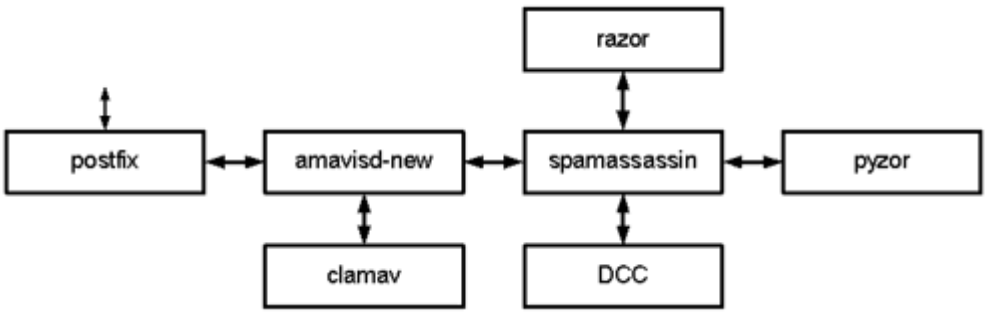
Debian Anti-Spam Anti-Virus Gateway Email Server using Postfix, Amavisd-new, SpamAssassin, Razor, DCC, Pyzor, and ClamAV

This document is not recommend as a guide to upgrade a system from amavisd-new 20030616-p10 to amavisd-new 2.5.3. I designed it as a guide for myself to do a fresh install. Note that I do not install amavisd-new via the Debian package. There may be other versions of this document available. This document uses Debian etch software. Document originally created June 2005. Last revised 03 OCT 2007 by Gary V. The basic software used is Debian etch - Postfix 2.3.x - amavisd-new 2.5.3 - SpamAssassin 3.1.x. Due to the size of this document, please save it to your computer if you plan on using it more than once, then use that copy to view it subsequently. The "Change Log" is http://www200.pair.com/mecham/spam/20061118_changelog.html

---



Introduction:

Index
****
This document was inspired by a document originally created by Scott L. Henderson http://www.freespamfilter.org. It is rewritten to reflect a Debian installation and contains a considerable amount of additional information. In Scott's document, Red Hat Linux 9.0 was used.

Why this document exists:

There is a desire to control the flow of spam and viruses into organizational email systems. Many IT department budgets are tight and many administrators and executives are looking to Open Source solutions to reduce costs. To fight spam, you could buy an expensive appliance or proprietary software, but what if you could take a server you may already own and turn it into a better anti-spam tool than you could buy, without spending a dime on software?

I've found that many administrators of small to medium organizations (say, from 5 to 2000 users) don't yet have the knowledge, experience, or confidence, to build an Open Source system like this powerful anti-spam tool. This document is an attempt to address that situation. With little risk, one can try this spamfilter between an email server and the Internet. If you already have a gateway email server in place in front of your email server, you can place this server on your internal network between the 2 machines or on the Internet in front of your existing gateway. Given a choice however, it is much better to put this server ahead of any other servers. Even if you completely botch something, you can always just yank this system out of the loop and set about repairing it enough to retrieve any queued mail off of it. It is a nice way to get a taste of Linux and get your feet a little wet. I hope you will be as pleased as I have been with the results.

Although any version of Linux, as well as BSDs and other *nixen can (and are) used for this configuration, this document describes using the Debian 4.0 'etch' version (possibly mixed with other packages). Debian was chosen because it is free and has a loyal following. You may find however that Debian requires a little more proactive approach regarding security patches.

My email address is **mr88talent at yahoo dot com**. Support for the various programs used in this document is available from the mailing lists of the respective programs. There is also a forum available for this document and others like it at http://www.freespamfilter.org/forum/.

FAST BUILD BOXES

If you have built this system before, are an experienced Linux administrator, or for other reasons you want to skip all explanations and just perform the

steps necessary to build this spamfilter, each major item that needs to be done is conveniently placed (where the Cat in the Hat keeps all his valuables)**:**

IN A BOX

Commands to be typed in (or copied and pasted) at a command prompt will have a slightly different font than the rest of the text on this page:

like this

And those items that you either need to read, make a decision on, etc, in addition to being "in the box", will be italicized:

like this

Items that display on the screen will be displayed:

like this

It will be assumed that if you follow the 'fast build' boxes, you will understand how to do certain things without explanation, like getting to a command prompt, basic vim commands, knowing when to replace example values (like "example.com" and "example2.com") with your own when appropriate, etc.

****

Document Description:

Index

****

This Guide documents a step-by-step Debian GNU/Linux install using well-established Open Source software. The intended audience is a System Administrator currently running an email server that may or may not have ever used Linux.

This document will allow you to create an anti-spam email relay server. That is, there is no local mail delivery on this box. All inbound mail goes through this system. Spam is filtered out and re-directed to a specified mailbox ("spambin" in our example), or to the intended recipient, depending how "spammy" it is. Ham (non-spam) is passed on to its original intended recipients at your final destination mail server. Thus, a spam "filter" server.

This setup gives the system administrator control over spam, removing the need for end user interaction. In this configuration we will tag a small quantity of email as Spam> and forward it on to the intended recipient, but the vast majority of spam will be quarantined to a mailbox that we specify (spambin). Yes, with experience you can change this configuration. With this setup, if a user actually misses receiving an intended email, it is easy for you

to find it and forward it on to them. It will be sitting in the quarantine area you have configured. The system is capable of using data from LDAP or SQL sources allowing per-user configuration but this document will not cover that aspect.

This system will work best when placed between a firewall and your Exchange server (any kind of SMTP/POP3/IMAP server) or you can sandwich it between your current gateway email server and your Exchange server if necessary (not a good choice). This machine will reject a lot of spam mail. If you place it behind another gateway server, that server will end up creating bounce notices, many of which can never be delivered (because spammers usually fake their address). That server will become what is known as a backscatter source and could eventually end up getting listed on blacklists. It is not good practice to accept a message you later bounce (especially if the sender is forged); if you are going to reject a message, it is best for all concerned to reject it immediately.

The design goal here is to filter and control spam and viruses originating from the Internet. The amavisd-new program was in fact originally written to be an interface between a mail server and various anti-virus packages. AMaViS is derived from "A Mail Virus Scanner". Local delivery of mail on this box could also be configured if desired, but this document will not cover that aspect. This system also has the ability to quarantine (or discard) any email that has attachments that you would like to ban from your organization. Since most email borne viruses come in the form of Windows executable attachments, this system could serve as a first line of defense, with the second line of defense in the form of your desktop Anti-Virus system.

****

Notes:

Index

****

1. This is not a "standard" Linux HOWTO doc. It is written with more detail, step-by-step, so that any Sys Admin who has not used Linux before can set it up. Hopefully it will be informative and useful to experienced Linux/Unix administrators as well.

2. Complete install as per this doc will require a minimum of 600MB of disk space. The system will additionally need whatever amount required for temporary mail storage, as email is spooling through, which depends on your email traffic flow. I can't hope to estimate that for you. It is not a huge amount however; the mail won't generally stay on this system long. If you configure your system to keep quarantined messages on this machine, you will need enough additional space to accommodate.

3. This entire procedure will take an experienced administrator about 7 hours

the first time around. A newbie, roughly twice as long. With some experience under your belt, your second box will take half as long.

4. This doc will not cover hardware problems. It assumes you have Linux compatible i386/ia64/amd64 architecture hardware, including one NIC (network interface card). Generally, your hardware is most likely to be supported if it is neither too old nor too new in respect to chipset and processor technology.

5. Before you begin you'll need to know the IP address, netmask, and other IP configuration details to be used on this box. I won't be helping you with that.

6. You don't need a GUI like Gnome or KDE to build this box so we will NOT install one. It's a waste of 500MB on this particular machine. If you are new to Linux and want to see what they are like, put them on a box other than this one. If you absolutely insist on installing one, do it during the initial setup (using tasksel). I tried to do it after the fact and ended up with a mess. Having a GUI will not make installation or maintenance go faster or easier.

7. My instructions list using the vi (vim) text editor to edit text files. If you are more familiar with another text editor, feel free to use that. If you are used to a Windows environment, vi will seem difficult at first, but I will explain the basic commands you need to get things done. Be brave, you'll be fine, and when you're done, you'll be a little comfortable with the most common text editor in the Unix/Linux world.

8. In this doc, "example.com", "example2.com" and "example3.com" will be the fictitious example domains we'll be receiving mail for. You can receive mail for as many domains as you like with this system. Our example spamfilter mail server will have a host name of "sfa".

