For our home system installation we will disconnect the system since in most cases a home user only has one PC.
4. Press enter to install a system running Mandrake. (Installation will begin though DrakX, the Mandrake GUI installation program)



(* DrakX)

5. Next choose the means of installation. For our purposes this will be cdrom.
6. Please choose your preferred language for installation and system usage. Language to use English (United States) for this installation guide.
7. License and legalize window (etc. etc.) click <Accept>.
8. Installation class (Recommended or Expert) Choose Expert. Recommended doesn’t allow software selection or a choice of partition sizes. Expert provides more control over the software installation.

9. Do you have scsi devices? For this install our Dell PC does not use a scsi harddrive adaptor. Mandrakes' DrakX will normally auto detect one if it is part of your hardware configuration at this step. Click No.
10. Mouse configuration window. The standard mouse driver will work for most installations. Please choose a driver here and click the okay button.
11. Keyboard Configuration window. US Keyboard or US keyboard (International) are the choices. Select US Keyboard.
12. Choose security level. High is recommended. More on the security level will be explained during the Bastille Linux configuration. For now choose High.
13. Partition your hard drives. Hard drive partitioning is a religious subject better left to others. This installation assumes you will be using the entire harddisk. If you're going to have another OS (called duel booting the system) you will see additional partitions in this window. Example: Win9x/Linux, Windows 2000/Linux or Dos/Linux please refer to your Linux manual for instructions on how to plan and configure your system. (For more information on what partitions to use and how big they should be please refer to your manual or one of the following links to installation and documentation sites:

    http://www.geocities.com/aboutlinux/
    http://www.linuxdoc.org/
    http://www.ugu.com/
    http://www.ibiblio.org/mdw/HOWTO/mini/Partition/index.html

    In any case at a minimum you should have a root partition "/", swap, and home. Servers and specialized systems (IDS, development, etc.) will require additional partitions.
14. Format the partitions you just created. (This is the point where any previous data on the drive will be destroyed.) Please make sure you know what you are doing and have any data you wish to keep backed up. The advanced options will check the drive for bad blocks.
15. CD check. Mandrake Standard addition comes with two CD's. This is to verify that you have both CD's for the installation before proceeding to package selection.
16. Package Group Selection. Mandrake 8 comes with predefined package groups to help you decide what type of system you are configuring in two columns labeled Workstation and Server. At this point what you decided will impact the security of the system. Generally selecting package groups saves time but results in the installation of packages you may not wish. I recommend de-selection of all package groups and the selection of individual packages (located at the bottom of the window). True this will require you to filter though numerous packages but will help in the over all security of the system by installing only the software required to fulfill your purposes.
17. After selecting the "individual package selection" all of the packages available are displayed in a expanding tree format in 4 main categories:
    a. Workstation
    b. Graphical Environment

c. Development
d. Server

Click the plus sign to the left to expand a category or subcategories for selection. Since our objective is to place a secure workstation on the Internet the following packages are what I picked for my home system.

Working from the bottom up:

Server category

Server>Firewall/Router>Other
iptables * requires you to search for the rpm
Server>Network Computer server
openssh-server
postfix-server
bind-utils *requires you to search for the rpm

Development category

Development>Documentation
howto-html-en (Great source of information)
lilo-doc
bash-doc
mandrake-doc-en (Another great source of info)
Development>Development
gcc packages
gcc-c++ packages
perl packages
cvs

Graphical Environment

Graphical Environment
kdebase
kdeadmin
kdenetwork
kdeutils
ktelnet

Workstation

Workstation>utilities
dump
lsof
nmap
procinfo
tcpdump
traceroute
unzip
Workstation>utilities>Other

logcheck
Workstation>Console Tools
file
ispell
mc
screen
slocate
symlinks
vim packages
vlock
Workstation>Configuration
userdrake
Workstation>Configuration>Other
Bastille-Tk-module
timeconfig
Workstation>NetworkComputer(client)
telnet
openssh-client
Workstation>Internet station
mailx
ncftp
netscape-communicator

Click install to begin the installation of your Mandrake system.

Next we see a warning screen about the ssh and postfix servers we installed click yes. This warning is just to inform you that you have installed software used for servers and that users can connect to these services over the Internet. We are going to turn off the daemons (services) a bit later here during the install.

Set the root password

The corner stone to your systems security will be the password you choose for your root and user accounts. Mandrake uses the shadow suite PAM (Pluggable Authentication Modules), and md5 hashing by default. What this means to you is that passwords are not stored in the /etc/passwd file (which has read write permission for owner and read for the group and world) but in the more secure file /etc/shadow (read only for owner) and these passwords are encrypted with the stronger algorithm md5 verses the old standard des used on older Unix versions.
Under /etc in the file login.defs is the settings for password aging. By default Minimum acceptable length is set to 5 at installation. After setting the password you can change this setting to fit your needs.
A word on setting good secure passwords

Mandrake uses the PAM software to help you choose a better password but still not as secure as following these simple rules.

- Passwords should not be your kids, wife or any person's name. This includes derivatives like Sam01 or M1keH00k.
- Passwords should not contain English words popular phrases. THisbuds4u
- Passwords should not be made up of SSN's, birth dates, favorite movies or your favorite hobbies. For example NFlf00t8all.

A good password is made up of a mix of upper and lower case letters, numbers and any of the other keys found in the ASCII character set. For info on the ASCII chart check this link out.
http://www.asciitable.com/
A favorite way of creating good passwords is to make an acronym out of a sentence picked out of a book or magazine. Example "What is this ~ mean to you?"
w1T~mt0U

Create user accounts---at least one at this time

Common practice is to create a user account for yourself here that you will use for day-to-day activities. The root account will only be used to install/compile software and make changes/additions to the system. Never use the root account to conduct normal user activities. The root account has the preverbal keys to your kingdom so use it wisely.

Network Configuration Wizard

This applet will start at this time and help you configure your connection to the Internet. Follow these steps:

1. Select auto-detection
2. Select LAN connection (It should be auto-selected for you).
3. Your network card should be found
4. Static IP address (add your IP address here)
5. Netmask (255.255.255.xxx)
6. Hostname (system@home.com)
7. DNS server (IP address of your ISP's DNS server)
8. Gateway (IP of the Internet gateway router provided by ISP)
9. HTTP proxy (Add web proxy server's address here)
10. FTP proxy (Add ftp proxy server's address here if your ISP uses one @home currently does not.)

Summary Check

Final systems check of some of your systems resources before continuing on with the configuration. You can make changes here for:

- Mouse
- Keyboard
- Timezone
- Printer

Services Check

Choose service that you wish to start at boot time. Here is where you have a last chance to look at the services that will be running on your system at boot time.

- Disable the sshd and postfix servers. * We will configure them later.

Create Custom Boot Disks Window

Here you will create boot disks for your system in case something goes wrong and you need to boot from disk. These are quite handy and a must when things break.

Boot loader main options

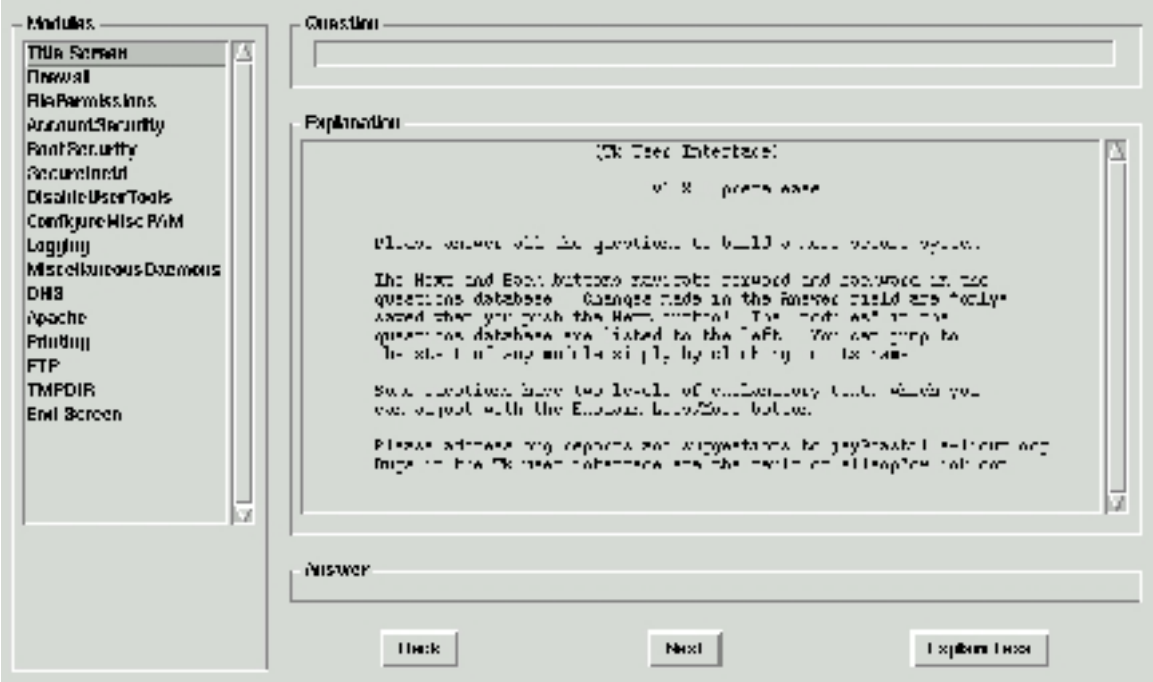
- Add clean /tmp directory under the advanced tab after every boot.
- Accept the default for now. * We will configure this later.

