Global Information Assurance Certification Paper

Implementing the Debian Linux Distribution as a Firewall for SOHO Networks

Submitted by: Renato Sabolboro
SANS GIAC Administrivia Version 2.2
Practical Assignment version 1.4b

Table of Contents

1. Abstract

2. Reasons for Choosing Debian GNU/Linux

    A. Definition of terms
    B. Important considerations in choosing a firewall solution
        a. Economical
        b. Security
        c. Support
        d. Availability
        e. Freedom
        f. Simplicity
        g. Cost of Ownership

3. The Debian Linux Distribution
    A. Where and how to get the software components
    B. Hardware requirements of a SOHO Firewall/Gateway

4. The process of assembling and installing the system
    A. Downloading the software components
    B. Assembling the hardware
    C. Installing the software
    D. Configuring/Securing the system

5. Conclusion

6. References

1. Abstract

Businesses today require Internet access even for a small office to be productive, be it for e-mail, web surfing, file transfers or perform transaction processing. The size of these offices could range from a large multinational with a remote office of a few dozen personnel to a niche office that just requires a secure mode of transferring information through the Internet. These could be law firms that need to transmit contracts securely to their clients, doctor's clinics that must provide regulatory information or communicate securely with a hospital, a business consultant that provide competitive advantage analysis, an accountant with financial information, and many others. The setup discussed here could even be implemented by anyone who would generally want to learn how to make his or her home network more secure against intrusion. The Internet is comparable to an electronic jungle and having a firewall provides a layer of protection in addition to the anti-virus software against viruses, Trojans, and other malicious software that tries to infiltrate its defenses.

Always on broadband is now the norm so that remote networks or computers if not adequately secured are faced with a risk that must be mitigated. Consequently, to increase the security of electronic data communication, these transactions that are conducted over the public Internet must be done in a way that there is a very low probability of being compromised from any end of the communication channel. This paper will develop a method of securing one end of the communication transaction using Debian GNU/Linux 3.0 [1] running on a laptop with the assumption that the other end is already a secure system so that two-way communication can be established. If a desktop is preferred, it will only require minimal changes to the steps outlined herein.

2. Reasons for Choosing Debian GNU/Linux

A. Definition of Terms

Before we begin to outline the reasons for choosing the Debian GNU/Linux distribution as the firewall of choice, first we must define some terms that are important concepts in computer firewalls and networks. When I started in computers and networking, many of these terms and concepts stumped me and either I had to research them or if not finding any understandable document pester others to explain to me these terms that I encounter a lot. Hopefully, you will understand the terms and concepts that I have distilled here.

Firewall – The first firewalls were on trains. Coal-powered trains had a large furnace in the engine room, along with a pile of coal. The engineer would shovel coal into the engine. This process created coal dust, which was highly flammable. Occasionally the coal dust would catch fire, causing an engine fire that would

sometimes spread into passenger cars. Since dead passengers reduced revenue, train engines were built with iron walls right behind the engine compartment. [2]

As applied to networks, it is a device to manage the flow of electronic data packets to prevent or allow access of a computer network or part of a network from unauthorized or authorized users respectively. This short description summarizes what a computer network firewall does:

> *… This means essentially that the firewall is the first program or process that receives and handles incoming network traffic, and it is the last to handle outgoing traffic.* [3]

A firewall has three major components, namely:

- Hardware – the physical device that contains networking, storage, and computing capabilities to perform the function. Usually, this is a computer with fast network interfaces and enough CPU processing power to perform its operations and adequate disk space to store software and policies.
- Software – the logic that controls the hardware. The software components include both the operating system (OS) tuned to the task of being a firewall and the application software that performs the actual logic operations. In a PC that operates as a firewall, the OS could be GNU/Linux 3.0 plus netfilter/iptables or ipchains that performs the actual checking of packets. [4] netfilter/iptables is touted to be more superior to ipchains because it can do statefull packet inspection. [2] This is analogous to ipchains checking each word for proper spelling as against netfilter/iptables checking the whole sentence for both grammar and spelling to see if the other party is trying to communicate properly or do something illegal or unusual and take appropriate action based on existing policy rules. While ipchains can check each packet as it goes through the firewall, netfilter/iptables analyzes the whole conversation.
- Rules or Policy – used by the software as reference to grant or prevent movement of packets through the firewall. The reference logic used by the application software to either allow or deny a packet trying to go through the firewall would be determined by business or user requirements. Thus, for a business, some activities that are used primarily for Internet gaming, chat, and instant messaging or other personal uses might be blocked but might be allowed in a home setup.

Packet – as applied to networks, is a discrete amount of data that is transmitted over a network as defined by the protocols implemented in such a network. A network packet contains at least a source address, destination address, sequence number, data and checksum, to enable the receiving end to understand and reconstruct the whole message. A packet is usually part of several that will form a complete message and it is the responsibility of the receiving end to reconstruct the whole

[2] Scheneier, Bruce. Secrets and Lies: Digital Security in a Networked World. New York, NY: John Wiley & Sons, Inc., (2000). Page 188 & 191.

message based on sequence number information because each packet can travel different routes such that the seventh packet in a message containing ten, can arrive ahead of the sixth.

Gateway – In TCP/IP networks a computer that has two or more network interfaces is called a multi-homed host. Each of these interfaces is connected to a different network and the computer routes traffic between these network segments. This was the old definition of a gateway, routing packets between the networks that are connected to it. Today, the term router is used to describe this routing function, whereas the term gateway is reserved for those functions that correspond to the upper layers of the OSI Reference Model. [5] Many interchange the terms gateway, router, and firewall because they are synonymous but I would define the firewall as a gateway that has special rules to allow or prevent data packets to go through it.

OSI Reference Model – This model has seven layers and was developed to enable efficient communications between computers. [5] Each layer will communicate with the layer above or below it if there is one and be able to understand the information prepared on the same layer of another computer.