9. If you are installing a single hard drive (non RAID) I suggest you start this project with 2 identical hard drives. This however is your choice. You will spend a considerable amount of time creating this server and the easiest, most reliable, and most cost effective way to back up this server is to duplicate the hard drive. I suggest hard drives of at least 4.3GB but it's much better to start out with a pair of modern (fast) hard drives. If you happen to have a pair of identical computers kicking around, with identical network cards, so much the better. I like spares. The hard drives will be wiped clean of all partitions. Use only known good, error free hard drives. This is not the only means of saving yourself a lot of work should your hard drive fail. You could simply ftp the most critical files to an ftp server of your choice. If you are interested in a RAID1 setup, see
http://www200.pair.com/mecham/raid/raid1.html

10. You will need to create at least two new mailboxes on your Exchange server to hold the quarantined spam and banned files. In this document,

these mailboxes will be called "spambin" and "banned"'. You will also need mailboxes named "root" "postmaster" "abuse" and "mailer-daemon". These can be aliases for mailboxes that currently exist, or you can create new mailboxes for them. You may also wish to create a separate mailbox for quarantined viruses; I use "virii". You might want a separate mailbox for "postmaster" because you may receive NDRs there. A NDR, or "Non Delivery Report" is a type of DSN (Delivery Status Notification). The "root" mailbox will receive important system information. The "spambin" and "banned" mailboxes must be monitored, and if "ham" (non-spam) is found, it must be forwarded to its intended recipient(s). Consider whitelisting the sender, so you don't have to worry about it again. With a little experience, you will find ways to quickly glance through the spam and delete it in bulk. You may want to empty the trash on exit. If you create a "virii" mailbox, that must also be monitored.

11. This system can also be configured to discard spam by simply changing one line in a configuration file but this is not appropriate in all organizations. SpamAssassin assigns a score to each email depending on how spammy it calculates it to be. If you choose to discard spam, then you should only consider that option for spam that scores a 14 or higher. There are a couple of ways we can do this, and I will discuss them later.

12. When I refer to the Exchange email server, this term is synonymous with "your current email server" or "your current SMTP/POP3/IMAP server". In other words, it's not specific to Microsoft Exchange.

13. You could build more than one of these boxes and use the second one as a secondary MX email server. Allocate an unused IP address to the secondary server and add a new A record and (secondary) MX record in your DNS records. Please note that this document does not cover issues such as MX records or changes to DNS records or adding reverse DNS records. It assumes that if you have set up your own email server you will have some understanding of these issues. Your ISP may offer assistance here. Be aware that changes to DNS records if improperly done may result in the loss of mail. It can take days for other DNS servers to recognize changes. You need to gain enough knowledge to understand the implications of DNS record changes. If this server is on an internal network, sandwiched between your existing email gateway server and your Exchange server, then no Internet DNS changes are required. In general, don't delete any existing DNS records!

14. There are many ways this spamfilter could be incorporated into your existing setup. I would like to give one possible scenario. Let's say you have one Exchange server handling all your email. You have an MX record set up for it with a priority of [10] and you managed to set up a reverse DNS record for it and everything is working fine. Take one of your unused public IP addresses and assign it to your new spamfilter. Once our new spamfilter is working as it should, create a new 'A' record for it 'sfa' and a new MX record with a priority of [5] 'sfa.example.com.' which will make it your new primary

Mail eXchanger. Your Exchange server will now be your secondary Mail eXchanger. After the DNS records records have been in place for **at least a week** (this gives all the DNS servers time to eliminate their cached records) and the system is running well, you may now configure the Exchange server to accept mail only from your spamfilter(s) and your internal clients (or block port 25 access to the Exchange server at the firewall/router). By this time the only mail that should be coming in directly to the Exchange server is spam, because spammers love to deliberately send mail to secondary servers. The spamfilter will be configured to send all of its mail to the Exchange server. We want to **leave the Exchange server's DNS records alone** so we can quickly re-enable Internet mail to that server in case our spamfilter should fail. If you would like an additional backup email server, just build another one of these and set it up with a priority [7] MX record.

15. It is extremely important that this box is **thoroughly tested** before being relied upon to handle large quantities of mail in a production environment. When building the system, send test emails through one at a time and evaluate what has transpired. Get intimately familiar with the mail.log file. Turn up the level of debugging in amavisd-new. Run amavisd-new in debug mode and monitor the activity. Monitor memory usage using the 'top' program. I will explain how to do these things as the document progresses. The following comments are personal interpretations/observations and may not be technically correct: I have seen on several occasions that even on a properly set up system, when the system is under load, there are memory allocation issues. Some of the processes tend to temporarily allocate enough memory that some of the 'swap' memory is allocated. This memory is then released for use by other processes. If an amavisd-new process begins using swap memory, it runs so slowly that it essentially becomes unavailable to Postfix. The mail then begins to build up in the queue waiting for another shot at it later. This makes matters worse because there is that much more mail to deal with. The system eventually chokes and you are left with thousands of messages in the queue. Lessons to learn here: if this system goes down and you need to take this system out of the loop, make sure you have another system (your original system, if nothing else) in place to accept mail until you get the problem solved. You can disable the content filter (amavisd-new) and requeue the deferred mail and Postfix will at least get the mail delivered and off your system. If you are filtering for multiple domains, start by having only one (least busy) domain have its mail sent through our spamfilter and keep an eye on things. Build from there. If your system exhibits this sort of behavior, it could be an indication you need more horsepower under the hood (or simply more RAM) or your system is not tuned properly.

16. Minimum hardware requirements:
For a small system with a light load (a couple emails per minute, or 2000 messages per day at peak) I suggest a Pentium II 450Mhz and 256MB RAM absolute minimum (384MB gives you a little breathing room). At 10,000 messages a day a 1Ghz PIII with 768MB RAM would be more appropriate. At

50,000, a modern dual processor machine with 2GB RAM should be a decent choice. Amavisd-new appears to use about 53MB per child process (parameter $max_servers) and will use 158MB right out of the box (one master and 2 child processes) and ClamAV will add around 15MB to that. If you configure your system to use more instances of amavisd-new, allocate at least 53MB for each additional instance (68 if you use ClamAV). Amavisd-new can reach 100MB per process if you use a lot of additional SpamAssassin rule sets and/or have large black/white lists. Generally speaking, a fast hard drive and adequate RAM may show more of a performance improvement than a fast processor will, but an adequate CPU is also necessary. Fifty percent more ram than you need is not a bad idea. The programs that run on this server are disk intensive and CPU intensive. A slow hard drive will make this system perform poorly. In this setup we forward quarantined email to another server. If you plan on keeping quarantined email on this server you will need a large enough hard drive to accommodate. You will need a known good floppy drive, a CD-ROM drive and of course, a connection to the Internet. It's a good idea to have hardware that you don't plan on changing. Your spamfilter should ideally be in its final form. Adding additional memory later is fine. If you get errors during installation that appear to be hardware related, find other hardware and start over. If you have a new motherboard, you may or may not get very far. Our Linux kernel may not support every new motherboard chipset out there. If you have an ancient NIC card, or one of a new design, you may not get very far either.

17. Benchmarking:
Sorry, I do not have benchmarking data for high-end use, such as at very large companies or ISPs, but I am aware of several small ISPs that currently run this configuration. The software components in this doc are all designed for high capacity and I would expect them to scale up very well. The 2 main executable programs used herein, Postfix and amavisd-new, both have configurable throttling and performance settings. They are also mature products, with a proven track record and a large following of users. SpamAssassin, with all the work it has to do, fetching information off the Internet on the fly, matching its rules to the content of the messages and such, will tax a machine quite a bit. If you add antivirus filtering, this will also put some pressure on our spamfilter. Large ISPs need powerful multi processor machines and fast SCSI hard drives to make this work well. I have heard of sites processing millions of messages a day using a cluster of 15 high powered multi processor machines. Mark Martinec has also written a paper that illustrates the capacity of a single high power dual processor machine http://www.ijs.si/software/amavisd/amavisd-new-magdeburg-20050519.pdf. Mailscanner is a product similar to amavisd-new. Here are some samples of servers used with Mailscanner: http://wiki.mailscanner.info/doku.php?id=mag:index#setup_examples.

18. This document was created from a Windows user perspective. UNIX/Linux users should have little problem translating Windows specific activities to your environment. I suggest you create a new folder on your

computer, preferably in the root directory of drive C: and call it "debian". This folder will be where all the work files we use will be stored. I would like you to save this html document there now and then open it up again in your browser. We are going to customize this document to make things easier for you. Once we have the spamfilter computer up and running you will do the entire configuration by remote control from the comfort of your Windows computer. Portions of this document can be copied and pasted into the spamfilter computer. **I would like you to customize this document** by doing a search and replace of the elements such as the spamfilter's IP address and hostname. I suggest using WordPad to edit this document. If you have a plain text html editor you like, you may use that instead. Avoid using any editor that modifies the html code. Once you open it in your editor you will see instructions at the top of this document. Go ahead and do that now.