X Configuration (depends on your video card and monitor)
Mandrakes Drak X auto configuration program is outstanding at setting up the right setting for most modern day video cards and monitors. If you have problems setting up your X windows please consult the Installation Manual or the Linux Documentation project's link given earlier in the paper.

OS Hardening before Connecting to the Internet

Bastille Linux

Bastille Linux Hardening system is a must have package that runs on the Red Hat and Mandrake distributions and comes as an rpm package with your distribution. The Bastille Linux script takes knowledge from many Unix/Linux security sources including SANS Step-by-Step guide, Kurt Seifried's Linux Administrator's Security Guide, others to ask the user a series of questions via a gui (Graphic User Interface) menu system to create a script that hardens the operating system. The version provided with Mandrake Linux (currently V1.2.0) can be run before connection to the Internet as well as any time after. To start the program open a terminal emulator window and at the prompt type the following commands at the command prompt:
/bin/su - *Note Bastille must be run as the root user *
/usr/sbin/InteractiveBastille



Title Screen

Gives you the latest version number (Currently version v1.2.0 as of 6/11/01).

Firewall

Click yes here to run the iptables packet-filtering script for 2.4 kernels.

1. Do you need advanced networking options? (No)*
   - ❖ If we were using this machine as a gateway we would click yes here.
2. DNS Servers (0.0.0.0/0). Here we can leave the default because we are using the iptables firewall script. The Bastille script will read the /etc/resolv.conf file for DNS servers to query for answers.
3. Public interfaces (eth0). Since we only have one Ethernet card eth0 or eth+ can be used.
4. TCP services to audit (telnet ftp imap pop3 finger sunrpc exec login 98 ssh). These are the default services that the kernel audits to the syslog service. For our purposes this is fine. Additional ports can be added as needed. *Note in the /etc/services file port 98 is listed as tacnews. This port is also used by linuxconf so it is best to type the port number instead of the service name.
5. UDP services to audit (31337). This is the Bastille default. I like to add additional ports here as well as in step 4 as Trojans and other vulnerabilities are discovered.
6. ICMP services to audit. (echo-request) Add any others you wish to audit. I always add the echo-request because I like to know who is pinging me.
7. TCP Services to allow on Public interfaces (22). SSH is the only service we are going to allow the public to see.
8. UDP Services to allow on Public interfaces (). Typical workstations will have none here.

9. Force passive mode? (Yes) This setting forces your FTP clients to use passive mode FTP rather than active mode FTP. Passive mode FTP makes it easier to block "high" specific TCP services. Warning this will cause you to configure some of your client software packages to use passive FTP services.
10. TCP services to block (2049 2065 2090 6000:6020 7100). Since we used passive FTP in step 9 we do not have to worry about blocking those ports. Bastille recommends that you run lsof –i as root to check which ports are listening.
11. UDP services to block (2049 6770). As with step 10 check the output of lsof –i run as root to check for any UDP services.
12. ICMP allowed types (destination-unreachable echo-reply time-exceeded) Use the default here. This allows you to use ping a traceroute to solve connectivity issues.
13. Enable source address verification (Yes). This configures the kernel to block traffic likely to have spoofed IP addresses.
14. Reject method (DENY). This setting tells the kernel how to reject blocked traffic. Setting "REJECT" is considered friendlier but will allow you to become visible to random scans because it lets the remote host know you rejected the connection. DENY drops the connection altogether.
15. Interface for DHCP queries. (). Since we are using static IP addresses we do not need to add anything here. If we were using DHCP we would add eth0.
16. NTP servers to query (). Since we are only using the system clock and don't really have a need to be accurate no IP addresses are added.
17. ICMP types to disallow outbound (destination-unreachable time-exceeded). These are set as default and will stop standard traceroutes against you.
18. Should Bastille run the Firewall and enable it at boot time? (Yes) This setting will insure the Bastille Iptables script will start when the system is booted. Later we will check our setting by probing the firewall with a scanner to test our rule set.

File Permissions

1. What security level should we set? (4) During the installation we set the file and directory permissions to level 4. Bastille recommends that we place the same setting here as well. Generally this setting will repeat your previous choice. Note. The Drake installation refers to the security settings as low, medium and high. The security chapter breaks down the setting into six levels from 0 to 5 (Zero being omitted on the Bastille script). A copy of the table has been provide for reference. Figure x. For complete detail on each feature please refer to the Mandrake Setup Installation Manual.

Table of Mandrake security levels[1]

Feature/Level	0/Crackers	1/Poor	2/Low	3/Medium	4/High	5/Paraniod
Global security check			Yes	Yes	Yes	Yes
Umask/Users	002	002	002	002	077	077
Umask/root	002	002	002	002	002	077

[1] Table Copied from Linux Mandrake Reference Manual

Shell w/o password	Yes					
Can connect to X display	All	Local	Local	None	None	None
User in audio group	Yes	Yes	Yes			
In $PATH	Yes	Yes				
Warnings in file var/log/security.log		Yes	Yes	Yes	Yes	Yes
Warnings directly on tty			Yes	Yes	Yes	Yes
Warnings in syslog			Yes	Yes	Yes	Yes
Warnings send by e-mail to root			Yes	Yes	Yes	Yes
Suid root files check			Yes	Yes	Yes	Yes
Suid root files MD5 check			Yes	Yes	Yes	Yes
Writable files check				Yes	Yes	Yes
Permissions check				Yes	Yes	Yes
Suid group files check				Yes	Yes	Yes
Unknown files check				Yes	Yes	Yes
Promiscuous check				Yes	Yes	Yes
Listening port check				Yes	Yes	Yes
Passwd file integrity check				Yes	Yes	Yes
Shadow file integrity check				Yes	Yes	Yes
System security check every day at midnight				Yes	Yes	Yes
All system events additionally logged to /dev/tty12				Yes	Yes	Yes
Only root can ctrl-alt-del					Yes	Yes
Unknown services are disabled					Yes	Yes
Boot password lilo/grub					Yes	Yes
Grants connections from	All	All	All	All	Local	None

2. Would you like us to modify your file permissions? (Yes) This setting will change permission on many of the utilities used in system administration so that only root can use them (Examples give are linuxconf, portmap and ifconfig).
3. SUID root status for particular programs. If you have any of these programs Bastille will list them here. You may consider disabling them for security reasons or not disabling for user convenience. Generally these programs are looked at by blackhats to find weaknesses or vulnerabilities. If you don't know what these programs do just answer "Yes" for now and rerun this tool if you have broken software you are using.

Understanding the basic file permissions should be a first priority for any Linux user. Please refer to the Reference manual, Chapter 2 Unix Basics, for an over view of file permissions.

Account Security

1. Would you like to enforce password aging? (Yes) This changes the default behavior on Mandrake as well as Redhat boxes from disabling passwords in 99,999 days to something more sensible, 180 days with a 7-day window warning the user that the password is going to expire. . This is done in the file /etc/login.defs.
2. Would you like to restrict the use of cron to admin accounts? (Yes) The file /etc/cron.allow is created. Since cron can be abused this practice gives you access control.
3. What umask would you like to set for users on the system? (077) This setting is the most secure making all files created by a user write and readable by that user and root only.
4. Should we allow root to login on tty's 1-6? (No) This makes it a little harder for someone to access your system if they have been able to obtain your root password. It will require them to also have the password of a user account to login into the system and then use /bin/su to become root. This can be a pain if you need to do lots of work as root on a system. In that case change your answer to yes.
5. Should we allow the PATH to include the current directory? (No) There is a security risk with being able to run programs just by typing the command at the prompt instead of type the full path or "./program.exe". It is a good practice to always type the full path of commands when running them. Trojan executables are one of the favorite ways to get users to execute code of the hackers choosing. For example, many administrators like to type su- instead of the full path /bin/su- to become superuser. A friend of mine (who shale be nameless) took a small script he had read about in a Unix scripting course and placed it in the directory path of the admins do to some loose directory permissions on an NFS server. The script follows:

```
stty –echo
echo " Password: \c"
read X
echo ""
```

```
stty echo
echo $1 $X | mail drguthr@home.com &
sleep 1
echo Sorry.
rm su
```

He was able to get the root password mailed to him on three different servers before it was noticed.