TCP/IP Protocol – The TCP/IP protocol actually is a suite of protocols. The Defense Advance Research Projects Agency (DARPA) originally developed Transmission Control Protocol/Internet Protocol (TCP/IP) to interconnect various defense department computer networks. [6] The white paper, Understanding IP Addressing: Everything You Ever Wanted To Know, by Chuck Semeria, and published while he worked at 3Com has been one of my longtime references and contributed much to a very good understanding of the subject matter and would be recommended reading if you have minimal knowledge on the subject and will want to implement something similar to what is presented here. [7]

Ethernet – A protocol developed in the early 1970s by Robert Metcalfe and his colleagues at Xerox's PARC to enable computers to connect to and communicate with each other. This implements layer 1 and partly layer 2 of the ISO Reference Model. This web site by Charles Spurgeon, http://www.ethermanage.com/ethernet/ [8], though already a good reference, together with his book, Ethernet: The Definitive Guide [9] discusses everything you need to know about Ethernet from its beginnings transmitting at 2.94 Mbps to its current metamorphosis transmitting at 10 Gbps. TCP/IP is a protocol and Ethernet is too, what's the difference? A simple analogy here is that if you are corresponding with someone in another country by mail, the letter you wrote and dropped in the mail box, and initially sorted by your post office is TCP/IP part, while the trucking and shipping service that hauls the sorted mail is Ethernet, and the gateway is a mail sorting and distribution center. The firewall now is a sorting center with detection systems in place to detect bombs, drugs, hazardous chemicals, etc.

[5] Hare, Chris and Siyan, Karanjit. Internet Firewalls and Network Security. Indianapolis, IN: New Riders Publishing, (1995). Page 154 & 275.

Port –For both computers to listen to and understand each other's communications, they must agree on what ports to send and receive their data so that they can communicate properly. RFC 1700 [10] describes some well-known ports and what service (like telnet, ftp, smtp, etc.) should use what port number, this is analogous to a channel or frequency in radio communications. Typically, only a single port is used for both sending and receiving data but this is not always the case. For example in ftp, port 21 is used for control while port 20 is used for data so it's a little bit tricky to set up rules for a firewall that protects an ftp server because there are two ports involved.

B. Important considerations in choosing a firewall solution

The following items are what I would consider important considerations that must be addressed if one is contemplating on installing a firewall for a small office/home office (SOHO) network or even just a home network. The reasons outlined here will highlight the strengths of the Debian GNU/Linux 3.0 firewall solution.

a. Economical – Any business that operates for profit would want to maximize its profits by minimizing its cash outlay. Instead of individual PCs dialing to the net, installing a firewall/gateway can save you monthly fees by sharing a broadband connection or if there are several users that are consolidated plus the inherent advantages of a more secure and faster connection compared to dialup. One way to implement a SOHO firewall is by using a cheap but robust solution consisting of a small or low end PC for the hardware and free software like GNU/Linux 3.0 plus netfilter or ipchains since the license is royalty-free and the software is freely downloadable. It can easily perform this job and in fact my recommendation discussed in more detail below, is to use a low-end laptop because it is easy to procure and virtually noiseless if one only needs to be connected to the internet and internal network computers just require basic access like email, file transfers and web surfing. But even with a laptop with a 1GB hard disk, this setup can easily support a dozen or more computers for basic Internet needs. Later, when your network needs expand such as requiring VPN connectivity to other hosts or networks, you can evolve your setup by just installing additional applications that are freely downloadable.
b. Security – In order for the firewall to be economical it must also be able to perform its function securely and also not easily compromised. By doing this yourself, you will be pretty confident that there are no trojans, logic bombs, or any other software components like startup scripts and back door accounts that could compromise your internal network than if you have bought a commercial solution. Others could argue that Open Source Software (OSS) like Debian GNU/Linux could also be laden with bugs and trojans. By its very nature of being freely available, the OSS code can be easily reviewed and is continuously being done by many to make sure bugs are weeded out or there are no malicious code in it. There were already several reported instances of OSS software being compromised, but due to its nature of publicizing the code, it is only a matter of

time that the malicious piece of code is detected, removed, and the public advised of precautionary measures. Commercial implementations on the other hand may not immediately publish vulnerabilities as this could have a negative effect on their immediate bottom line say for example they are in the middle of negotiations with a large customer and their solution has been found to be insecure, they might not immediately announce it until after the customer signs up making their other existing customers vulnerable to the exploit. Although there is a tradeoff of time in setting up the system, over the long run, you will be confident that your internal network is secure from outsiders if you properly configured your policy and frequently patch your system through automatic cron jobs. Subsequent maintenance can be performed without paying others to do the job for you. Debian GNU/Linux as well as the other OSS software packages allows you to do this, as it is a freely available operating system while commercial solutions would not allow you to tweak their system as easily.

c. Support –If you encounter problems along the way, particularly if you are new to Linux, the available documentation from Debian's web site alone could overwhelm you and they are all free. Available and adequate support is at the same time free through the various users groups and developer groups and in my experience, the turnaround to resolving problem is even faster than those offered by for-profit entities. Debian has several dozen mailing lists that you can join for free. [11] Often, the solution to your problem will be in your mailbox less than 24 hours after you ask about it on the appropriate mailing list. In times that there is no one who can answer your question as it is not the proper forum, they will point you to the right direction.
d. Availability – While some commercial software are not made available by their publishers in some markets due to reasons like legal restrictions, economics, support infrastructure, etc., Debian GNU/Linux 3.0 and its associated packages are available freely to anyone anywhere without restriction on any person, group or fields of endeavor. Either you buy the media from entities that burn the software into CD-ROMs for a minimal fee or download the software, as what I did, from a mirror that is near you. Near in a sense that the download is fast because there are only a few hops to traverse. And then the Debian GNU/Linux distribution is free of any commercial interest unlike some of the other Linux distributions so there is no pressure to succeed commercially. It is this reason together with Freedom why I chose Debian GNU/Linux because there are no commercial entities that directly control it.
e. Freedom - The major advantage of Open Source Software published under the GNU General Public License [12] is that it is freely modifiable and your obligation when you make changes to the source code and distribute it is you must include the source code and the changes in source form. But if you don't intend to distribute your changes to the code you have no obligation at all to reveal your modifications. With commercial software you have no choice at all because the software is provided to you in compiled format and non-modifiable so it's a take it or leave it proposition. Also, you are not forced to upgrade compared to commercial solutions where the vendor will not anymore support the product you bought just over a year ago unless you upgrade holding you forever hostage to

their whims. Try reverse engineering commercial security software even for the purpose of making it suitable for your needs and their lawyers will be after you. With the enactment of the Digital Millenium Copyrights Act (DMCA), it is now illegal in the United States to reverse-engineer software with the intent of changing the security mechanisms in it. [13] With Debian GNU/Linux that distributes its software using the GNU General Public License you have total control of your software and you can do whatever you want with it within its license, and that is almost anything, except claiming it as your own.