After you edit the document, it might be a good idea to print it out so you can check things off as you complete them (it's about 80 pages).

19. Precautionary note: Be very cautious obtaining any and all software from links on these web pages. Spend some time looking at the URLs on the web pages that pop up. Verify the web sites are legitimate. You could be reading a forgery of this document designed to cause malice. The websites that you link to could be hijacked. Read the disclaimer.

20. Every link in this document opens in a new window, so if it appears nothing happens when you click a link, take a look at your Taskbar.

21. Assuming you are not using a RAID configuration and would like the ability to clone your hard drive:
If you have an IDE CD-ROM drive, set the jumper on the CD-ROM drive to SLAVE and install the CD-ROM drive as the secondary slave. Install the CD-ROM drive so it is in the second position of the data cable; so the primary part of the cable sets loose on top of the drive. If you have a tower case, place the CD-ROM drive in the slot that is at least one down from the top. The reason we are doing this is we want to be able to place a duplicate hard drive in the top position. This drive will not be plugged in during the installation. We will only plug it in when we want to duplicate the primary hard drive. This drive will be used to back up the entire system on occasion. Leave the cover off the case while we build the box. In the future you may wish to purchase one of those mobile hard drive racks. This is why we left the top slot free. If you have a SCSI hard drive, well, hopefully you know what to do as far as jumper settings go. This is all optional, but recommended. It would be nice to have some means of quickly recovering the system after a catastrophe.

While you are building the box, it would be a good idea to back it up when you have reached a major milestone. We can simply place the duplicate drive on top of the CD-ROM drive and plug it in (with the power off, of course). Set your BIOS to auto detect your drives (if you have that option). There are two

simple methods I use to duplicate disks, one is to use dd. It makes an exact duplicate of your hard drive bit by bit, sector by sector, even empty ones. This is why (1) your hard drives must be identical; (2) they must be error free. dd does not work well if both of these conditions are not met. dd is also painfully slow, it can take hours for a large (40GB) disk and some people say it is not a reliable way to clone a hard drive. Each hard drive has a 'defect table' and this method of cloning will overwrite that table on the new drive. One other alternative is Norton Ghost 2003 or the Enterprise edition. Earlier versions will not work. Version 8 will work great if you are fortunate enough to have a copy. See Ghost compatibility with Linux. Ghost has the option to clone a drive sector by sector similar to what dd might do. Like dd, the disks should be identical when using this method. At the very least, the destination drive should be the same size or larger and the geometry should be similar. Going to a smaller drive using this method would fail. If Ghost complains that your hard drive has errors, I suggest you run **shutdown -r -F now** and let it reboot. When it starts back up it will run "fsck" which is the conceptual equivalent of "chkdsk /f" in the Windows world. Ghost 2003 comes with SystemWorks 2003 or often comes with motherboard software. Ghost 2003 also works in (the recommended) normal mode but after the disk is cloned the boot record must be repaired on the cloned drive by first booting to a rescue disk. Actually, I have had great success using the -ib (Image Boot) setting. It appears to make an exact duplicate of the boot sector, then does a regular clone operation on the remainder. There is other software out there that has the ability to clone Linux hard drives. I'm just used to Ghost.

Let's talk about dd. Here's an example of how it can be used. Boot up using the etch CD, answer the first few prompts (Language, Country, Keyboard), let it discover devices (but go no further!), then use [Alt]+F2 to open a console. Log in as root. Then, with both drives installed, issue the command:

dd if=/dev/hda of=/dev/hdc bs=8192
This assumes IDE hard drives. Type this very, very carefully. Then wait a long, long time. Your disk drive LED should be lit solid.

if = input file, of = output file. IDE disks are numbered hda hdb hdc hdd - primary master, primary slave, secondary master, and secondary slave - respectively. SCSI hard drives are sda sdb sdc etc. (depending on jumper settings - so take care that sda remains sda when you install a second drive). Any time you clone a hard drive, you should test the cloned drive. If it boots up, reboot it by using **shutdown -r -F now** to repair any potential problems.

22. Linux, Postfix and amavisd-new are flexible, complex systems. There are innumerable ways to configure a spam filtering server like this one. This document will not attempt to teach you everything there is to know about Linux, Postfix, SpamAssassin and amavisd-new, nor are the instructions I provide meant to give the impression that this is the best way to configure this device. This box meets MY needs and hopefully will provide a solid base for others to work with. It is my hope that this document will provide a

degree of familiarity with these programs sufficient enough to enable you to resolve problems as they arise and enable you to reconfigure the system to meet your needs. If you need help with problems, or just want to tweak the system to meet your particular needs and you need help doing so, the mailing lists for the various programs are your best source. When using mailing lists, state the versions of the programs you are using; try to state the issue in a concise manner and provide examples when examples are necessary. Search through the mailing list archives to see if your question has been answered before. I feel the biggest flaw in this document is that you will learn only a little about Linux and the aforementioned programs and you know what they say: a little knowledge is a dangerous thing. If you are new to Linux I suggest at the very least you purchase a Linux Pocket Guide or equivalent to have on hand. Here is a 16 page PDF on Unix command line basics.

23. This machine will have to resolve a lot of IP addresses and read a lot of DNS records. If it takes a long time to retrieve an answer from a DNS server, this delay could affect the performance of this box. It is much better to have a local caching DNS server available than not. You may have a local proxy server capable of caching DNS queries or a server on your network running a true DNS server like BIND or Windows 2000/2003 DNS server. If you do, use one of these as your primary name server. I provide instructions in this document to install a local DNS cache on this machine if you do not.

24. So what do these various programs do? Postfix is a powerful and flexible MTA. In its most basic configuration it receives and routes email. We will configure it to use amavisd-new as a content filter. Postfix will listen on the standard SMTP port 25 and any mail that comes in on that port and is not rejected will be sent to amavisd-new on port 10024. Amavisd-new will process it and send the mail back to Postfix on port 10025. Postfix will then relay it to the intended recipient(s) on another mail server. Amavisd-new acts like a specialized MTA. To prevent the loss of mail in amavisd-new, amavisd-new will not actually say it has accepted a message from Postfix until it has returned it to Postfix (or bounced or discarded it). In our case, amavisd-new will load SpamAssassin and use it as though it is part of the program itself. It will also call ClamAV (clamd or clamscan) to scan email for viruses. SpamAssassin will query Pyzor, Razor2 and DCC servers and the result of the queries may influence the score that SpamAssassin produces. SpamAssassin is a sophisticated system using a number of means to identify spam. It uses hundreds of its own static and dynamic tests and it queries other servers on the Internet in order for it to produce a spam "score". The higher the score, the more likely the message is spam. Razor, Pyzor and DCC are each different in the way they work, but they have at least one thing in common: they are collaborative mechanisms. Computers all over the world feed them spam or spam signatures. If they receive the same spam signatures from many different sources, it is assumed that the message can in fact be considered spam. SpamAssassin checks each email to see if it appears in any of their databases. SpamAssassin also queries a number of other real time blacklists (RBLs) and several URIDNSBL servers. SpamAssassin sends the URLs found in

the message body to URIDNSBL servers to see if they have been blacklisted. These are also collaborative mechanisms that are manually reviewed by humans. SpamAssassin merely scores the email. We configure amavisd-new to take various actions depending on the score. I have found a paper that further describes the actions of these programs and how it all fits together: http://www.giac.org/practical/GSEC/Greg_Williamson_GSEC.pdf

****

Create Debian Installer CD:

Index
****

There are, at any given time, three versions of Debian. They are 'stable', 'testing' and 'unstable'. At the time of this writing 'stable' is named 'etch', 'testing' is named 'lenny' and unstable is always named 'sid'. The Debian etch 4.0r1 netinst CD-ROM is currently the method chosen to install the operating system. A "Business Card" CD is another possible choice, but is some times problematic due to "package churn". It's also possible to use floppy disks.

You need a CD-RW drive and CD burning software on your PC to create the CD from an ".iso" file.

Make a new directory on your Windows computer and call it 'debian' or something. Then download the latest version of the Debian installer for 'etch' and save it there. Go to: http://www.debian.org/releases/etch/debian-installer/. Read the errata while you are on that page. One interesting errata is http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=401435.

Note that there are etch 4.0r1 i386 or etch 4.0r1 ia64 or etch 4.0r1 amd64 CDs available from this location but I have only tested this setup using the i386 CD (32bit). By default it installs the Linux kernel version 2.6.18.

Create a CD from the image using your CD burning software. When you label the CD include the creation date of the software. I use Roxio 5.0 Easy CD Creator. From the Data CD Project window I choose File -> Record CD from CD image. For Nero 5 Burning ROM, choose File -> Burn Image after getting to the data CD window (ISO).