6. Should we deactivate this list of users? (yes/no) This setting is a minor security issue into whether you wish a graphical icon list of users displayed who can login into the system or not. If your system is not starting X then this setting is a no.
7. Would you like to password protect the LILO prompt? (Yes)

   LILO or the Linux loader is one of the more popular boot loaders available for Linux. Its purpose is to give the user a menu of booting options for the system. By default the Mandrake installation LILO menu will wait for 5 seconds before loading the default flagged operating system. Newer versions list the menu options on the screen for the user to pick while older versions just display a prompt.

   LILO Boot:

   Bastille adds the following the following after the "prompt" line in /etc/lilo.conf.

   ```
   password = "your password here"
   restricted
   ```

   Save lilo.conf
   Since the password is saved in clear test please verify that the permissions on the file lilo.conf are rw-------
   You can do this by run the command:

   ```
   ls –la /etc/lilo.conf
   ```

   The output should be something similar to the following:

   ```
   -rw------- 1 root    adm          402 Sep 11 15:46  /etc/lilo.conf
   ```

8. Would you like to reduce the LILO delay time to zero? (Yes) This setting makes it harder to type anything at the LILO prompt.
9. Do you ever boot Linux from the hard drive? (Yes) This setting writes the LILO configuration to hard disk.
10. Would you like to write the LILO changes to a boot floppy? (Yes) This one updates the boot floppy we created earlier.

11. Would you like to disable CTRL-ALT-DELET rebooting? (Yes)

The /etc/inittab file, as defined from the man page, describes which processes are started at bootup and during normal operations. Bastille makes the following modification to this file.

Adds a "#" sign before the following line:
ca::ctrlaltdel:/sbin/shutdown –a –t3 –r now

So that it looks like
#ca::ctrlaltdel:/sbin/shutdown –a –t3 –r now

This setting is really more important for systems that are going to be used as servers but I have found it to be helpful for keeping family members from reboot the system to the Windows partition without my knowledge.

12. Would you like to password protect single –user mode? (Yes)
    This is the second modification Bastille makes to the /etc/inittab file. By default Mandrake does not require a password to get to the single user mode. If your system is powered down and forced into single user mode root access will be granted without the password. For just pain common sense to never leave an easy way to get around your systems security by leaving a way to obtain root privileges without knowing the password.
    Bastille adds the following line after the "si" option in the inittab:

    ~~:S:wait:/sbin/sulogin

13. May we disable Autologin? (Yes) This is just plain stupid. You should never allow a login to your system without a password and autologin would enable this. With autologon anyone that knows what your logon is would be you on the system!
14. Would you like to set a default deny on TCP-Wrappers? (Yes) OPENSSH if installed via RPM uses this library. More on this when we configure the SSH service.
15. May we deactivate telnet? (Yes/No) Telnet uses a cleartext protocol that can easily intercepted by blackhats. You run a risk using it. In some cases it is the only avenue you have so this one is up to you. Since we are not using inetd or xinetd in this installation it really doesn't matter what you answer here.
16. May we deactivate ftp? (Yes/No) This is the same as #15 and the setting pertains to running an ftp daemon.

Disable User Tools

1. Would you like to disable the compiler? (No) Since we are going to be using this workstation to write code it is more convenient to answer no here. However, hackers love to compile rootkits and other nasty programs on host machines and disabling the compiler may slow them down a little giving you a better chance to

noticing the activity. In any case, Bastille only changes the permissions to the compiler here so that root is the only one that can access it.

ConfigureMisc PAM

2. Would you like to put limits on system resources usage? (No)
   This setting makes changes to the /etc/security/limits.conf file and will set the following limits:
   - Core files to zero
   - Individual users are limited to 150 processes each
   - Individual users are limited to 40 mb of space.

   Since this is a home machine we will not add this setting. Warning if you do answer yes here you will have to manually edit the limits.conf file to undo them.
3. Should we restrict console access to a small group of user accounts? (No)
   Since this is a home machine restricting access to a few accounts is over kill. This is more suited to server operations.

Logging

4. Would you like to add additional logging? (Yes) This setting modifies the syslog service to add logging to the 7th and 8th virtual terminals (you would see this by typing the command ALT-F7 and ALT-F8) as well as two additional logging files to the setup /var/log/kernal for kernal messages and /var/log/syslog for warning and error messages. More logging is always a good thing.
5. Do you have a remote logging host? (No) If you have another machine you can send syslog data to this is a great way of insuring you log data is true data. In most cases home users will not have a remote-logging hosts available. If you do say yes here and add the IP address of the host at the next screen. Hackers will in most cases wipe the logs or clean them to cover there tracks on hosts they have compromised.
6. Would you like to set up process accounting? (No) This setting is rather disk and CPU intensive and will eat up a lot of space in a hurry. Bastille recommends that you answer no here even though the data gained here can be very helpful in reconstructing a break-in. Installing log rotation is also a must here and a module is included to help keep the size of the logs down.

Miscellaneous Daemons

1. Would you like to deactivate the routing daemon? (Yes) This a safe bet since we are not going to be using routing or require the routing daemons to be active on this workstation. In most cases workstations connected to the Internet by one connection type (Modem, Ethernet Card) will not need a routing daemon.
2. Should we disable most chkconfig'd services? (No) Bastille recommends this setting only for the extremely paranoid. In most cases it is better to say no here.

TMPDIR

1. Would you like to install TMPDIR/TMP scripts? (No) In multi-user environments many programs tend to use the /tmp folder in ways that can be dangerous. These scripts are a workaround to this vulnerability. For our purposes here it is not necessary.

End Screen

At this point you are asked if you wish to implement the settings you have configured. Answer yes and apply configuration to the system. If all went well you will see a credits screen with a button to exit the program. Any errors that occurred running the script will be displayed in your consol window. Reboot the machine for all changes to take effect.

Bastille creates a directory under /var/log named simply enough Bastille. Here is a set of logs and an undo script that attempts to remove all of the changes made. Note as of this version some changes are not automatically removed. You may have to manually edit the configuration file by checking the action-log or running though the gui front in to see which file was changed.

Finally before connecting your machine to the Internet run the following commands as root from the command prompt.
lsof –i or netstat –atp

You should get something similar to this.
lsof -i

```
COMMAND  PID USER   FD   TYPE DEVICE SIZE NODE NAME
kdm     1156 root    4u  IPv4  13385       UDP *:xdmcp
kdm     1156 root    5u  IPv4  13386       TCP *:blackjack (LISTEN)
X       1165 root    1u  IPv4  13390       TCP *:6000 (LISTEN)
kdm     1171 root    5u  IPv4  13386       TCP *:blackjack (LISTEN)
kdeinit 1257 root    6u  IPv4  13559       TCP *:1026 (LISTEN)
```

These are the services listening on your system that can be scanned from the Internet if you were not using a packet filtering firewall or other access controlling software like xinetd and portsentry. For more information on lsof and netstat please check the man pages by typing the command “man ” followed by the command you wish to look up.

Now run
iptables –L

```
Chain INPUT (policy DROP)
target     prot opt source               destination
DROP       tcp  --  anywhere             127.0.0.0/8
ACCEPT     all  --  anywhere             anywhere           state
RELATED,ESTABLISHED
ACCEPT     all  --  anywhere             anywhere
```

```
ACCEPT     all  --  anywhere             anywhere
DROP       all  --  BASE-ADDRESS.MCAST.NET/4  anywhere
DROP       all  --  anywhere             192.168.0.0/16
DROP       all  --  192.168.0.0/16       anywhere
PUB_IN     all  --  anywhere             anywhere
INT_IN     all  --  anywhere             anywhere
DROP       all  --  anywhere             anywhere
```

* Output cutoff to fit

This command will list out your firewall rules and show you that your firewall is up and running. For more information on writing your own rules and the iptables firewall please use the following links.
http://people.unix-fu.org/andreasson/index.html Iptables Tutorial with examples (one of the best I have seen)
http://t245.dyndns.org/~monmotha/firewall/index.php MonMotha's Iptables Firewall Script. A well commented bash script that attempts to make setting up iptables easier than other software packages.