f. Simplicity – While the Debian GNU/Linux 3.0 is available in seven CDs, the components of the software for a firewall solution can be implemented in a hard disk that is less than 1 GB in size. You will only need to install components needed to perform the task of a firewall so other services and daemons, together with their vulnerabilities can be eliminated. No web server, ftp, email, news, files, etc. and it makes the system more secure.
g. Cost of Ownership – Being free and with a vast support community, the only cost associated with using this solution is your time if you can quantify it. But it would still be cheaper because commercial software also has this cost because although the learning curve is steep, once you have mastered it, the subsequent time required to maintain it is much less. Debian GNU/Linux is know for its superior apt-get tool that automates many of the support activities like patching your system so I would strongly believe that it would not be more expensive to employ over the life of the system compared to commercial solutions.

3. The Debian GNU/Linux Distribution

A. Where and how to get the software

The Debian GNU/Linux 3.0 distribution main page has links on the left side entitled "Getting Debian" and clicking on it provides one with two main options, namely, buy a set of CDs or download it. I chose to download it to highlight one of its strengths but purchasing from official distributors help as well because part of their sales are donated to the developers of Debian. The current way of downloading Debian GNU/Linux 3.0 is by using jigdo[14], short for Jigsaw Download. The concept of jigdo is simple, it will download all the files necessary to create the CD-ROM .iso image ("filename".jigdo and "filename".template where filename is the name of the .iso image), assemble each image after downloading the necessary files, then you can burn the completed image into a CD-ROM. There are more Debian regular file mirrors than there are CD image mirrors due to the resources involved for the later so it would be usually faster to download individual files because these mirrors would be nearer network wise. Jigdo has compiled versions for MS Windows, Solaris, and Linux as well as source if you want to port to other platforms or paranoid not to use compiled versions. I am describing a download using jigdo on MS Windows on the assumption that the implementer following the steps here has not yet installed a Debian GNU/Linux 3.0 distribution previously plus the fact that PCs running MS Windows are ubiquitous. The .jigdo and .template files for Debian GNU/Linux 3.0 can be found at http://cdimage.debian.org/jigdo-area/current/jigdo/i386/. The

directory above it would enable one to choose images for different computer architectures. I would suggest creating separate directories for each .iso image as jigdo creates temporary files.

B. Hardware requirements of a SOHO Firewall/Gateway
The PC hardware requirements to implement a firewall for a SOHO are very minimal by current standards. A PC with an Intel Pentium running at 100 MHz with two 10 Mbps Ethernet network interface cards will not be the bottleneck in the link if one is using either a cable or DSL modem connection. Ethernet has shown to sustain 2.5Mbps [15] from its theoretical 10Mbps for a network with several dozen nodes. My cable Internet provider only claims a maximum of up to 50 times 28kbps (they throttle the speed) so that's just 1.4 Mbps (See Figure 1).

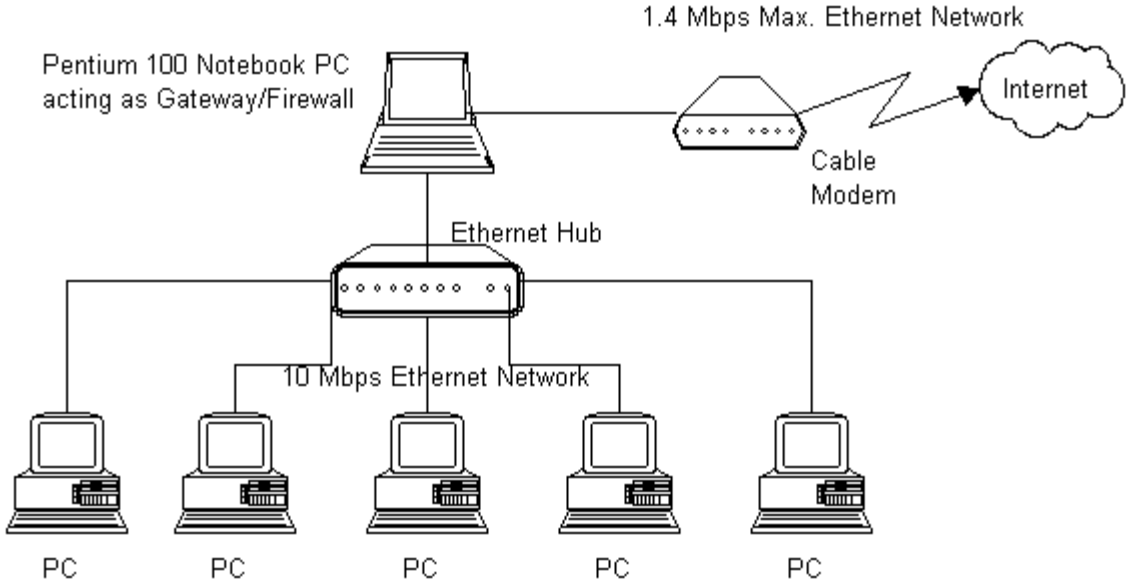

Figure 1. Ethernet Network for a SOHO

4. The process of assembling and installing the system

I sometimes cook by experimenting with new ingredients or procedures and the outcomes are mixed. For those times that the dish turns out good I always rue that I did not write down the procedures and materials needed. I would want to make this procedure similar to cooking so that it is repeatable by others and hopefully they get the same good results. As with cooking, there are some basic skills expected like being able to navigate within a bash shell, edit a file, navigate through directories, and some other similar tasks.

A. Downloading the software components

1. Go to Debian's main page [1] and click on the link on the left side entitled "Getting Debian." If it is your first time there, I would suggest reading the links

under "About" to understand the principles espoused by the developers of the software distribution. After clicking on the link, you will be presented with a new page (http://www.debian.org/distrib/) giving you a choice of either buying or downloading.