****

Debian Installation:

Index
****

Set the BIOS in your computer to boot from the CD-ROM drive if necessary. This may be a good time to check the date and time set in the hardware clock. If for any reason you cannot create the CD or you cannot boot from the CD,

you may be able to boot from Debian Installer floppies. A separate document http://www200.pair.com/mecham/spam/installer-floppies.html provides instructions on creating the Installer floppies. Before you start the installation, you may need to know what brand and model your Ethernet card is. You may need to know what chip set the card has. I was unable to install a 3Com 3c509 ISA card. The fun part of the installation is that no matter how I describe the process to you, it may be different when you do it. The Debian installation process somewhat follows a linear pattern, but I cannot guarantee what screens will come up. In general, if you are asked questions during installation, the installer guesses what the most appropriate response would be, and it usually is what we want. **If things don't go well** and you need to start over, I strongly suggest you delete everything off the hard drive during [Partition a hard drive]. Starting with a clean slate helps a lot.

Additional information on the Debian Installer is located at http://d-i.alioth.debian.org/manual/en.i386/apa.html

The main objective of the Debian Installer is to gather enough information to enable it to install the Debian base system. It needs to know what language to continue in, what keyboard you are using, what network hardware you have (it should figure this out automatically), and what your network settings are (we don't want it to figure this out via DHCP). We will also have to configure the partitions on the hard drive at this time so we have a place to put the software that is installed.

We are going to erase the hard drive *so make sure you don't have any data on it you might need.* ***Boot up the computer using the Installer CD or the Installer floppy #1.*** *If you use the floppy to boot up, it will prompt you for remaining floppies. I recommend using the CD-ROM. The instructions below pertain to the CD-ROM method in the default "ask as few questions as possible" mode. When the system boots up to the Debian screen, simply* ***press [Enter]*** *at the boot: prompt.*

[**!! Choose Language**]
This determines the language of the installer and picks a keyboard.
This installation has only been tested with English - English
[**Choose country, territory or area**]
Choose what is appropriate

Unplug the ethernet cable.

[**! Select a keyboard layout**]
American English *selects a standard qwerty keyboard.*

There will be a few screens of activity, then this will come up:
[**Configuring the network with DHCP**]
Hit [Cancel] *because we want DHCP configuration to fail.*

Plug the ethernet cable back in.

[**Module needed by your ethernet card**]

If you see this menu, it only means one thing. The Installer does not recognize your Ethernet card.
Look through the list; the majority of cards that are supported will NOT be on the list, this is a list of somewhat obsolete or possibly bleeding edge cards, not the majority. If it does not find your card, try another NIC Card. The machine I am building right now has an old 3Com ISA PnP 3c509, and it's not working, possibly because PnP may not be supported this early in the game, or because there is a bug in the installer. This is going to be an email server, so a reliable NIC card is important. If you have and old ISA NE2000 compatible card you can use the "ne" driver but you will need to know the interrupt and I/O address beforehand. If you have a problem, the fastest way to solve the problem may be to replace the card with another model.

[**!! Configure the network**]
Network autoconfiguration failed

We wanted that to happen, simply press:
[**Continue**]

On the next screen, choose the default of:
[**Configure network manually**]

[**!! Configure the network**]
Make sure Num Lock is on!
[IP address:]
111.111.111.111
[Netmask:]
255.255.255.x
[Gateway:]
333.333.333.333
[Name server addresses:]
444.444.444.444 555.555.555.555
[Hostname:]
sfa
[Domain name:]
example.com

****

Partition the Hard Drive:

Index
****
The disk partitioning software that comes with this version of the Debian Installer seems to be geared toward novices and as such makes a lot of assumptions in order to make partitioning easy. However, if you want to deviate from what the software provides, it is somewhat cumbersome. At this point you can play with the partitioning software all you like. If you have problems, simply erase the disk and start over.

[**!! Partition disks**]

[Partitioning method:]
Choose Guided - use entire disk

[Select disk to partition:]
Choose what is appropriate

[Partitioning scheme:]
You are free to choose any of the three partitioning schemes provided but we need at least 1GB of space for each data partition. If you choose the [Separate /home partition] or [Separate /home, /usr, /var, and /tmp partitions] method of partitioning, ideally you would have 4GB or more for either the /var partition or the /var/spool partition respectively.

If you are building this with a small drive (not recommended) or simply want to maximize disk space (like I often do), choose
[All files in one partition]

For a larger drive you may optionally choose:
[Separate /home partition]

Then arrow up and change the "Mount point:" of partition "#6 logical" from /home to /var
Here is an example of what the finished product could look like:

```
IDE1 master (hda) - 10.0 GB Maxtor 5T010H1
   #1 primary   2.8 GB B f ext3  /     (bootable) (root partition)
   #5 logical 353.7 MB   f swap        (swap partition)
   #6 logical   6.8 GB   f ext3  /var
```

The same drive using and modifying the [Separate /home, /usr, /var, and /tmp partitions] *partitioning scheme: It took me about 10 minutes of playing with the software to figure out how to modify what the partitioning software came up with, this may provide a little better performance (due to reduced file fragmentation) but will also waste more disk space. You need a 6GB or larger drive and a little patience to do this.*
Change the mount point of "#1 primary" from / to /boot
Change the mount point of "#5 logical" from /usr to /

Change the mount point of "#6 logical" from /var to /var/lib (Enter manually) Delete both partitions #9 and #8, then recreate logical partitions #8 and #9 from the free space and change the mount points to what is illustrated below.

Each data partition should be at least 1GB as shown. The /var/spool directory is where our mail queues will be, so it would be desirable to make it 3GB or larger. If you store quarantined mail on this system then you need to make whatever partition it's stored on is adequately large. The default for amavisd-new is /var/lib/amavis/virusmails so in this next example the /var/lib partition should be large.

IDE1 master (hda) - 10.0 GB Maxtor 5T010H1
#1 primary 279.6 MB ext3 B f /boot (bootable)
#5 logical 3.6 GB ext3 f / (root)
#6 logical 1.8 GB ext3 f /var/lib
#7 logical 386.6 swap f swap
#8 logical 1.0 GB ext3 f /var/log
#9 logical 3.0 GB ext3 f /var/spool

Once you have what you like, choose
[Finish partitioning and write changes to disk]
[Write changes to disk?] [**Yes**]

[**! Configure time zone**]
[Select your time zone:]
Simply choose what is appropriate.

[**! Configure the clock**]
[Is the system clock set to UTC?]
If this comes up it may be an indication the system clock is set to UTC. I prefer to set the system clock to local time so I [tab] over and answer [NO] but this is up to you.

[**!! Set up users and passwords**]
This will ask for root's password and allow you to create a "normal" user and a password for that user. Watch your [Num Lock] status. ***Use really good passwords and don't forget them****. Please add one, and just one, normal user here. If you plan on storing mail locally on this machine (not documented here), or even if you don't, create a user who's main purpose in life might be to hold root's mail. I suggest calling the user* **myroot** *or something similar. Keep in mind that all the best hacker tools run on Linux. If a hacker gains root access to this box, your entire network is history.*

[**Installing the base system**]
Wait....

[**! Configure the package manager**]
[Use a network mirror?]
Choose [**Yes**]
[Debian archive mirror country:]
Choose your country
[Debian archive mirror:]
Choose a mirror near you (mirrors.kernel.org works very well in the US)
[HTTP proxy information]
(configure if needed, otherwise leave unconfigured)

Scanning the mirror...

[**! Configuring popularity contest**]
You decide if you would like participate.

[**Debian software selection**]
[Choose software to install:]
This is 'tasksel'. You only want to select 'Standard system' here (nothing else). Use the [spacebar] to deselect 'Desktop environment'. Then, simply [Tab] over and select [**Continue**]. *(I heartily recommend you do not run a GUI; however, if you absolutely insist on doing so, leaving 'Desktop environment' selected is the way to install it).*

Software will download now. I hope you have a fast Internet connection. What software we don't have now, we can easily get later. We are trying to keep this system somewhat clean. We will use apt-get to install most software after the fact. Some software may also be upgraded, and as a result, you may be asked some questions. When asked a question, usually the default answer will be the correct answer.

This section should not show up, but just in case it does:

[**Configuring console data**]
IMPORTANT! choose "Don't touch keymap"
You chose one earlier whether you knew it or not and choosing
any keyboard here may remove the keyboard mapping
and you may not be able to get it back without starting the installation over!

This may not come up, but in case it does:

[**Configuring Exim v4 (exim4-config)**]

[General type of mail configuration:]
choose [**no configuration at this time**]
[Really leave the mail system unconfigured?] [**Yes**]

[Root and postmaster mail recipient:]
The "normal" user we added earlier will display here. This is fine, so simply

accept this. Since all mail will be relayed to another server, this setting will actually end up being ignored. However, if you configure your system to store mail locally, all of root's mail will be redirected to this "normal" user's mailbox. This is necessary because you typically cannot access root's mailbox remotely.