Initial Backup
At this point I like to create a backup of configuration files and dump them to zip or tape drive. At a minimum backup the /etc folder here although a full backup would be preferred. Mandrake comes with many standard unix backup utilities like tar, dump, kbackup, ark and so on. To keep it simple do the following from the command prompt:

```
# cd /
  tar –cvf  etcsep21.tar /etc
  gzip etcsep21.tar
```

Now you have a backup of most of your initial configuration files stored in one tar archive. Move the file off your system and store it on tape or zip drive.

Now it is time to make the first connection to the Internet.
Plug the twisted pair cable into the cable modem.

What needs to be Done After Connection

Checking The Network Connection
Run ifconfig.
This command will check the connectivity of your devices and will provide output that looks something like this:
ifconfig

```
eth0      Link encap:Ethernet  HWaddr 00:00:86:5C:50:E4
          inet addr:192.168.0.2  Bcast:192.168.0.255
          Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:619 errors:0 dropped:0 overruns:0 frame:0
```

```
          TX packets:3 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          RX bytes:196890 (192.2 Kb)  TX bytes:199 (199.0 b)
          Interrupt:3 Base address:0x300

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:16 errors:0 dropped:0 overruns:0 frame:0
          TX packets:16 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1568 (1.5 Kb)  TX bytes:1568 (1.5 Kb)
```

We now have two interfaces, eth0 our network card, and lo our loopback device enabled. If you only have lo up then there is a problem with your Ethernet device configuration and you'll either have to manually start it using ifconfig (See man page) or run network configuration applet located in the Mandrake Control Center.

Ping your gateway.
Ping (Packet InterNet Groper) is a versatile utility for troubleshooting network connectivity issues. For more information see the man page.

```
# ping -c 5 192.168.0.1


PING 192.168.0.1 (192.168.0.1) from 192.168.0.2 : 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=0 ttl=128 time=1.022 msec
64 bytes from 192.168.0.1: icmp_seq=1 ttl=128 time=982 usec
64 bytes from 192.168.0.1: icmp_seq=2 ttl=128 time=1.019 msec
64 bytes from 192.168.0.1: icmp_seq=3 ttl=128 time=1.002 msec
64 bytes from 192.168.0.1: icmp_seq=4 ttl=128 time=967 usec

--- 192.168.0.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.967/0.998/1.022/0.035 ms
```

Okay now that we have connectivity it time to configure a security update site and begin patching our system.

Configuring A Security Update Site

1. Click on the Mandrake Control Center Icon located on your desktop.
2. Expand the System Icon and click on the Services Icon.
3. Click on the Software Manager.
   The Window Configure a source pops up requesting a source on the Internet for security updates.

4. Click Yes.
5. Click the button "Update the list of mirrors" A list of Mandrake mirrors will be updated an on can be selected from the pull down menu. Pick the closest one to you.
6. Click Okay.
   A list of Patches will appear.

7. Begin patching your system.

Dependency messages may appear for some packages. What this means is for the upgrade to work properly another package must be installed. Keep the CD's close by because this tends to happen quite a lot and is normal behavior.

Note kernel packages will also appear in this list. **Do not** install kernal packages from MandrakeUpdate! Read the install instructions that explain the reasoning for the update, decide if they apply to your system, if they do follow the checklist.

If you use MandrakeUpdate, the verification of md5 checksum and GPG signature is performed automatically for you. If you would rather install the rpm packages yourself you can download them from a list of Mandrake mirror FTP sites around the world.

You can manually check RPM packages to insure their integrity by typing

```
# rpm --checksig package.rpm

PGP and GPG public keys for the Mandrake security team are available
for download at the following link.

http://www.linux-mandrake.com/en/security/
```

A warning listed in the Mandrake Users Manual that should be taken to heart. The paragraph is copied here verbatim.

> You may also receive a message, telling you that a package is not signed or has an incorrect signature. The signature is used to make sure a package can be safely installed, from a security point of view: it has been validated by someone, and not altered by a malicious user. If you get this message, then make sure you know where the package comes from. You can install it, but do it at your own risk!

It doesn't happen often, I have only had the above message three times since I started using Mandrake a few years ago but in those cases I went to another mirror to download the RPM just to be on the safe side.

SSH Configuration

SSH (Secure Shell) is the secure encrypted replacement for the older "clear text" telnet, rlogin, rsh, rcp, and rdist connection tool. Mandrake 8.0 comes with the openssh version provided by the folks that bring us Open BSD another free version of Unix. SSH provides the ability to remotely connect to the system from anywhere (The only requirements are client software and access rights). On this system we have installed the ssh client and server packages. Let configure the server piece first.

SSHD

The server piece of SSH is run by the daemon sshd and can be started at boot time by the /etc/rc scripts. Before we do that we are going to take a look at the /etc/ssh/sshd_config file and make some minor changes being as the default setting are perfect for what we are using this system for.
Starting for top to bottom make the following changes:

1. Port 22
   Normally ssh uses port 22 so you can leave it as such. But if you truly paranoid you an make it any port you wish that isn't in use on your system. For example port 465 (smtps) is one that I use every now and then just to make it a little harder trying to find out what I'm using. *Note if you change your port setting you will need to rerun the Bastille script and change the port in Step #7 on your firewall script.
2. Protocol 2
   The default setting has protocol 2 and 1 set as fallback for compatibility. SSH1 has had some problems in the recent past and the new ssh2 protocol is now the preferred method in security circles. For more information on ssh's security history consult the Cert Vurnerability data http://www.kb.cert.org/vuls and get on their mailing list. A lot of great information is located at this site.
3. LoginGraceTime 120
   The default time is set to 600 seconds and in many cases that is way to long. I personally like two minutes. Find what is right for you.
4. PermitRootLogin no
   Login to your system as an unprivileged user first and then su to root. Never login to your system as root at anytime.

SSHD has many other settings and configurations. Information can be found using the man page for ssh, sshd and by reading the package documentation located in the directory /usr/share/doc/openssh-2.9p1 as of this writing.

SSH and TCP Wrappers

Since SSHD was compiled to use the /etc/hosts.allow and /etc/hosts.deny files of TCP Wrappers fame we are going to make a few changes to both files.

In the /etc/hosts.deny file we are going to verify that the Bastille hardening script had done it's job and place the following line in the file:

ALL : ALL

Note on some versions of Bastille nothing is placed in this file and a statement denying all connections and using the finger utility to gather information on who is trying to connect to your system is used in the /etc/hosts.allow file. Delete or comment out the settings added by Bastille in the /etc/host.allow file. I prefer to start with a deny everyone and allow few mentality. For more information please read the white papers on

tcpwrappers written by the guy how created the software at ftp.porcupine.org/pub/security/index.html Wietse Venema.

Next we edit the /etc/hosts.allow file and add:

sshd : 192.168.0.10

The service being allowed (sshd) followed by a colon and the ip address of the system you are allowing to connect to your service (192.168.0.10). Additional addresses and domain can be added as need by using commas.
sshd : 192.168.0.10, 127.0.0.1, joe.home.com.

SSH Client

Tweaking the ssh client is quite easy and straight forward. The file is located in the same directory as the server daemon /etc/ssh/ file name is ssh_config. An example configuration is displayed below.

```
#     $OpenBSD: ssh_config,v 1.10 2001/04/03 21:19:38 todd Exp $

******* data omitted *****
# Site-wide defaults for various options

# Host *
#   ForwardAgent no
#   ForwardX11 no
#   RhostsAuthentication no
#   RhostsRSAAuthentication yes
#   RSAAuthentication yes
#   PasswordAuthentication yes
#   FallBackToRsh no
#   UseRsh no
#   BatchMode no
#   CheckHostIP yes
#   StrictHostKeyChecking yes
#   IdentityFile ~/.ssh/identity
#   IdentityFile ~/.ssh/id_dsa
#   IdentityFile ~/.ssh/id_rsa
#   Port 22
#   Protocol 2,1
#   Cipher blowfish
#   EscapeChar ~

Host *
  ForwardAgent yes
  ForwardX11 yes
  Cipher blowfish
  Protocol 1,2
    StrictHostKeyChecking ask
```

Items to change:

1. ForwardAgent No
   Default setting for this item is no in the man pages but for some reason is yes in the configuration file. Change it to no as it is not needed.
2. ForwardX11 No

Using X window tools though your' ssh connection doesn't challenge security but may cause lag during your session. No is recommended.