2. Click on the "make a CD set yourself" link somewhere near the middle of the page and you are again presented with a new page (http://www.debian.org/distrib/cd). Click on "Assemble images using jigdo:" link somewhere at the middle of the page. Read the whole page, it will help you understand the jigdo downloading process.
3. Right-click on the "jigdo" link on the first paragraph of the page (not on the tan menu bar at the top of the page) and open a new browser window. This is the home page for jigdo and is on a server outside of Debian's. Read through the whole page to understand jigdo in more detail then download the file from the link "jigdo-lite 0.6.8 for Windows" somewhere near the top of the page and save it to a folder that you can remember (I suggest to drive:\Debian where drive is C:, D:, etc.). The drive needs roughly 5GB of free space for the Debian .iso files if you download them all simultaneously from one or different mirrors before burning to a CD.
4. Create folders drive:\Debian\cd1 up to drive:\Debian\cd7 under drive:\Debian for the seven CDs that you will download. Actually there are eight files but CD 1 with the "NONUS" has extra files and is provided to get around US export restrictions and is downloaded from a server outside the US.
5. You will need the Winzip utility in this step to extract the compressed jigdo installer file. Go to the folder where you have downloaded the file and double-click on it. If you have Winzip installed or another software that does a similar job, extract the file seven times once each into drive:\Debian\cd1 up to drive:\Debian\cd7 and a new subfolder is created named jigdo-bin on each. On each of the seven folders where the extracted files are located, two files will be created, jigdo-lite.bat and README.txt. The structure will look as follows assuming a C: drive:

```
C:\Debian\CD1\jigdo-lite.bat
             \README.txt
             \<jigdo-bin>
C:\Debian\CD2\jigdo-lite.bat
             \README.txt
             \<jigdo-bin>
.
.
.
C:\Debian\CD7\jigdo-lite.bat
             \README.txt
             \<jigdo-bin>
```

6. Open, read, and understand any of the README.txt files using a text editor. This will explain how to connect to a mirror and download the files. This is the trickiest part of the download and must be read and understood carefully or you will not be able to download the files even if you are able to connect to

the mirrors because jigdo needs to know the root directory of the Debian mirror and this could be different for each.

7. From the browser window of step 3 above, click on the "Available images" heading, "USA mirror" (or your mirror of choice if outside the USA), then "i386" directory tree. Download each pair of the .jigdo and .template files into each of the seven directories, say C:\Debian\CD1 to C:\Debian\CD7 in our example above.
8. Open a DOS shell or using Windows Explorer, run the jigdo-lite.bat file described in step 5 above. When prompted for a jigdo file, enter the name of the file that you have downloaded on that directory. So for C:\Debian\CD1, it would be woody-i386-1.jigdo. On the next prompt, enter "us" if you are in the USA or you might need to use your country code like "ph" for Philippines or "de" for Germany. You will then be presented with a list of servers. Read through the list and choose a few that you believe is near to you geographically or network wise so your download would be fast and quick, and note them down on slip of paper as you will need them for the other CDs. Do not download from only one Debian mirror but instead try to distribute them from a few to distribute the load among mirrors. Try to navigate through the mirrors and check the location of the Debian woody (version 3.0) files and note their location in the directory tree so you will know where to point jigdo when prompted.
9. Do the previous step above for the other six. The download will take a while so you can watch a movie or do this before you go to sleep at night.
10. After successfully completing the download, an .iso file is generated in each of the seven folders. Check that the md5 sum of each file is correct against the one provided on the root directory where the jigdo and template files were located. Md5 sum is a 32-byte string that is generated by a utility to make sure the file copied or created is the same as the source. If they match, it is now its time to fire up your CD burner and burn the seven .iso image files into seven blank CDs. Don't forget to label them properly as you will be prompted the names of the CDs during install. Now your official Debian GNU/Linux 3.0 CDs are ready.

B. Assembling the hardware

1. As with a cooking recipe you will need utensils to prepare your dish. Below is a list of hardware items you will need with reference to Figure 1 above:

Name	Qty	Purpose/Description
Pentium 100 Notebook or Desktop	1	The Firewall machine. With at least 16MB RAM and about 1.0 GB hard disk (mine is only 800MB). If this has a bootable CD-ROM drive, the IDE adapter and Desktop PC is not needed.
10 Mbps PCMCIA Ethernet NIC or PCI Ethernet NICs if using a desktop	2	One connects to a DSL or cable modem the other to the hub on the internal network.

10 Mbps Ethernet Hub	1	With enough RJ-45 ports to support your internal computers.
RJ-45 terminated Ethernet cables	>3	Number of computers on internal network plus 2 with enough length on them to reach each one but each segment should not exceed 100 meters to satisfy Ethernet signaling requirements.
IDE adapter for notebook HDs to standard IDE cable	1	Notebooks have a different size IDE connector so if your notebook does not have a bootable CD-ROM drive this item will be needed to connect the hard drive to a desktop PC that has a CD-ROM drive to install the software.
PC Compatible Desktop with bootable CD-ROM drive	1	Needs a BIOS that can recognize notebook hard drives and can boot from the CD-ROM drive.
Cable modem or DSL connection	1	I am using hereon a cable modem setup but I would surmise that the DSL setup is not very much different.

Table 1. List of Hardware Components

2. Assuming your notebook does not have a CD-ROM drive, remove the hard drive from your notebook and connect the IDE adapter to the drive. If you do have a notebook or a desktop that does have a bootable CD-ROM drive you can proceed to installing the software.
3. Boot the desktop and note the BIOS settings for your hard disk and system in general. Open the PC compatible desktop, remove the IDE and power supply cables that connects to the hard disk and connect these to the notebook hard disk adapter. Make sure that the jumpers are properly set and the cable orientation is correct.
4. Boot the system. You may need to set up the PCs BIOS to recognize the notebook hard drive if it is not recognized automatically.