[**! Install the GRUB boot loader on a hard disk**]
[Install the GRUB boot loader to the master boot record?]
If you would like the install the GRUB boot loader choose [Yes]
If you would like the install the LILO boot loader [Tab] over and select [Go Back]
Then select the 'Install the LILO boot loader...'
[**Finish the installation**]
Remove the CD or floppy when prompted, then hit [**Continue**] *This will reboot.*

Once you get the login prompt, login as root *and issue the following command:*
apt-get install ntpdate ssh vim gnupg

It may ask you to insert the installation CD; do so, then please remove it afterwards. We installed ntpdate so we can set our clock to the correct time. Note that if you have problems communicating with the download server (download seems stuck at [0%] - nothing seems to be happening for a long time), you can use [Ctrl]+c to break out of the communication session then try again. You should not use [Ctrl]+c when software is actually installing however, doing so could trash your system.

Enter the following command:
dpkg-reconfigure locales

[**Configuring locales**]
You use [PgUp] [PgDn] [up-arrow] [down-arrow] [tab] and [spacebar] to navigate and select.
The etch installer software installed en_US.UTF-8 UTF-8 on my system. *I suggest you install the en_US ISO-8859-1 locale (in addition to any other ISO-8859-x locales you may require). If you need to change the locale, or add additional locales, use the [arrow] [spacebar] and [tab] keys. A UTF-8 locale should not be used as the default system LANG (set in /etc/environment or /etc/default/locale), SpamAssassin and amavisd-new may have problems if you do. You should keep the UTF-8 locale in addition to the ISO-8859-x file or Perl may complain.*

[**Which locale should be the default in the system environment?**]
I suggest you do NOT choose [None], I suggest you choose [en_US] or other non UTF-8 locale (an ISO-8859-x locale).

We need to make sure we have a keymap file:

ls -l /etc/console

This lists the contents of the /etc/console *directory. You should see a file named "boottime.kmap.gz"*
If you get "Total: 0" then we have no keymap file.

***If, and only if, we have no keymap file**, run the command:*
dpkg-reconfigure console-data

And choose [**Select keymap from arch list**]
Follow the prompts that apply to you and when the program exits check again to see if there is now a file called "boottime.kmap.gz" in the /etc/console *directory.*
If the file is not there, reboot and try again. We cannot continue until a keymap file is installed. Worst case is we would have to start the installation over again!

Note that you can use the [up-arrow] key to recall previously entered commands (which can then be edited and executed).

Once you are back at the shell prompt, reboot the system with:
reboot

This is the end of the basic Debian installation.

If you don't feel good about the way things went, or you would like to experiment with one of the other methods of installation, this would be the time to start over from scratch!

****

PuTTY and additional programs:

Index
****
Now turn the monitor off and head on over to your trusty old Windows computer. We are going to configure every thing else from there!

You should have this document open in a window on your Windows computer because we are going to use a Windows SSH client called PuTTY to operate our spamfilter remotely. I am going to save you a lot of typing because you are going to select text with the mouse, copy it to the clipboard with [Ctrl]+c and then paste it into the PuTTY window with a right-click of the mouse. This will save you a ton of typing.

Download putty.exe from somewhere like:
http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

Place putty.exe on your desktop, open it up, select SSH, input the IP address of the spamfilter then enter a name for your session in the Saved Sessions box. In the Category window, expand Terminal and click on Features. Check the box "Disable application keypad mode". Just below this in the Category window, click on Window, and increase "Lines of scrollback" from 200 to 400. Click on Session (at the top), then Save. the "Disable application keypad mode" enables us to use the numeric keypad when using vi.

When you use PuTTY again, simply double click on the saved session. Make sure you are at a command (shell) prompt before exiting PuTTY. You can log off the PuTTY window by issuing the command 'logout' or 'exit' or [Ctrl]+d.

If you are selecting some text to paste onto the Linux command prompt (the bash shell), you normally should not select the empty line below the text we are selecting. If you do, this has the same effect as hitting the [enter] key when it is pasted into the PuTTY screen. Sometimes this is desirable. If you select only to the end of the text, you will have the opportunity to review what was pasted before you hit [enter]. Make sure the command line is empty before you paste something into it.

If you are editing a configuration file (we will use the vi editor to do that), you may select multiple lines to copy and paste. To insure there is a newline character at the end of each line I suggest extending your selection to the empty line below the text you are selecting. I will provide empty lines below any text that ends up getting pasted into a document we are editing with vi.

Some of the text in this document will have to be edited either before or after you copy and paste it into the PuTTY window. If you have not already done so I recommend you save this document as a text file or an html file and do a search and replace of key items like your domain name and IP address, then use that document to continue. I suggest using WordPad or your favorite plain text HTML editor to edit this file. Do not use a program that will modify the HTML code. If you open this document in WordPad there are instructions at the top of the page. Once your changes have been made, open the saved document in your browser. You will now have your own semi custom document.

Note.
Your [Home] and [End] keys will work when editing a file using vi, but will not work at the command prompt (you can use [Ctrl]+a and [Ctrl]+e for this).

Open up a PuTTY session now and log in as root.
The command prompt at the bash shell will look something like:
sfa:~#
The ~ (tilde) represents our home directory, and because we logged in as

root, our home is /root

This would be a good time to also download and install WinSCP. WinSCP is a great GUI file browser that lets you transfer files between your Windows machine and your new Debian box. You can also edit files on your Debian box from your Windows machine using WinSCP's editor. I suggest when you save sessions you leave the password blank so you are prompted for it each time you log in.

Please install and configure PuTTY and WinSCP per the notes above.

Important note: The machine is very vulnerable at this point. Any time you are not working on the spamfilter, you should unplug the ethernet cable! This machine should be connected to the Internet only when necessary to configure it. We need to get all our security measures in place before it "goes live". You will need to leave it plugged in to complete the installation however. If you are familiar with editing files on a Linux system, it might be a good idea to jump ahead to "Create Firewall Rules" and then return here to continue.

****

The 2 minute vi tutorial:

Index
****
We are going to use vi (vim actually) to do most of our editing. Fortunately we only need to learn a few commands to be able to accomplish our tasks. There are 3 operating modes in vi. There is the "Command" mode, the "Write" mode and the "Command line" mode. When you first open a file for editing, you are in Command mode. You change to Write mode by entering the letter "i", (short for "insert"). You can edit text pretty much as you would expect in Write mode. You exit out of Write mode and return to Command mode by hitting the [Esc] key. There are many commands that can be learned in Command mode but we only need to learn two more in addition to "i". Those commands are ":" (a colon) and "/" (a forward slash). The colon is used to enter the third mode, the Command line mode and the slash enables the Search command. When you are in Command line mode, you will see a colon at the bottom of the screen. Here is a list of commands we will use while in Command line mode:

:q quit (provided you have not made any changes) By the way, the lower case q is used often in *nix as a way to exit a screen.
:q! exits vi and discards changes (great when you trashed the file and just want to start over!)
:wq saves the changes and exits vi (write and quit)
:w saves the current changes but does not exit vi (write)

And in command mode:
G The capital "G" Goes to the bottom of the page (very handy)

And here is how the Search command works:
/text_to_search_for moves the cursor to the first occurrence
of text_to_search_for

Once the first occurrence of the text we searched for is found, use a lower case 'n' to find the next occurrence.

That's all we need to know for now!
If you would like a cheat sheet for additional commands:
http://www.fprintf.net/vimCheatSheet.html and
http://amath.colorado.edu/computing/unix/vi/

****

Verify System Settings:

Index
****
We need to take a look at a few important files to make sure they are accurate.

vi /etc/resolv.conf

Make sure our domain name is at the top, in the form:
search example.com

The file should look something like:

search example.com
nameserver 444.444.444.444
nameserver 555.555.555.555

Repair it if it is not. (Use "i", then edit it).
If you made changes, Exit the file with [Esc] : wq
If you did not need to change anything, Exit the file with [Esc] : q

vi /etc/hosts

The top of file should look something like:
127.0.0.1 localhost
111.111.111.111 sfa.example.com sfa
Repair it if it does not. Remember, use "i" to insert. Since we are here, you might as well add any other hosts you would like our spamfilter to know about. I suggest you (at least) put your internal mail server(s) here. Simply append any other entries to the bottom of the list.