3. Cipher blowfish
   This is a personal preference based on encryption speed over heavy usage periods. All the other encryption method can be used here.
4. Protocol 2
   This setting is to match the sshd daemons protocol settings. If you connect to other systems you may have to add a comma and a 1 to this. To check this use ssh –v hostname.
5. StrictHostKeyChecking ask
   When ssh connects the first time to a host a key exchange occurs and host keys are added to your /home/<user>/.ssh directory. When "ask" is set you will be prompted to accept or reject the servers' keys. Once you have done this you should never see this again. If you do it's either because you regenerated the servers keys or something malicious is underway. Trojan sshd servers have been found in the wild and this behavior is a good sign that something has gone wrong with the keys. For more information of Trojan attacks refer to the security book list in the bibliography "Hacking Linux Exposed".

Starting The SSH Server.

Click on your Mandrake Control Center Icon on your desktop and after providing the root password select the services icon under the System Menu.

Start the SSHD server by clicking the start button and set it to start on boot. A black check mark will appear and the service's status will change to running. Changes have been made to your boot up scripts located at /etc/rc.d/rc1.d though 5. Now when you boot the machine the sshd server will start.
Typing netstat –atp or lsof –i will know show the sshd listing on the port you selected.
lsof -i

```
COMMAND  PID USER   FD   TYPE DEVICE SIZE NODE NAME
kdm     1156 root    4u  IPv4  13385       UDP *:xdmcp
kdm     1156 root    5u  IPv4  13386       TCP *:blackjack (LISTEN)
X       1165 root    1u  IPv4  13390       TCP *:6000 (LISTEN)
kdm     1171 root    5u  IPv4  13386       TCP *:blackjack (LISTEN)
kdeinit 1257 root    6u  IPv4  13559       TCP *:1026 (LISTEN)
sshd    1909 root    3u  IPv4  13566       TCP *:ssh (LISTEN)
```

Postfix Configuration

Postfix is an alternate mail transfer agent software package for the popular but difficult to administer Sendmail package. Postfix is Sendmail compatible making its integration with other software painless in most cases while hopefully being more secure (most of the processes are chrooted. This is the system equivalent of running the server is a permissions jail). Mandrake stopped providing the Sendmail rpm with the version 8

downloadable distributions, instead providing the Sendmail package on their web site and Professional (4CD) distribution. Since we are not going to accept mail via the Internet though Postfix, our configuration will be quite simple. Postfix's functionality will only be required to send mail to the local host accounts from services such as syslog. Internet mail will be handled by another software package and will not be discussed in this paper.

main.cf

Postfixes /etc/postfix/main.cf file has over one hundred configuration settings that can be changed/applied. The nice thing about Postfix is that the default settings are pretty good for what we are using it for and will require little manipulation if any. Since the firewall isn't allowing the smtp port to be available to the Internet we don't have to worry about it being a target for exploitation.
But lets say your firewall script had a bug in it and didn't inialize when you booted up. Here are some things you can do to make it a little harder for the blackhats to slow them down.

Change the line:
smtpd_banner = $myhostname ESMTP $mail_name ($mail_version)

To something less revealing like

smtpd_banner = who.what.com ESMTP -Is understood here and logged -

So when someone telnets to your systems mail port instead of getting:

telnet 192.168.0.10 25
220 flea.home.com ESMTP Postfix (Release-20010228) (Linux-Mandrake)

They get this

telnet 192.168.0.10 25
220 who.ru.bum ESMTP –Is understood here and logged-

Starting Postfix

Click on your Mandrake Control Center Icon on your desktop and after providing the root password select the services icon under the System Menu.

Start the Postfix server by clicking the start button and set it to start on boot. A black check mark will appear and the service's status will change to running. Changes have been made to your boot up scripts located at /etc/rc.d/rc1.d though 5 Now when you boot the machine the Postfix server will start.
Typing netstat –atp or lsof –i will know show the sshd listing on the port you selected.
lsof -i

```
COMMAND  PID USER   FD   TYPE DEVICE SIZE NODE NAME
sshd     895 root    3u  IPv4   1006       TCP *:ssh (LISTEN)
master  1275 root    9u  IPv4   3626       TCP *:smtp (LISTEN)
kdm     1687 root    4u  IPv4   4119       UDP *:xdmcp
kdm     1687 root    5u  IPv4   4120       TCP *:1029 (LISTEN)
X       1694 root    1u  IPv4   4124       TCP *:6000 (LISTEN)
kdm     1698 root    5u  IPv4   4120       TCP *:1029 (LISTEN)
kdeinit 1784 root    6u  IPv4   4269       TCP *:iad1 (LISTEN)
```

For more information on Postfix, how-to configure, security information and example setups follow this link.
www.postfix.org

Risk Assessment

Now you're at the point where testing you system and firewall can take place. Fortunately there are sites on the Internet where you can test your firewall setup and get basic idea of what you are advertising to the world.

Port Scanning Sites
Sygate Security Scan
http://scan.sygatetech.com
A free service from Sygate Inc, the purpose of the scans is to show Internet users how vulnerable their systems are. Run the scans and you should get a list of blocked and open ports. The only port that should be open is the ssh port (22).

Gibsons Research Corp. Shields UP!
http://grc.com/default.htm
Another free service that will scan your machine for open ports and provide you with the results.

All Nettools
http://www.all-nettools.com/
Not really a port scanning site, All Nettools provides you a set of common tools like ping, traceroute, and others to check the configuration of your firewall and logging system. Run the command
tail –f /var/log/messages from the root prompt while you are scanning your system and watch the entries. If your firewall is working properly you should not be able to ping of traceroute your system here.

Additional Checks and tools to try

Nmap
http://www.insecure.org/nmap/index.html

Nmap is a must have port scanner that you will end up using again and again. Run against your localhost it will give you a good idea of what ports are open on your system. However, remember that your firewall is not filtering localhost on your machine. In other words what Nmap sees is not necessarily what the Internet is seeing. Read the man pages and manual provided at /usr/share/doc/nmap-2.53 for command line instruction.
Example command from the root prompt
Nmap –sS –O –P0 127.0.0.1
Nmap will do a TCP syn scan without ping the host. Nmap will also try to guess the remote operating system.

Logcheck
Logcheck is a automated script run via cron to check for security violations and other unusual active in your logs. It mails finding to root by default on a daily bases (This is configurable). Also to can be configured to ignore certain events and to report specific events you are looking for. Over a great log checking utility right out of the package.

Portsentry
Portsentry is a port scan detection and active response tool. Once a scan is detected Portsentry will apply a tcpwrapper, firewall rule to block the connection. Portsentry is part of the **Abacus** Project suite of security tool available at http://www.psionic.com/ and as RPM with your Mandrake distribution software.

Tripwire
The Mandrake distribution comes with an up to date RPM of Tripwire, a file integrity tool. Tripwire works by creating a database (a snap shot of your critical files) and running regular check against it and alerting you to when files have changed by email or log file. For more information please read the manual located at /usr/share/doc or the Tripwire web site http://www.tripwiresecurity.com/.

Project Summary

Your operating system is now as safe as the current status of the packages you have installed as well as the services(s) you have left open to the world. With some common security practices you can keep your system secure from unwanted remote guests.

Additional things to check and learn

1. Check your logs often! Logs are located in /var/log.
2. Subscribe to the Mandrake Security list server. Security updates and patch are announced to the public here as well as the web site. It's not a bad idea to subscribe to other security services like bug track, CERT or the SANS newsletters to name a few.
3. Make backups and set up a regular backup system. In many cases if something goes really wrong (harddrive failure, deletion of critical data) a

backup may be the only way to recover the data. Your Reference manual and the install how-to page will get you started in developing a backup solution you can stick with.

4. Develop an information library. Usenet groups. Web links, mail lists and conventional sources (books and periodicals) will help keep you current with the security as well as the blackhat trends.
5. Conduct security tests/audits on a regular bases using some of the tools mentioned in this paper and others you will find. Many new exploits, Trojans and other hacks come out every month. Remember the time it takes for Mandrake to develop patches verses the time the blackhat community takes to spread the word on a exploit may mean a compromised system for you.
6. Learn your system. Experiment with new commands and examine the file systems. It will help you later and knowledge is power.