C. Installing the software
1. Boot the system and place the first CD into the CD-ROM drive to install a basic 2.2 kernel (Base System) and load necessary source programs.
2. During the install of the Base System, you will be presented with the following menu and depending on your system, the installer will try to guess your setup and prompt you automatically what items need to be configured starting from the first item to the last and might skip some if they are not needed. The following menu items are show:
   Configure the Keyboard
   Preload Modules from a Floppy
   Partition a Hard Disk
   Initialize and Activate a Swap Partition
   Activate a Previously Initialized Swap Partition

Do without a Swap Partition
Initialize a Linux Partition
Mount a Previously Initialized Partition
Unmount a Partition
Install Kernel and Driver Modules
Configure Device Driver Modules
Configure PCMCIA Support
Install Foreign Modules
Configure the Hostname
Configure the Network
Install the Base System
Edit Kernel Boot Parameters
Make System Bootable
Make a Boot Floppy
Reboot the System
View the Partition Table
Execute a Shell
Report a Problem
Restart Installation System

3. I will not go into details for each of the menu items but I suggest you read carefully each of the prompts and I will only highlight those ones that are important to a properly configured system. When you are in the disk partitioning process I recommend the following partitions:

Partition Name	Qty	Purpose/Description
/dev/hda1	10-32 MB	/boot
/dev/hda2	32-64 MB	Swap
/dev/hda3	Balance	/

Table 2. Suggested Hard Disk Partitioning Scheme

4. You will be prompted to partition the disk and upon partitioning of swap, you will be prompted to activate it so that the installer can immediately use the swap partition during the install process. Just make sure there are no important data on your disk, as these will be erased.
5. After the partitions are created and initialized, mount the partitions according to suggestions in Table 2 above. In MS Windows, swap is just a file within the C: drive but UNIX, Linux, and many other OSes implement separate partitions for swap and other major directories for valid technical reasons such as control, performance, etc.
6. From your Windows-based PC that's currently connected to the Internet through the cable modem, run the command "winipcfg" (Win 95/98/98SE/ME) or "ipconfig /all" (Win NT/2000/XP) and note down the information for IP address, subnet mask, default gateway, hostname, DNS suffix, and DNS servers.
7. In one of the steps, you will be prompted if you want a rescue boot floppy. I suggest you create the boot floppy, label it, and place it in a secure place so

you can boot the system if in case something goes wrong such as if you want to make changes and you forgot the root password.

8. After installing the base system, you will be prompted to reboot. Reboot the machine and remove the CD-ROM on the drive and the install scripts will continue where it left off before the reboot. If you want you can run this setup process later by executing /usr/sbin/base-config.
9. On the next part, you will need to use the space bar, tab key and return key for your choices. It will then prompt you on the following items:

Prompt	Response/Reasons
MD5 passwords	Yes, stronger passwords as they can be more than 8 characters while default is only up to 8.
Shadow passwords	Yes, as the passwords cannot be seen from the /etc/passwd file and hence cannot be guessed by password cracking routines.
root password	Remember this and make it a strong one.
Remove PCMCIA module	No, as you will need this to detect your network cards on the notebook and make sure it is working.
apt source	cdrom, as this is what you have downloaded above and is much faster and secure.
Scan another CD	Yes, and do for all seven CDs.
Add another apt source	No, as of this time but this can be added later by editing the /etc/apt/source.list
Security updates	Yes, so you will get the latest patches to your system but this will not work here yet because the box is not yet connected to the net.
tasksel	Yes, to simplify choices on what to load. Choose, laptop and UNIX Server but make sure you disable unnecessary services like telnet and ftp later below when you harden your system.
dselect	No, as this is a very complicated interface and would only confuse you. tasksel is sufficient.
Prompt if you want to continue	The script will then build a package list and check dependencies. Yes, to continue the installation.
Locales	Not necessary if in the US but you may choose local language if you want/need.
Information on statd daemon	To configure in tcpwrappers, edit the /etc/hosts.allow and /etc/hosts.deny.
SSH protocol 2 only	Yes, as SSH2 is much better than SSH1.
SSH privilege separation	On by default as it is more secure (file to edit is /etc/ssh/sshd_config).
ssh-keysign	SUID. To change dpkg-reconfigure ssh
sshd server	Yes, but disable and enable only when needed.
/var/lib/cvs	OK, default but not important.
IrDA settings	Native, can be edited on /etc/irda.conf or via debconf

laptop-net config	/etc/default/laptop-net & /etc/laptop-net/schemes
modules	Not necessary for networking as you will use PCMCIA NICs. You might need to configure for some other devices you need.
DHCP	No. Use the information you got from step 4 on IP address, subnet mask, etc.
netenv	Choose 3 (disable netenv usage at boot). Can be configured at /etc/init.d/netenv

10. Pressing the Alt key and any of the function keys F1 to F10 simultaneously will give you a new login prompt that you can login as root or any user. Login into one of these and copy the kernel source and some other important files from Woody disk 1 that you will need later to reconfigure the system but first load the disk into the CD-ROM drive. Thus for 2.4 kernel, login as root user and issue the following commands: [16]

```
# cd /usr/src
# mount /cdrom
# "apt-get install kernel-package pcmcia-source kernel-source-2.4.18
# tar jxvf kernel-source-2.4.18.tar.bz2
# tar zxvf pcmcia-cs-2.4.18.tar.gz
# ln –s kernel-source-2.4.18 linux
# atp-get remove ipchains
#
```

11. After configuring the basic operating system parameters, it is now time to shutdown and move the hard disk to the notebook where it will be further configured to make a strong firewall. Reconnect the desktop PC hard disk power cable and signal cable, and then reinstall the cover. During boot of the desktop PC, reconfigure the BIOS to its original settings.
12. Install the hard disk on the notebook and boot. Check that the BIOS is properly configured and the hard disk should be the first boot device, and it boots properly. Log in as root to configure the system.

D. Configuring/Securing the system

First things first, do not connect your system to the Internet yet unless you have configured your system properly so that would be after this part has been completed.

a. Install a new kernel

The kernel that was installed is a stock kernel and does not support firewalling so we need to compile and install a new kernel. To do that, we needed to install three packages, kernel–package, pcmcia-source, and kernel-source-2.4.18 in step 8 above during software install.