If you made changes, Exit the file with [Esc] : wq
If you did not need to change anything, Exit the file with [Esc] : q
If you have a mess on your hands, Exit the file with [Esc] : q! *and try again.*

Since you are using an etch CD, our default language will be an UTF-8 locale. We want our system wide language to be an ISO-8859-x (non UTF-8) locale. You can set the language in /etc/environment (if it exists, otherwise it is set in /etc/default/locale). This file is read when we log in. We need to use a non UTF-8 locale so characters will appear as we expect them to and to avoid other problems. It is best to run amavisd-new in a non-UTF8 locale environment. The 'dpgk-reconfigure locales' program previously automatically updated /etc/environment, but it no longer does when using the etch version so we are going edit it manually (it now updates /etc/default/locale). ***Make sure you have installed a corresponding ISO-8859-x locale*** *for the UTF-8 locale we are going to change:*

```
cat /etc/environment
```

If the above returns "No such file or directory"*, then the setting is in /etc/default/locale and* ***you can skip editing this file****, otherwise please continue.*

```
vi /etc/environment
```

Change LANG from a UTF-8 setting:
```
LANG="en_US.UTF-8"
```

to a non UTF-8 setting:
```
LANG="en_US"
```

Save and exit the file as before. Note: you can ***run the command 'locale' to see the current settings****. It is best to reboot after changing the /etc/environment file. Changes are not recognized until you at least log out, then back in.*

We will use ntpdate to query a couple stratum 2 time servers in order to accurately set the system clock, then use a script (/etc/init.d/hwclock.sh) that will correct our hardware clock each time the system shuts down. We will later install the ntp daemon which will keep the clock accurate while the system is running. Hopefully at least one of these two servers will answer our request:

```
/etc/init.d/hwclock.sh reload
```

If this process hangs and you get a time out error, it's possible you have a bios incompatibility with the hwclock software. This is most common on some

*Dell machines. If **and only if** you have a problem here, perform this next step:*
sed -i 's/HWCLOCKPARS=/HWCLOCKPARS="--directisa"/' /etc/init.d/hwclock.sh

Continue on:
ntpdate clock.fmt.he.net
ntpdate ntp1.tummy.com
/etc/init.d/hwclock.sh reload

*If **and only if** you upgraded from sarge and do not have the hwclock.sh script:*
hwclock --systohc

If you made changes to any of the above files:

reboot
[Ctrl]+d

[Ctrl]+d works the same as 'logout' or 'exit'

FYI, to power down the system, the command is: shutdown -h now

****

Change apt-get settings:

Index
****
In Red Hat you would use "yum update" to get updates to installed packages. Red Hat automatically updates the local database of available packages before it updates packages to the newest version. In Debian, you use **apt-get update** to update the local database of available packages followed by **apt-get upgrade**, to install the latest version of any and all packages it found on our system. This is fine when we are using the 'stable' version of Debian. If you install 'testing' and 'unstable' versions of some (or all) software, this could spell disaster if we allow newer packages to be installed indiscriminately. This could make stuff stop working. Fortunately there is something called "Apt-Pinning" that enables us to prioritize the order of 'stable', 'testing', and 'unstable' software sources. This file has to be created by us. The most succinct explanation of this can be found at http://jaqque.sbih.org/kplug/apt-pinning.html. If you ever use "apt-get upgrade", I strongly recommend using **apt-get -s upgrade** to "simulate" the upgrade process before you actually upgrade. **Make a mental note of this**: if you were to have 'testing' software configured as your top priority, and you were to run 'apt-get upgrade', then many of your programs will be installed from the 'testing' group of packages. Once this happens, those packages will continue to update from the 'testing' branch even if you change your top priority to 'stable'. This action cannot be undone gracefully. Note that you can

prevent any package you want from upgrading by placing the package on hold.
I use `echo "packagename hold" | dpkg --set-selections` to place package 'packagename' on hold and `echo "packagename install" | dpkg --set-selections` to allow it to upgrade.

This next file is critical to the way our system functions. I suggest you read the notes above before you continue. Note that since we installed etch (stable), the default priority for etch is 500 (the default Pin Priority for the stable release).
vi /etc/apt/preferences

Enter this text in the file ("i" to insert) EXACTLY as shown.
Yes, you can select the text with your mouse, hit [Ctrl]+c , and then right-click in the vi editor window.

```
Package: *
Pin: release a=unstable
Pin-Priority: 400

Package: *
Pin: release a=testing
Pin-Priority: 450
```

Exit the file with [Esc] : wq *as usual.*

I recommend you use **apt-get -s install [package]** before you install any package. It lets you "simulate" what would happen. You will find that **apt-cache policy [package]** is also helpful. If you want a package that is an 'unstable' version (or any version that is not top priority), you would have to specifically request the 'unstable' version or change the priority before you install it (unless the only version is 'unstable' or your current version is 'unstable'). For example **apt-get -t unstable install [package]** will install the package and also satisfy dependencies from 'unstable'. If you use **apt-get install [package]/unstable** then apt will try to meet any dependencies from stable. If you use tools like tasksel, you may have to temporarily change the priority prior to installing a new set of packages. The most stable situation is to only upgrade to new packages if a security flaw is found and make sure you have the ability to completely restore the hard drive if upgrades don't go well. So I don't frighten you too much, the Debian package maintainers are amazing, so apt-get usually works very well.

Use **apt-cache** to search the local database for available packages.
apt-cache search [search terms] will find packages that sound like what you want and:
apt-cache show [packagename] will return more details on a particular package.
apt-cache showpkg [packagename] will return more details on a particular package.

apt-cache policy [packagename] will return which versions are available along with the priority of each version.
apt-setup will enable you to change mirrors. The alternative is to edit /etc/apt/sources.list manually (which I prefer).
apt-get clean clears the local repository of all retrieved package files.
apt-get autoclean clears the local repository of retrieved package files of programs that are no longer installed.
dpkg -l [packagename] will list the version and a short description of the package we have installed.

You can also search for packages at http://www.debian.org/distrib/packages or http://packages.debian.org. At a later time you can study these great instructions for searching your local package database: http://newbiedoc.sourceforge.net/tutorials/apt-get-intro/info.html.en Also grab http://www.oreilly.com/catalog/linuxnut4/chapter/ch05.pdf for later review.

We are going to add 'unstable' and 'testing' sources to our list of available Debian packages.

```
cp /etc/apt/sources.list /etc/apt/sources.backup
```

This creates a backup file. Then:

```
vi /etc/apt/sources.list
```

At this point, the contents of the file may look something like this:

```
#
# deb cdrom:[Debian GNU/Linux 4.0 r0 _Etch_ - Official i386 ]/ etch contrib main

deb cdrom:[Debian GNU/Linux 4.0 r0 _Etch_ - Official i386 ]/ etch contrib main

deb http://mirrors.kernel.org/debian/ etch main
deb-src http://mirrors.kernel.org/debian/ etch main

deb http://security.debian.org/ etch/updates main contrib
deb-src http://security.debian.org/ etch/updates main contrib
```

We need to modify this file so the result will look something like this:
(with only the http server unique to your particular system)

```
deb http://mirrors.kernel.org/debian/ etch main contrib non-free
deb-src http://mirrors.kernel.org/debian/ etch main

deb http://security.debian.org/ etch/updates main contrib
deb-src http://security.debian.org/ etch/updates main contrib

deb http://mirrors.kernel.org/debian/ unstable main contrib non-free
deb-src http://mirrors.kernel.org/debian/ unstable main

deb http://mirrors.kernel.org/debian/ testing main contrib non-free
```

```
deb-src http://mirrors.kernel.org/debian/ testing main

deb http://volatile.debian.net/debian-volatile etch/volatile main
```

Note what I have done here. Any lines that use the cdrom have been erased. ([up-arrow] to the top of the file and hold down the [Delete] key.)
The 2 'unstable' lines and the 2 'testing' lines have been copied from the top 2 'etch' lines, and then modified slightly as indicated.
The words "contrib non-free" have been added to 3 of the lines.
An etch 'Volatile' source has been added.
You are welcome to simply copy and paste what I have listed above.
Save and exit the file.

Here's a hint on how to quickly make a copy of the first 2 lines:
Enter "i" to get into write mode, highlight the 2 lines with your mouse then right click your mouse in the PuTTY window.

Because we are using a number of sources, it may be necessary to increase the apt cache limit:

```
echo 'APT::Cache-Limit "25165824";' >> /etc/apt/apt.conf
```

I also suggest adding gnupg keys for a few of the apt sources to apt. Note that if you are unable to retrieve keys from subkeys.pgp.net it might be an indication of some sort of firewall or proxy issue. If that is the case, you may end up having problems with other programs such as the DCC client::

```
gpg --keyserver subkeys.pgp.net --recv-key BBE55AB3
gpg --armor --export BBE55AB3 | apt-key add -
gpg --keyserver subkeys.pgp.net --recv-key 6070D3A1
gpg --armor --export 6070D3A1 | apt-key add -
gpg --keyserver subkeys.pgp.net --recv-key 16BA136C
gpg --armor --export 16BA136C | apt-key add -
gpg --keyserver subkeys.pgp.net --recv-key 276981F4
gpg --armor --export 276981F4 | apt-key add -
```

You must run apt-get update next.

```
apt-get update
```

If you have any problems, please check for errors in your sources.list file and run apt-get update again.