Bibliography

James Stanger, Patrick T. Lane (2001). Hack Proofing Linux. Syngress Publishing Inc. Rockland, MA 02370

Brian Hatch, James Lee, George Kurtz. (2001) Hacking Linux Exposed: Linux Security Secrets & Solutions. Osborne/McGraw-Hill, Berkeley Ca 94710

The SANS Institute (2000) Securing Linux Step-by-Step Version 1.0. The SANS Institute Copyright 1999, 2000

Kirk Waingrow (1999). Unix Hints & Hacks. QUE Indianapolis, IN 46290

Dee-Ann LeBlanc (2000). Linux System Administration Black Book. Coriolis Group LLC Scottdale AZ 85260

Anonymous (2000) Maximum Linux Security. SAMS Indianapolis, IN 46290

MandrakeSoft (2000) Installation and User Guide. MandrakeSoft , http://www.linux-mandrake.com.
MandrakeSoft (2000) Reference Manual, MandrakeSoft, http://www.linux-mandrake.com.

Appendix A VI Editor

This reference is provided as a quick start to using vi and has been copied in its entirety from the web page http://unixhelp.ed.ac.uk/vi/ref.html.

vi Reference

Warning: some vi versions don't support the more esoteric features described in this document. You can edit/redistribute this document, as long as you don't make false claims on original authorship.

Author: Maarten Litmaath <maart@nikhef.nl>
Version: 8

vi Reference Table of Contents

---

contributions

Rich Salz <rsalz@bbn.com>
Eamonn McManus <emcmanus@cs.tcd.ie>
Diomidis Spinellis <diomidis%ecrcvax.uucp@pyramid.pyramid.com>
Blair P. Houghton <bph@buengc.bu.edu>
Rusty Haddock <{uunet,att,rutgers}!mimsy.umd.edu!fe2o3!rusty>
Panos Tsirigotis <panos@boulder.colorado.edu>
David J. MacKenzie <djm@wam.umd.edu>
Kevin Carothers <kevin@ttidca.tti.com>
Dan Mercer <mercer@ncrcce.StPaul.NCR.COM>
Ze'ev Shtadler <steed@il4cad.intel.com>

Paul Quare <pq@r2.cs.man.ac.uk>
Dave Beyerl <att!ihlpl!db21>
David C Johnson <johnson@wrs.com>
Lee Sailer <UH2@psuvm.psu.edu>
David Gast <gast@cs.ucla.edu>

legenda

```
default values          : 1
<*>                     : `*' must not be taken literally
[*]                     : `*' is optional
^X                      : <ctrl>X
<sp>                    : space
<cr>                    : carriage return
<lf>                    : linefeed
<ht>                    : horizontal tab
<esc>                   : escape
<erase>                 : your erase character
<kill>                  : your kill character
<intr>                  : your interrupt character
<a-z>                   : an element in the range
N                       : number (`*' = allowed, `-' = not appropriate)
CHAR                    : char unequal to <ht>|<sp>
WORD                    : word followed by <ht>|<sp>|<lf>
```

move commands

N	Command	Meaning
*	h | ^H | <erase>	<*> chars to the left.
*	j | <lf> | ^N	<*> lines downward.
*	l | <sp>	<*> chars to the right.
*	k | ^P	<*> lines upward.
*	$	To the end of line <*> from the cursor.
-	^	To the first CHAR of the line.
*	_	To the first CHAR <*> - 1 lines lower.
*	-	To the first CHAR <*> lines higher.
*	+ | <cr>	To the first CHAR <*> lines lower.
-	0	To the first char of the line.
*	|	To column <*> (<ht>: only to the endpoint).
*	f<char>	<*> <char>s to the right (find).
*	t<char>	Till before <*> <char>s to the right.
*	F<char>	<*> <char>s to the left.
*	T<char>	Till after <*> <char>s to the left.
*	;	Repeat latest `f'|`t'|`F'|`T' <*> times.
*	,	Idem in opposite direction.
*	w	<*> words forward.
*	W	<*> WORDS forward.
*	b	<*> words backward.
*	B	<*> WORDS backward.
*	e	To the end of word <*> forward.
*	E	To the end of WORD <*> forward.
*	G	Go to line <*> (default EOF).
*	H	To line <*> from top of the screen (home).
*	L	To line <*> from bottom of the screen (last).
-	M	To the middle line of the screen.

```
 * | )                   | <*> sentences forward.
 * | (                   | <*> sentences backward.
 * | }                   | <*> paragraphs forward.
 * | {                   | <*> paragraphs backward.
 - | ]]                  | To the next section (default EOF).
 - | [[                  | To the previous section (default begin of
file).
 - | `<a-z>              | To the mark.
 - | '<a-z>              | To the first CHAR of the line with the mark.
 - | ``                  | To the cursor position before the latest
absolute
                         |   jump (of which are examples `/' and `G').
 - | ''                  | To the first CHAR of the line on which the
cursor
                         |   was placed before the latest absolute jump.
 - | /<string>           | To the next occurrence of <string>.
 - | ?<string>           | To the previous occurrence of <string>.
 - | n                   | Repeat latest `/'|`?' (next).
 - | N                   | Idem in opposite direction.
 - | %                   | Find the next bracket and go to its match
                         |   (also with `{'|`}' and `['|`]').
```

searching (see above)

```
:ta <name>               | Search in the tags file[s] where <name> is
                         |   defined (file, line), and go to it.
^]                       | Use the name under the cursor in a `:ta'
command.
^T                       | Pop the previous tag off the tagstack and
return
                         |   to its position.
:[x,y]g/<string>/<cmd>   | Search globally [from line x to y] for
<string>
                         |   and execute the `ex' <cmd> on each
occurrence.
:[x,y]v/<string>/<cmd>   | Execute <cmd> on the lines that don't match.
```

undoing changes

```
u                        | Undo the latest change.
U                        | Undo all changes on a line, while not having
                         |   moved off it (unfortunately).
:q!                      | Quit vi without writing.
:e!                      | Re-edit a messed-up file.
```

appending text (end with <esc>)

```
 * | a                   | <*> times after the cursor.
 * | A                   | <*> times at the end of line.
 * | i                   | <*> times before the cursor (insert).
 * | I                   | <*> times before the first CHAR of the line
 * | o                   | On a new line below the current (open).
                         |   The count is only useful on a slow
terminal.
 * | O                   | On a new line above the current.
                         |   The count is only useful on a slow
terminal.
```

```
 * | ><move>            | Shift the lines described by <*><move> one
                       |   shiftwidth to the right.
 * | >>                 | Shift <*> lines one shiftwidth to the right.
 * | ["<a-zA-Z1-9>]p    | Put the contents of the (default undo) buffer
                       |   <*> times after the cursor.
                       |   A buffer containing lines is put only once,
                       |   below the current line.
 * | ["<a-zA-Z1-9>]P    | Put the contents of the (default undo) buffer
                       |   <*> times before the cursor.
                       |   A buffer containing lines is put only once,
                       |   above the current line.
 * | .                  | Repeat previous command <*> times.  If the
last
                       |   command before a `.' command references a
                       |   numbered buffer, the buffer number is
                       |   incremented first (and the count is
ignored):
                       |
                       |   "1pu.u.u.u.u      - `walk through' buffers
1
                       |                       through 5
                       |   "1P....           - restore them
```

deleting text

```
Everything deleted can be stored into a buffer. This is achieved by
putting a `"' and a letter <a-z> before the delete command. The
deleted text will be in the buffer with the used letter. If <A-Z>
is used as buffer name, the conjugate buffer <a-z> will be augmented
instead of overwritten with the text. The undo buffer always
contains the latest change. Buffers <1-9> contain the latest 9
LINE deletions (`"1' is most recent).

 * | x                  | Delete <*> chars under and after the cursor.
 * | X                  | <*> chars before the cursor.
 * | d<move>            | From begin to endpoint of <*><move>.
 * | dd                 | <*> lines.
 - | D                  | The rest of the line.
 * | <<move>            | Shift the lines described by <*><move> one
                       |   shiftwidth to the left.
 * | <<                 | Shift <*> lines one shiftwidth to the left.
 * | .                  | Repeat latest command <*> times.
```

changing text (end with <esc>)

```
 * | r<char>            | Replace <*> chars by <char> - no <esc>.
 * | R                  | Overwrite the rest of the line,
                       |   appending change <*> - 1 times.
 * | s                  | Substitute <*> chars.
 * | S                  | <*> lines.
 * | c<move>            | Change from begin to endpoint of <*><move>.
 * | cc                 | <*> lines.
 * | C                  | The rest of the line and <*> - 1 next lines.
 * | =<move>            | If the option `lisp' is set, this command
                       |   will realign the lines described by
<*><move>
```

```
                          |   as though they had been typed with the
option
                          |   `ai' set too.
 - | ~                    | Switch lower and upper cases
                          |   (should be an operator, like `c').
 * | J                    | Join <*> lines (default 2).
 * | .                    | Repeat latest command <*> times (`J' only
once).
 - | &                    | Repeat latest `ex' substitute command, e.g.
                          |   `:s/wrong/good'.
 - | :[x,y]s/<p>/<r>/<f>| Substitute (on lines x through y) the pattern
<p>
                          |   (default the last pattern) with <r>.
Useful
                          |   flags <f> are `g' for `global' (i.e. change
                          |   every non-overlapping occurrence of <p>)
and
                          |   `c' for `confirm' (type `y' to confirm a
                          |   particular substitution, else <cr>).
Instead
                          |   of `/' any punctuation CHAR unequal to <lf>
                          |   can be used as delimiter.
```