1. Boot your system and login to a shell prompt. Edit /etc/lilo.conf and make another entry for image=/vmlinuz so that after compiling a new kernel, you will still have the old kernel in case something goes wrong. Part of the file would look something like:

```
.
.
.
image=/vmlinuz
          label=Linux
          read-only
          alias=1

image=/boot/vmlinuz-2.2.20-idepci
          label=LinuxOLD
          read-only
          alias=2
```

2. Issue the following commands, to, (1) activate the changes made to /etc/lilo.conf, (2) go to the kernel source directory, and, (3) create a kernel configuration file:

```
# /sbin/lilo
# cd /usr/src/linux
# make menuconfig       (or "make config" alternatively without the quotes)
```

3. The following parameters are important in compiling a new kernel to make your box a gateway/firewall but there might be some differences with respect to hardware settings but networking and firewalling should be very similar:

#	CONFIG_IP_NF_MATCH_OWNER=y
# Automatically generated make config: don't edit	CONFIG_IP_NF_FILTER=y
#	CONFIG_IP_NF_TARGET_REJECT=y
CONFIG_X86=y	CONFIG_IP_NF_TARGET_MIRROR=y
CONFIG_ISA=y	CONFIG_IP_NF_NAT=y
CONFIG_UID16=y	CONFIG_IP_NF_NAT_NEEDED=y
#	CONFIG_IP_NF_TARGET_MASQUERADE=y
# Code maturity level options	CONFIG_IP_NF_TARGET_REDIRECT=y
#	CONFIG_IP_NF_NAT_SNMP_BASIC=y
CONFIG_EXPERIMENTAL=y	CONFIG_IP_NF_NAT_FTP=y
#	CONFIG_IP_NF_MANGLE=y
# Loadable module support	CONFIG_IP_NF_TARGET_TOS=y
#	CONFIG_IP_NF_TARGET_MARK=y
CONFIG_MODULES=y	CONFIG_IP_NF_TARGET_LOG=y
CONFIG_KMOD=y	CONFIG_IP_NF_TARGET_ULOG=y
#	CONFIG_IP_NF_TARGET_TCPMSS=y
# General setup	CONFIG_VLAN_8021Q=m
#	#
CONFIG_NET=y	# ATA/IDE/MFM/RLL support
CONFIG_PCI=y	#
CONFIG_SYSVIPC=y	CONFIG_IDE=y
CONFIG_BSD_PROCESS_ACCT=y	#
CONFIG_SYSCTL=y	# IDE, ATA and ATAPI Block devices
CONFIG_KCORE_ELF=y	#
CONFIG_BINFMT_AOUT=y	CONFIG_BLK_DEV_IDE=y

CONFIG_BINFMT_ELF=y	#
CONFIG_BINFMT_MISC=y	# Please see Documentation/ide.txt for help/info on IDE drives
CONFIG_PM=y	#
CONFIG_APM=y	CONFIG_BLK_DEV_IDEDISK=y
#	CONFIG_BLK_DEV_IDECD=y
# Plug and Play configuration	#
#	# Network device support
CONFIG_PNP=y	#
CONFIG_ISAPNP=y	CONFIG_NETDEVICES=y
#	#
# Networking options	# Ethernet (10 or 100Mbit)
#	#
CONFIG_PACKET=y	CONFIG_NET_ETHERNET=y
CONFIG_NETLINK_DEV=y	CONFIG_NET_ISA=y
CONFIG_NETFILTER=y	CONFIG_EEXPRESS_PRO=m
CONFIG_NETFILTER_DEBUG=y	#
CONFIG_FILTER=y	# Character devices
CONFIG_UNIX=y	#
CONFIG_INET=y	CONFIG_VT=y
CONFIG_IP_ADVANCED_ROUTER=y	CONFIG_VT_CONSOLE=y
CONFIG_IP_MULTIPLE_TABLES=y	#
CONFIG_IP_ROUTE_FWMARK=y	# File systems
CONFIG_IP_ROUTE_NAT=y	#
CONFIG_IP_ROUTE_MULTIPATH=y	CONFIG_FAT_FS=y
CONFIG_IP_ROUTE_TOS=y	CONFIG_MSDOS_FS=y
CONFIG_IP_ROUTE_VERBOSE=y	CONFIG_VFAT_FS=y
CONFIG_SYN_COOKIES=y	CONFIG_ISO9660_FS=y
#	CONFIG_JOLIET=y
# IP: Netfilter Configuration	CONFIG_PROC_FS=y
#	CONFIG_EXT2_FS=y
CONFIG_IP_NF_CONNTRACK=y	#
CONFIG_IP_NF_FTP=y	# Console drivers
CONFIG_IP_NF_QUEUE=y	#
CONFIG_IP_NF_IPTABLES=y	CONFIG_VGA_CONSOLE=y
CONFIG_IP_NF_MATCH_LIMIT=y	CONFIG_VIDEO_SELECT=y
CONFIG_IP_NF_MATCH_MAC=y	#
CONFIG_IP_NF_MATCH_MARK=y	# Kernel hacking
CONFIG_IP_NF_MATCH_MULTIPORT=y	#
CONFIG_IP_NF_MATCH_TOS=y	# CONFIG_DEBUG_KERNEL is not set
CONFIG_IP_NF_MATCH_AH_ESP=y	
CONFIG_IP_NF_MATCH_LENGTH=y	
CONFIG_IP_NF_MATCH_TTL=y	
CONFIG_IP_NF_MATCH_TCPMSS=y	
CONFIG_IP_NF_MATCH_STATE=y	
CONFIG_IP_NF_MATCH_UNCLEAN=y	

4. After completing all prompts, you need to execute the steps below. [16] What these commands do are: (1) check dependencies so the compile of the kernel has a good chance of being successful, (2) create a new kernel for you notebook (I am using a Toshiba so yours may use a different filename, say, HP or Compaq, etc.) in the .deb format using a relatively high revision number

so it will not be overridden later in case an update is made of the kernel, (3) create a new module for the notebook on the new kernel so that the PCMCIA interface can be used by the network cards, (4) move to a directory above the present, and, (5) install the two new Debian packages created.

```
# make dep
# make-kpkg –revision=5:toshiba.1 kernel_image
# make-kpkg modules_image
# cd ..
# dpkg –i kernel-image-2.4.18_toshiba.1_i386.deb  \
   pcmcia-modules-2.4.18_3.1.33-6+toshiba.1_i386.deb
```

5. Copy the kernel configuration file .config under the /usr/src/linux directory to a floppy or your ordinary user directory as backup.

b. Configure

Configuring a Debian GNU/Linux system using the command line is admittedly more difficult than point and click GUI-based systems. However, there is also more granularity and the satisfaction that you know what is being done "under the hood." You can edit the configuration files using a utility like vi in Debian GNU/Linux. First thing to do is to download the Securing Debian Manual (Debian at http://www.debian.org/doc/manuals/securing-debian-howto/securing-debian-howto.txt file or you can use the HTML version at http://www.debian.org/doc/manuals/securing-debian-howto/). [17] Read through this manual and follow those steps that are appropriate to your system most specifically Section 4, 5, 6 and Appendix A, and I also highly recommend going through David Ranch's TrinityOS [18] recommendations as cross reference to harden your system. The TrinityOS recommendations file is large but you can omit those sections that are not pertinent to your setup. These documents are very detailed and well explained and I have no reason to duplicate them here.