***If you are using a multi-processor machine**, then use a multi-processor kernel!*
To locate available smp kernels for etch, you could run:
apt-cache search linux-image | grep smp | grep linux-image

If you are running a 2.6.18 (etch) kernel and have a dual core Intel system

you could for example use the 'linux-image-2.6-686-smp' kernel. You would pick the kernel that most closely matches your system (and your current kernel). To install it, you would simply run:
apt-get install linux-image-2.6-686-smp

If you were to install a new kernel, please reboot *afterwards.*

*Earlier I mentioned an errata dealing with tcp_window_scaling. You may want to consider what may happen (large files fail to transfer between systems) when there is a buggy router between you and someone else, and may wish to make this change to the system (**you decide**):*

```
echo "net.ipv4.tcp_wmem = 4096 65536 65536" >>/etc/sysctl.conf
echo "net.ipv4.tcp_rmem = 4096 65536 65536" >>/etc/sysctl.conf
sysctl -p
```

I am going to assume this may slow down communications between systems under certain circumstances. Here is another setting I have not tried: http://en.wikipedia.org/wiki/TCP_window_scale_option.

****

Navigating the system:

Index

****

A quick word about less. less is a great file and directory viewer.
You can [page-up] and [page-down] and search for text in the same manner you search for text using vi.
Use a lower case "q" to exit less.
To view a file using less:

```
less /path/file
```

To view directory contents, pipe it through less:

```
ls -l | less      (current directory, or)
ls -l /path/directory | less
```

I also like this one:

```
history | less
```

then enter an upper case G to go to the bottom of the display.
It displays the commands you have entered previously. Use q to quit. You can even search the same way you do with vi.

I don't mean to break your concentration, but there is another cool tool called locate. locate allows you to search a database of every file name on the system. It's kind of like Windows Find. You first have to build the database with the updatedb command, and then you can search through it.
Try this: we are going to use locate and less together:

```
updatedb
```

locate kmap | less
What you are looking at is every keymap file on the system along with any file name that has the string "kmap" in it.
Play with it; then "q" to exit less
Now we are going to do something cool.
Take your mouse and highlight any directory you see above, only highlight the directory and not past it.
For example: /usr/share/keymaps/i386/qwerty/
Now right click your mouse anywhere on the screen. You will notice the text has been pasted to the command line.
Now [left-arrow] over to the beginning of the line (or hit [Ctrl]+a) and type in:
cd
Put a space after cd and hit [return]. We just saved ourselves having to type the entire path name just in order to change to that directory. I like that.
OK, simply enter cd to get back home.

Sorry for the diversion.

****

Create Firewall Rules:

Index
****
I like to set up a firewall on the boxes I biuld. This is a subject that could (and does) fill volumes. We are going to use something quick and simple that will give us a basic firewall. Something is better than nothing, and we just don't have time to read volumes on the subject right now.

I worked for a couple days trying to figure out what iptables was all about. I tried using tools like lokkit and shorewall and others, only to get frustrated and confused because I kept getting errors and the firewall simply would not work. My best guess is iptables did not like any rules file it did not create itself. Lokkit was a snap in Red Hat, and a nightmare in Debian. That's pretty much how this whole experience went. But I'm learning a LOT more about GNU/Linux by working with Debian. After day 2 it dawned on me iptables is somewhat like Cisco access lists (which I am a little familiar with).

I am going to give you a set of commands below that I want you to paste into the command line, in the correct order. You MUST change the IP addresses to fit your needs, if you have not already done so. The line with '--dport 22' on it is SSH and the network address to the left needs to be the network that both your computer and the spamfilter computer are on. You could also limit access to a single computer (yours, of course) by using your_ipaddress_goes_here/32. This is a security measure. If you do that part wrong, it will lock you out. The lines with '--sport 53' on them are for access to DNS servers. BTW, all you have to do to change your DNS

servers is change the entries in /etc/resolv.conf.

If you would like to add more rules in the future or make modifications, simply copy and paste these lines into a text editor like notepad, make the changes you would like, and then copy and paste them to a command prompt in your PuTTY window. You can copy and paste all the lines at once. The first line deletes all the entries that were in the rule-set previously and the next to the last line saves the new rule set. The last line shows how one would load a rules file into iptables. Keep a copy of the text file on your computer and call it firewall-rules.txt. I learned to never edit the /etc/firewall-rules file directly on the spamfilter computer. It looks like iptables will reject the file if anything other than itself has modified it.

DO NOT USE AS IS, CHANGE NETWORK ADDRESS FIRST IF YOU HAVE NOT ALREADY DONE SO:
You can copy and paste this whole section to the command prompt:

```
iptables -F
iptables -N FIREWALL
iptables -F FIREWALL
iptables -A INPUT -j FIREWALL
iptables -A FORWARD -j FIREWALL
iptables -A FIREWALL -p tcp -m tcp --dport 25 --syn -j ACCEPT
iptables -A FIREWALL -p tcp -m tcp -s 222.222.222.222/24 --dport 22 --syn -j ACCEPT
iptables -A FIREWALL -i lo -j ACCEPT
iptables -A FIREWALL -p udp -m udp --sport 53 -j ACCEPT
iptables -A FIREWALL -p tcp -m tcp --sport 53 -j ACCEPT
iptables -A FIREWALL -p udp -m udp --dport 123 -j ACCEPT
iptables -A FIREWALL -p udp -m udp --sport 6277 -j ACCEPT
iptables -A FIREWALL -p udp -m udp --sport 24441 -j ACCEPT
iptables -A FIREWALL -p tcp -m tcp --syn -j REJECT
iptables -A FIREWALL -p udp -m udp -j REJECT
iptables-save > /etc/firewall-rules
iptables-restore < /etc/firewall-rules
```

Now run:

```
iptables -L
```

To list the rule set. This is informational only.

We have written the firewall rules to a file on the spamfilter computer and then used iptables to load the rules, but iptables starts with an empty rule set each time the computer restarts. The rule set we saved to /etc/firewall-rules must be loaded into iptables every time the system starts up.

We are going to insert the command to configure iptables into a file that starts up the network interfaces when the system boots up:

```
vi /etc/network/interfaces
```

And insert the following text (remember, it's "i" to insert) in the blank line ***just below*** *"iface lo inet loopback":*
pre-up iptables-restore < /etc/firewall-rules

Save and exit the file as usual with [Esc] : wq
From now on I will assume you know how to edit, save, and exit files using vi. If not stated, it will be implied that after editing a file, you need to save and exit it,
or if necessary, discard changes and start over.

Please don't think this is where you would stick any old command you would like. This is not the place, and not the way, to do so. That's a whole 'nuther subject. This file is the right place (along with /etc/resolv.conf) to change network settings however.

That's all there is to it. You have just used what I believe is the fewest possible steps to create a simple functional personal firewall for this machine. I will admit that it should have been a lot easier by utilizing one of the firewall tools, but it just didn't work out.

At this point our firewall allows external users to connect to SSH and Mail. It also allows replies from Pyzor, DCC, DNS servers and NTP servers. It blocks (I hope) everything else except any sessions that originate from us. This allows us to connect to the outside world. This box should be behind another firewall at any rate. If so, that firewall/screening router will need to be configured to allow tcp port 25 traffic to this machine, but only after this box is fully functional. If you have things locked down really tight; take a look at http://flakshack.com/anti-spam/wiki/index.php?page=Provide+firewall+access for some ideas. Keep in mind we also need udp port 24441 for Pyzor and access to external DNS servers. As far as DCC, Razor and Pyzor go, try them before you start messing with your Internet firewall. I have my spamfilter behind a screening router, a hardware firewall, and software NAT box firewall and none of them required reconfiguration for these programs to work. Port 25 SMTP will probably need to be opened however.

If you have not done so, reboot *again and run* iptables -L *to verify the firewall loaded during start up.*

If you have problems, enter the command iptables -F from the console to clear out iptables. This will allow you another shot at it.

****

Disable Unnecessary Daemons:

Index
****

We are now going to remove some services (daemons) that start up at boot time. I only want you to remove the services I have listed below, no more than that. You could (and probably would) end up with an unusable system if you disabled more than this. Our basic system does not start up many services anyway but "you can't hack a service that isn't running". The only secure system is a system that doesn't exist.