substitute replacement patterns

```
The basic meta-characters for the replacement pattern are `&' and `~';
these are given as `\&' and `\~' when nomagic is set.  Each instance
of `&' is replaced by the characters which the regular expression
matched.  The meta-character `~' stands, in the replacement
pattern, for the defining text of the previous replacement
pattern.  Other meta-sequences possible in the replacement pattern
are always introduced by the escaping character `\'.  The sequence
`\n' (with `n' in [1-9]) is replaced by the text matched by the
n-th regular subexpression enclosed between `\(' and `\)'.  The
sequences `\u' and `\l' cause the immediately following character
in the replacement to be converted to upper- or lower-case
respectively if this character is a letter.  The sequences `\U' and
`\L' turn such conversion on, either until `\E' or `\e' is
encountered, or until the end of the replacement pattern.
```

remembering text (yanking)

```
With yank commands you can put `"<a-zA-Z>' before the command, just as
with delete commands.  Otherwise you only copy to the undo buffer.
The use of buffers <a-z> is THE way of copying text to another file;
see the `:e <file>' command.

* | y<move>            | Yank from begin to endpoint of <*><move>.
 * | yy                | <*> lines.
 * | Y                 | Idem (should be equivalent to `y$' though).
 - | m<a-z>            | Mark the cursor position with a letter.
```

commands while in append|change mode

```
^@                        | If typed as the first character of the
                          |   insertion, it is replaced with the previous
                          |   text inserted (max. 128 chars), after which
```

```
                        |   the insertion is terminated.
^V                      | Deprive the next char of its special meaning
                        |   (e.g. <esc>).
^D                      | One shiftwidth to the left, but only if
                        |   nothing else has been typed on the line.
0^D                     | Remove all indentation on the current line
                        |   (there must be no other chars on the line).
^^D                     | Idem, but it is restored on the next line.
^T                      | One shiftwidth to the right, but only if
                        |   nothing else has been typed on the line.
^H | <erase>            | One char back.
^W                      | One word back.
<kill>                  | Back to the begin of the change on the
                        |   current line.
<intr>                  | Like <esc> (but you get a beep as well).
```

writing, editing other files, and quitting vi

```
In `:' `ex' commands - if not the first CHAR on the line - `%' denotes
the current file, `#' is a synonym for the alternate file (which
normally is the previous file).  As first CHAR on the line `%' is a
shorthand for `1,$'.  Marks can be used for line numbers too: '<a-z>.
In the `:w'|`:f'|`:cd'|`:e'|`:n' commands shell meta-characters can be
used.

:q                      | Quit vi, unless the buffer has been changed.
:q!                     | Quit vi without writing.
^Z                      | Suspend vi.
:w                      | Write the file.
:w <name>               | Write to the file <name>.
:w >> <name>            | Append the buffer to the file <name>.
:w! <name>              | Overwrite the file <name>.
:x,y w <name>           | Write lines x through y to the file <name>.
:wq                     | Write the file and quit vi; some versions
quit
                        |   even if the write was unsuccessful!
                        |   Use `ZZ' instead.
ZZ                      | Write if the buffer has been changed, and
                        |   quit vi.  If you have invoked vi with the
`-r'
                        |   option, you'd better write the file
                        |   explicitly (`w' or `w!'), or quit the
                        |   editor explicitly (`q!') if you don't want
                        |   to overwrite the file - some versions of vi
                        |   don't handle the `recover' option very
well.
:x [<file>]             | Idem [but write to <file>].
:x! [<file>]            | `:w![<file>]' and `:q'.
:pre                    | Preserve the file - the buffer is saved as if
                        |   the system had just crashed; for
emergencies,
                        |   when a `:w' command has failed and you
don't
                        |   know how to save your work (see vi -r).
:f <name>               | Set the current filename to <name>.
:cd [<dir>]             | Set the working directory to <dir>
                        |   (default home directory).
```

```
:cd! [<dir>]              | Idem, but don't save changes.
:e [+<cmd>] <file>        | Edit another file without quitting vi - the
                          |   buffers are not changed (except the undo
                          |   buffer), so text can be copied from one
file to
                          |   another this way.  [Execute the `ex'
command
                          |   <cmd> (default `$') when the new file has
been
                          |   read into the buffer.]  <cmd> must contain
no
                          |   <sp> or <ht>.  See vi startup.
:e! [+<cmd>] <file>       | Idem, without writing the current buffer.
^^                        | Edit the alternate (normally the previous)
file.
:rew                      | Rewind the argument list, edit the first
file.
:rew!                     | Idem, without writing the current buffer.
:n [+<cmd>] [<files>]     | Edit next file or specify a new argument
list.
:n! [+<cmd>] [<files>]    | Idem, without writing the current buffer.
:args                     | Give the argument list, with the current file
                          |   between `[' and `]'.
```

display commands

```
^G                        | Give file name, status, current line number
                          |   and relative position.
^L                        | Refresh the screen (sometimes `^P' or `^R').
^R                        | Sometimes vi replaces a deleted line by a
`@',
                          |   to be deleted by `^R' (see option redraw).
[*]^E                     | Expose <*> more lines at bottom, cursor
                          |   stays put (if possible).
[*]^Y                     | Expose <*> more lines at top, cursor
                          |   stays put (if possible).
[*]^D                     | Scroll <*> lines downward
                          |   (default the number of the previous scroll;
                          |   initialization: half a page).
[*]^U                     | Scroll <*> lines upward
                          |   (default the number of the previous scroll;
                          |   initialization: half a page).
[*]^F                     | <*> pages forward.
[*]^B                     | <*> pages backward (in older versions `^B'
only
                          |   works without count).
```

If in the next commands the field <wi> is present, the windowsize will change to <wi>. The window will always be displayed at the bottom of the screen.

```
[*]z[wi]<cr>              | Put line <*> at the top of the window
                          |   (default the current line).
[*]z[wi]+                 | Put line <*> at the top of the window
                          |   (default the first line of the next page).
[*]z[wi]-                 | Put line <*> at the bottom of the window
                          |   (default the current line).
```

```
[*]z[wi]^               | Put line <*> at the bottom of the window
                        |   (default the last line of the previous
page).
[*]z[wi].               | Put line <*> in the centre of the window
                        |   (default the current line).
```

mapping and abbreviation

```
When mapping take a look at the options `to' and `remap' (below).

:map <string> <seq>     | <string> is interpreted as <seq>, e.g.
                        |   `:map ^C :!cc %^V<cr>' to invoke `cc' (the
C
                        |   compiler) from within the editor
                        |   (vi replaces `%' with the current file
name).
:map                    | Show all mappings.
:unmap <string>         | Deprive <string> of its mapping.  When vi
                        |   complains about non-mapped macros (whereas
no
                        |   typos have been made), first do something
like
                        |   `:map <string> Z', followed by
                        |   `:unmap <string>' (`Z' must not be a macro
                        |   itself), or switch to `ex' mode first with
`Q'.
:map! <string> <seq>    | Mapping in append mode, e.g.
                        |   `:map! \be begin^V<cr>end;^V<esc>O<ht>'.
                        |   When in append mode <string> is preceded by
                        |   `^V', no mapping is done.
:map!                   | Show all append mode mappings.
:unmap! <string>        | Deprive <string> of its mapping (see
`:unmap').
:ab <string> <seq>      | Whenever in append mode <string> is preceded
and
                        |   followed by a breakpoint (e.g. <sp> or
`,'), it
                        |   is interpreted as <seq>, e.g.
                        |   `:ab ^P procedure'.  A `^V' immediately
                        |   following <string> inhibits expansion.
:ab                     | Show all abbreviations.
:unab <string>          | Do not consider <string> an abbreviation
                        |   anymore (see `:unmap').
@<a-z>                  | Consider the contents of the named register a
                        |   command, e.g.:
                        |       o0^D:s/wrong/good/<esc>"zdd
                        |   Explanation:
                        |       o             - open a new line
                        |       0^D           - remove indentation
                        |       :s/wrong/good/ - this input text is an
                        |                        `ex' substitute
command
                        |       <esc>         - finish the input
                        |       "zdd          - delete the line just
                        |                       created into register
`z'
                        |   Now you can type `@z' to replace `wrong'
```

```
                          |   with `good' on the current line.
@@                        | Repeat last register command.
```