We will be editing a lot of configuration files and I would like to point out Figure 2 below as the hypothetical reference configuration of our network. It is presumed that non-routable private IP addresses starting with 192.168.x.x (0 > x > 255) should be implemented in your network as it can handle a maximum of 254 hosts and is more than enough for a SOHO network. Please refer to RFC 1718 for more information on non-routable IP addresses. [19]

In configuring Debian Woody as a firewall, we need to make sure that it does not become insecure at any point in time most particularly during the boot up process when the rules are being added into iptables.

1. Clear all variables and flush all policies.
2. Start the network interfaces but without forwarding enabled
3. Set up the proper firewall rules/policies
4. Enable forwarding.

With the above procedure we can implement the Debian firewall for a basic setup. Considering that each site has unique requirements and consequently access policies, the rules that you would be setting up could somehow differ with my or any other setup but the above steps still hold.

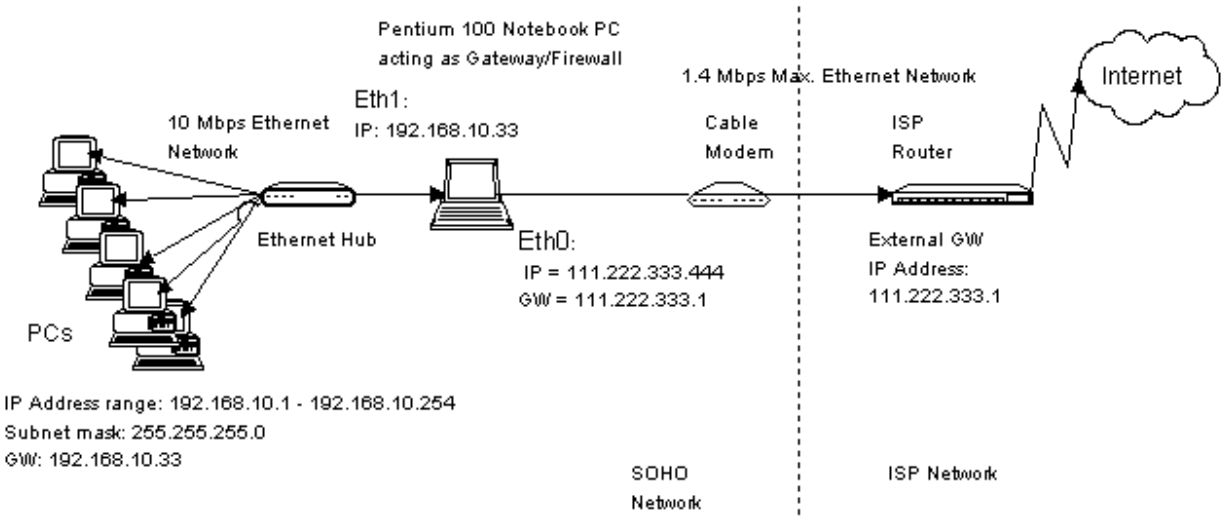

Figure 2. Ethernet Network for a SOHO

Since our ISP (Internet Service Provider) will only provide us with a single IP address and we want to connect several computers to the internet simultaneously, we will be replacing the MS Windows machine currently connected to the Internet via the cable modem, with the machine we are configuring, and, doing NAT (Network Address Translation), masquerading, and packet filtering on the gateway/firewall machine so that it will be the last to handle any outgoing packets and the first to handle incoming packets for all internal computers that communicate with the Internet. [3] There's probably no one more qualified to write the documentation than the one who wrote the code. Paul Russel's documentation for netfilter/iptables does a very good job of explaining the details, as he is also currently the Linux kernel IP firewall code maintainer. [4]

From an internal machine's point of view (refer to Figure 2) it is communicating directly with hosts on the Internet and from a server on the Internet, it is communicating with the gateway/firewall itself and is accomplished by the NAT code on the firewall. The external IP addresses of 111.222.333.444 and 111.222.333.1 are invalid and are only used here for reference since the maximum value of each octet is 255 so both 333 and 444 are beyond the range. You must use the IP address provided by your ISP, via the Windows box where you ran winipcfg or ipconfig previously or else you will not be able to connect to your ISP properly, or, you must configure eth0 to get its address via a DHCP request from your ISP DHCP server. Check with your ISP that you use a static IP address and this is OK with them. If not, then you have to configure the

external facing NIC to use the DHCP of your ISP to get its IP address. I used 192.168.10.33 as the gateway IP address because most commercial routers use 192.168.1.1 and this adds a step of complexity to make your system more secure.

Debian configures the network cards through the file /etc/network/interfaces. For the two cards, you should have something similar to the following:

```
#
# auto eth0
iface eth0 inet static
   address 111.222.333.444
   netmask 255.255.255.0
   network 111.222.333.0
   broadcast 111.222.333.255
   gateway 111.222.333.1

iface eth1 inet static
   address 192.168.10.33
   netmask 255.255.255.0
   network 192.168.10.0
   broadcast 192.168.10.255
```

If you will remove the hash character in front of the second line, this will activate eth0 to use DHCP. The hash character means that anything beyond it on the line is a comment and is usually used to explain or clarify of code so the command interpreter may go to the next line to execute the next instructions.

Configure the routing table by editing /etc/init.d/networking and within the case=start branch, modify as follows:

```
case "$1" in
   start)
         .
         .
         .
         echo –n "Configuring network interfaces: "
         ifup –a
         route add –host 127.0.0.1 lo
         route add –net 192.168.10.0 netmask 255.255.255.0 eth1
         route add default gw 111.222.333.1 eth0
         echo "done."
```

The three route commands, provides the necessary connections so the firewall knows where to send the packets. [20] The Linux Networking HOWTO by Joshua Drake explains this clearly in Sections 5.6 and 5.7. They are being placed here so that every time networking is activated, the proper settings are made, and only after the network has been up. However, this file can be replaced in future revisions to the kernel and thus should be backed up so you will have an easy reference in case it is modified. [21]

We need to configure the rules on what packets to allow, deny, forward, modify, etc. We need to go back to Paul Russel's netfilter online documentation and Section 5.14 the Securing Debian Manual. In Paul Russel's netfilter [4]

documentation page, click on one of the links for Packet Filtering HOWTO and read sections 5 and 6 to understand how to develop the firewall rules for your own setup. On the other hand, Section 5.14 of Securing Debian Manual describes how the files are arranged and the Debian way of locating the various files. Debian differs from other distributions in how the scripts are setup for firewalling but the current recommended setup is to install startup scripts in /etc/network/if-pre-up.d/ and shutdown scripts in /etc/network/-if-post-down.d/. I consciously did not include code for setting up the rules so that these should be studied and understood in detail because they are key to making an effective firewall.

A lot of the smaller details are well covered in the many How-To documents organized by The Linux Documentation Project. [22] The documents are arranged in various document formats and are even available in several languages and all of them are available for free.

5. Conclusion

I hope this document has been useful to all who are contemplating on implementing their own SOHO firewall. It is worthwhile and doable by someone with basic knowledge of computers with enough drive to succeed and in my experience very satisfying because I have a high degree of certainty that I have a secure Internet connection. There is a lot of work involved in reading and understanding various reference materials indicated because of the large body of knowledge required to understand and implement a firewall effectively.

Note that having a firewall is not an assurance that you are more secure than you used to because the firewall is only a tool and as with most tools it must be used properly to be able to attain what it has been designed to do. A firewall can only minimize some of the risks and as exploits are continuously being found, one needs to be vigilant and update the rules as necessary, as well as implement other methods, such as anti-virus software, etc., to have a safe and secure computing environment.

6. References

1. Debian.org. "What is Debian?" URL: http://www.debian.org (9 December 2002).
2. Scheneier, Bruce. Secrets and Lies: Digital Security in a Networked World. New York, NY: John Wiley & Sons, Inc., (2000). Page 188, 191.
3. Little, Keith. "What is a firewall?" URL: http://www.pc-help.org/www.nwinternet.com/pchelp/security/firewalls.htm (7 June 2002).
4. Russel, Paul, et. al. " What is netfilter?" URL: http://www.netfilter.org/documentation/ (26 Aug 2002).

5. Hare, Chris and Siyan, Karanjit. Internet Firewalls and Network Security. Indianapolis, IN: New Riders Publishing, (1995). Page 154, 275.
6. "TCP/IP Suite." URL: http://www.protocols.com/pbook/tcpip.htm (14 July 2002).
7. Semeria, Chuck. "Understanding IP Addressing: Everything You Ever Wanted To Know." URL: http://www.3com.com/other/pdfs/solutions/en_US/50130201a.pdf, URL: http://www.3com.com/other/pdfs/solutions/en_US/50130201b.pdf, URL: http://www.3com.com/other/pdfs/solutions/en_US/50130201c.pdf, (6 July 2001).
8. Spurgeon, Charles. " Welcome to Charles Spurgeon's Ethernet Web Site." URL: http://www.ethermanage.com/ethernet/ (24 Aug 2002).
9. Spurgeon, Charles E. Ethernet: The Definitive Guide. Cambridge, MA: O'Reilly and Associates, (2000).
10. Postel, J., Reynolds, J. "RFC 1700: ASSIGNED NUMBERS." Internet Assigned Numbers Authority. October 1994. Internet FAQ Consortium. " RFC1700." URL: http://www.faqs.org/rfcs/rfc1700.html (2 April 2002).
11. Debian.org. "Mailing List Subscription." URL: http://www.debian.org/MailingLists/subscribe (14 November 2002).
12. GNU.org. "Licenses." URL: http://www.gnu.org/licenses/licenses.html (20 November 2002).
13. Second Session of 105th Congress of the United States of America. "HR 2281." URL: http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=105_cong_bills&docid=f:h2281enr.txt.pdf. Page 6.
14. Atterer, Richard. "jigdo." URL: http://home.in.tum.de/~atterer/jigdo/ (8 December 2002)
15. Lantronix, Inc. "Networking Tutorials Part 2: Adding Speed." URL: http://www.lantronix.com/learning/tutorials/etntadsp.html (22 November 2002)
16. Perry, Sean. E-mail: "Re: Debian Laptop as Gateway." (8 December 2002)
17. Peña, Javier Fernández-Sanguino et. al. "Securing Debian Manual" version 2.7. (15 November 2002). URL: http://www.debian.org/doc/manuals/securing-debian-howto/securing-debian-howto.txt (15 November 2002).
18. Ranch, David. "Linux." 20 October 2002. URL: http://www.ecst.csuchico.edu/~dranch/LINUX/index-linux.html (1 November 2002).
19. Rekhter, Y. et. al. "RFC 1918: Address Allocation for Private Internets." February 1996. URL: http://www.faqs.org/rfcs/rfc1918.html (2 April 2002).
20. Drake, Joshua et. al. "Linux Networking HOWTO" version 1.7.0. Commandprompt, Inc. 2000. URL: http://www.ibiblio.org/pub/Linux/docs/HOWTO/other-formats/html_single/Net-HOWTO.html (21 April 2002).
21. Jackson, Ian, Schwarz, Christian, et. al. "Debian Policy Manual" version 3.5.8.0 dated Nov 15, 2002 section 10.3.2. URL: http://www.debian.org/doc/debian-policy/ (15 November 2002).
22. TLDP.org. "The Linux Documentation Project." 21 November 2002. URL: http:/tldp.org (21 November 2002).