switch and shell commands

```
Q | ^\ | <intr><intr>    | Switch from vi to `ex'.
:                         | An `ex' command can be given.
:vi                       | Switch from `ex' to vi.
:sh                       | Execute a subshell, back to vi by `^D'.
:[x,y]!<cmd>              | Execute a shell <cmd> [on lines x through y;
                          |   these lines will serve as input for <cmd>
and
                          |   will be replaced by its standard output].
:[x,y]!! [<args>]         | Repeat last shell command [and append
<args>].
:[x,y]!<cmd> ! [<args>]   | Use the previous command (the second `!') in
a
                          |   new command.
[*]!<move><cmd>           | The shell executes <cmd>, with as standard
                          |   input the lines described by <*><move>,
                          |   next the standard output replaces those
lines
                          |   (think of `cb', `sort', `nroff', etc.).
[*]!<move>!<args>         | Append <args> to the last <cmd> and execute
it,
                          |   using the lines described by the current
                          |   <*><move>.
[*]!!<cmd>                | Give <*> lines as standard input to the
                          |   shell <cmd>, next let the standard output
                          |   replace those lines.
[*]!!! [<args>]           | Use the previous <cmd> [and append <args> to
it].
:x,y w !<cmd>             | Let lines x to y be standard input for <cmd>
                          |   (notice the <sp> between the `w' and the
`!').
:r!<cmd>                  | Put the output of <cmd> onto a new line.
:r <name>                 | Read the file <name> into the buffer.
```

vi startup

```
vi [<files>]              | Edit the files, start with the first page of
                          |   the first file.
```

The editor can be initialized by the shell variable `EXINIT', which looks like:

```
        EXINIT='<cmd>|<cmd>|...'
        <cmd>: set options
               map ...
               ab ...
        export EXINIT (in the Bourne shell)
```

However, the list of initializations can also be put into a file. If this file is located in your home directory, and is named `.exrc' AND the variable `EXINIT' is NOT set, the list will be executed automatically at startup time. However, vi will always execute the contents of a `.exrc' in the current directory, if you own the file.

```
Else you have to give the execute (`source') command yourself:

        :so file

In a `.exrc' file a comment is introduced with a double quote
character:
the rest of the line is ignored.  Exception: if the last command on the
line is a `map[!]' or `ab' command or a shell escape, a trailing
comment
is not recognized, but considered part of the command.

On-line initializations can be given with `vi +<cmd> file', e.g.:

vi +x file              | The cursor will immediately jump to line x
                        |   (default last line).
vi +/<string> file      | Jump to the first occurrence of <string>.

You can start at a particular tag with:

vi -t <tag>             | Start in the right file in the right place.

Sometimes (e.g. if the system crashed while you were editing) it is
possible to recover files lost in the editor by `vi -r file'.  A plain
`vi -r' command shows the files you can recover.
If you just want to view a file by using vi, and you want to avoid any
change, instead of vi you can use the `view' or `vi -R' command:
the option `readonly' will be set automatically (with `:w!' you can
override this option).
```

the most important options

```
ai                      | autoindent - In append mode after a <cr> the
                        |   cursor will move directly below the first
                        |   CHAR on the previous line.  However, if the
                        |   option `lisp' is set, the cursor will align
                        |   at the first argument to the last open
list.
aw                      | autowrite - Write at every shell escape
                        |   (useful when compiling from within vi).
dir=<string>            | directory - The directory for vi to make
                        |   temporary files (default `/tmp').
eb                      | errorbells - Beeps when you goof
                        |   (not on every terminal).
ic                      | ignorecase - No distinction between upper and
                        |   lower cases when searching.
lisp                    | Redefine the following commands:
                        |   `(', `)'   - move backward (forward) over
                        |                S-expressions
                        |   `{', `}'   - idem, but don't stop at atoms
                        |   `[[', `]]' - go to previous (next) line
                        |                beginning with a `('
                        |   See option `ai'.
list                    | <lf> is shown as `$', <ht> as `^I'.
magic                   | If this option is set (default), the chars
`.',
                        |   `[' and `*' have special meanings within
search
```

```
                        |   and `ex' substitute commands.  To deprive
such
                        |   a char of its special function it must be
                        |   preceded by a `\'.  If the option is turned
off
                        |   it's just the other way around.  Meta-
chars:
                        |   ^<string>    - <string> must begin the line
                        |   <string>$    - <string> must end the line
                        |   .            - matches any char
                        |   [a-z]        - matches any char in the
range
                        |   [^a-z]       - any char not in the range
                        |   [<string>]   - matches any char in <string>
                        |   [^<string>]  - any char not in <string>
                        |   <char>*      - 0 or more <char>s
                        |   \<<string>   - <string> must begin a word
                        |   <string>\>   - <string> must end a word
modeline                | When you read an existing file into the
buffer,
                        |   and this option is set, the first and last
5
                        |   lines are checked for editing commands in
the
                        |   following form:
                        |
                        |     <sp>vi:set options|map ...|ab ...|!...:
                        |
                        |   Instead of <sp> a <ht> can be used, instead
of
                        |   `vi' there can be `ex'.  Warning: this
option
                        |   could have nasty results if you edit a file
                        |   containing `strange' modelines.
nu                      | number - Numbers before the lines.
para=<string>           | paragraphs - Every pair of chars in <string>
is
                        |   considered a paragraph delimiter nroff
macro
                        |   (for `{' and `}').  A <sp> preceded by a
`\'
                        |   indicates the previous char is a single
letter
                        |   macro.  `:set para=P\ bp' introduces `.P'
and
                        |   `.bp' as paragraph delimiters.  Empty lines
and
                        |   section boundaries are paragraph boundaries
                        |   too.
redraw                  | The screen remains up to date.
remap                   | If on (default), macros are repeatedly
                        |   expanded until they are unchanged.
                        |   Example: if `o' is mapped to `A', and `A'
                        |   is mapped to `I', then `o' will map to `I'
                        |   if `remap' is set, else it will map to `A'.
report=<*>              | Vi reports whenever e.g. a delete
                        |   or yank command affects <*> or more lines.
```

```
ro                      | readonly - The file is not to be changed.
                        |   However, `:w!' will override this option.
sect=<string>           | sections - Gives the section delimiters (for
`[['
                        |   and `]]'); see option `para'. A `{'
beginning a
                        |   line also starts a section (as in C
functions).
sh=<string>             | shell - The program to be used for shell
escapes
                        |   (default `$SHELL' (default `/bin/sh')).
sw=<*>                  | shiftwidth - Gives the shiftwidth (default 8
                        |   positions).
sm                      | showmatch - Whenever you append a `)', vi
shows
                        |   its match if it's on the same page; also
with
                        |   `{' and `}'.  If there's no match at all,
vi
                        |   will beep.
taglength=<*>           | The number of significant characters in tags
                        |   (0 = unlimited).
tags=<string>           | The space-separated list of tags files.
terse                   | Short error messages.
to                      | timeout - If this option is set, append mode
                        |   mappings will be interpreted only if
they're
                        |   typed fast enough.
ts=<*>                  | tabstop - The length of a <ht>; warning: this
is
                        |   only IN the editor, outside of it <ht>s
have
                        |   their normal length (default 8 positions).
wa                      | writeany - No checks when writing
(dangerous).
warn                    | Warn you when you try to quit without
writing.
wi=<*>                  | window - The default number of lines vi
shows.
wm=<*>                  | wrapmargin - In append mode vi automatically
                        |   puts a <lf> whenever there is a <sp> or
<ht>
                        |   within <wm> columns from the right margin
                        |   (0 = don't put a <lf> in the file, yet put
it
                        |   on the screen).
ws                      | wrapscan - When searching, the end is
                        |   considered `stuck' to the begin of the
file.

:set <option>           | Turn <option> on.
:set no<option>         | Turn <option> off.
:set <option>=<value>   | Set <option> to <value>.
:set                    | Show all non-default options and their
values.
:set <option>?          | Show <option>'s value.
:set all                | Show all options and their values.
```

Converted to html by Jeff Aspinall (aspinall@umich.edu) with help from Meng Weng Wong's txt2html script, 22 June 1994

 *This site maintained by unixhelp@ed.ac.uk*

Last Updated: October 18th, 2017

Upcoming Training

SANS 2018	**Orlando, FL**	**Apr 03, 2018 - Apr 10, 2018**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**