Below is a list of commands I found useful to determine what services were running.
Run them one at a time **if you care to**.

```
top
ps afx
ps afxl
ps -A
ls -F /etc/rc2.d
lsof -i | grep LISTEN
lsof -P | grep LISTEN
netstat -pn -l -A inet
netstat -pn -l inet
```

These are from http://linuxgazette.net/issue89/gonzales.html#4

I also liked the lsconfig script I found here:
http://www.shallowsky.com/software/scripts/lsconfig
Save it as /usr/bin/lsconfig and make it executable.
Like this:

```
cd /usr/bin
wget http://www200.pair.com/mecham/debian/lsconfig
chmod +x /usr/bin/lsconfig
lsconfig
```

If you run lsconfig, the stuff just scrolls by on the screen. You can choose "Copy all to Clipboard" from the drop down menu of the PuTTy window. Click on the two little computers in the upper left-hand corner of the PuTTY window to access the menu. Then open a spreadsheet and paste it into it. Play with it from there.

We need to make a backup of the init scripts in /etc/init.d because after we remove some services, the system may delete the scripts.

```
cp -r /etc/init.d /etc/init.d-original
```

These commands assume you are not hooking up a printer to this machine and you are not using NFS (Network File System). Feel free to copy and paste these next two boxes in their entirety.

```
/etc/init.d/lpd stop
update-rc.d -f lpd remove
/etc/init.d/nfs-common stop
update-rc.d -f nfs-common remove
/etc/init.d/portmap stop
update-rc.d -f portmap remove
/etc/init.d/exim4 stop
update-rc.d -f exim4 remove
```

The inetd service (InterNET Daemon) starts multiple services that can be enabled or disabled individually.

```
update-inetd --disable time
update-inetd --disable daytime
update-inetd --disable echo
update-inetd --disable chargen
update-inetd --disable ident
update-inetd --disable discard
```

Check that we got everything:

```
lsof -i | grep LISTEN
```

The only daemon you should see is at this point is *:ssh

If there are other programs shown, try rebooting and test again.

If you would like to get any of these services back, we can reverse the events. For example, to enable 'ident':

```
update-inetd --enable ident
```

For example, to re-enable the nfs-common service, and start it up right now:

```
update-rc.d nfs-common defaults
/etc/init.d/nfs-common start
```

If you get an error that the file does not exist, first restore it from the backup we made, and then try again:

```
cp -i /etc/init.d-original/nfs-common /etc/init.d
update-rc.d nfs-common defaults
/etc/init.d/nfs-common start
```

This is an example only, you probably don't want to enable NFS.

We have just made our machine more secure than when we started. I will talk about additional security measures at the end of this document, but we want to get this thing up and running first! We want to see this puppy actually do something! You used a few commands above to help you see what services were running before we made changes. Run them again, if you like, to see the effect of disabling them.

****

Configure the NTP daemon:

Index
****

Install the program:
apt-get install ntp

We actually may not need to configure the ntp daemon (ntpd). We installed ntp which does a good job of setting everything up for us. It is set up to use a different time server each time the daemon starts up (http://www.pool.ntp.org/). It configures our machine as an ntp client. If you have a favorite ntp server that you wish to use you can edit /etc/ntp.conf and insert it per the example in the file. NTP is a flexible and complex system so I leave it up to you to research it further if you care to. If you care to choose your own servers from the list of Public NTP Secondary (stratum 2) Time Servers at http://support.ntp.org/bin/view/Servers/StratumTwoTimeServers we can use the little ntpdate program to quickly test them prior to insertion in /etc/ntp.conf:

For example:

```
/etc/init.d/ntp stop
ntpdate clock.fmt.he.net
ntpdate ntp1.tummy.com
/etc/init.d/ntp start
```

By the way, the command to modify the date and time is date and to change the time zone it's tzconfig

Since we are using the etch CD, our default system editor will be nano, and not vim. We soon need to edit a system file called crontab but we don't want to have to learn another new editor so we will change our default system editor:

vi /root/.profile

and just below the line "fi" insert this entry:
export EDITOR=/usr/bin/vim.basic

Save and exit the file, then logout of PuTTY ([Ctrl]+d), then connect back in.

****

Installing Programs:

Index

****

We need to install a number of additional programs. Go ahead and select **ALL** the text in the box below with your mouse, then use [Ctrl]+c to copy it to the clipboard, then right-click the PuTTY window, then hit [enter] to issue the command. **I suggest you select from right to left (bottom to top). Going the other way always wants to select one extra space character which can be a problem with apt-get commands.**

```
apt-get install arc arj autoconf automake1.7 bzip2 cabextract db4.4-util libarchive-tar-perl libarchive-zip-perl libauthen-sasl-perl libberkeleydb-perl libconvert-binhex-perl libconvert-tnef-perl libconvert-uulib-perl libdb4.4-dev libdbd-mysql-perl libdbi-perl libdigest-hmac-perl libdigest-sha1-perl libhtml-format-perl libhtml-parser-perl libhtml-tagset-perl libhtml-tree-perl libio-multiplex-perl libio-socket-ssl-perl libio-string-perl unrar arj
```

And there are more:

```
apt-get install libio-stringy-perl libio-zlib-perl libldap2 libmail-spf-query-perl libmailtools-perl libmime-perl libnet-dns-perl libnet-ldap-perl libnet-ph-perl libnet-server-perl libnet-snpp-perl libnet-telnet-perl libsocket6-perl libtimedate-perl libtool libunix-syslog-perl liburi-perl libwww-perl lynx lzop make ncftp nomarch pax perl-doc rblcheck unzip zip zlib1g-dev pyzor razor libcompress-zlib-perl psmisc re2c curl
```

```
apt-get install lha

If this fails you are probably using the amd64 kernel, if (and only if) lha cannot be installed, you can try a Java based lha if you like (but you will also install a bunch of Java stuff):
apt-get install jlha-utils
```

For lha license information see http://lists.debian.org/debian-devel/1999/11/msg00549.html and for rar (unrar) see http://www.rarsoft.com/index.htm

This is a new system, so let's make sure everything is current.

```
First, run a simulation:
apt-get -s upgrade

Then if everything is as you might expect:
apt-get upgrade
```

Upgrading certain programs may bring up dialog boxes requiring you to respond to configuration questions. Typically, the default answers are OK.

```
If the kernel is upgraded, once the upgrade process is complete, you must:
reboot
```

We are using apt-get to download and install most of our core programs. Note that when we install Postfix, apt-get is smart enough to remove exim4

(because it conflicts with Postfix). This document is based of the following versions of these programs: postfix 2.3.x spamassassin 3.1.x amavisd-new 2.5.3. The instructions may differ significantly if newer (or older) versions are installed. Please run:

```
apt-cache policy postfix spamassassin
```

This will give output similar to the following:

```
postfix:
  Installed: (none)
  Candidate: 2.3.8-2+b1
  Version table:
     2.4.6-1 0
        400 http://mirrors.kernel.org unstable/main Packages
        450 http://mirrors.kernel.org testing/main Packages
     2.3.8-2+b1 0
        500 http://mirrors.kernel.org etch/main Packages
spamassassin:
  Installed: (none)
  Candidate: 3.1.7-2
  Version table:
     3.2.3-1 0
        400 http://mirrors.kernel.org unstable/main Packages
     3.2.1-1 0
        450 http://mirrors.kernel.org testing/main Packages
     3.1.7-2 0
        500 http://mirrors.kernel.org etch/main Packages
```

This tells us that the etch versions of Postfix and SpamAssassin will be installed. If we wanted to install the testing version of a program (for example), we would have to override the choices, e.g. apt-get install [packagename]/testing, or if necessary apt-get -t testing install [packagename]. Note that another option is to momentarily make testing the highest priority in /etc/apt/preferences, then override what will be installed, e.g. apt-get install [packagename]/testing. Read this. Remember that it's a good idea to simulate an installation first (using the -s switch). There is a Debian version of amavisd-new available, but we are NOT going to install it. The configuration files for newer Debian (testing/unstable) versions of amavisd-new are not consistent with the typical way amavisd-new is configured. Instead of one configuration file, the newer Debian versions split the configuration files into half a dozen files in a couple different directories. We will instead install amavisd-new from the original author. So with this in mind:

Read instructions above before you proceed.

```
apt-get install spamassassin
```

```
apt-get install postfix postfix-pcre postfix-mysql postfix-ldap
```

Debconf will pop up a Postfix configuration screen.

For [General type of configuration?] *select:*
No configuration

Don't worry, we will configure Postfix in a few minutes.

***If (and only if) you already have amavisd-new installed**, you need to remove it (this will not remove your configuration files which is a good thing). First make sure amavisd-new is the only thing that will be removed by 'simulating' the removal:*
apt-get -s remove amavisd-new

If it is, then remove it:
apt-get remove amavisd-new

If it is not, then you must make a note of any and all programs that will be removed, because you will have to reinstall them. Good luck with all that. ;)

Here are all the steps needed to install amavisd-new. Some of the files it tries to delete or copy may not be on your system but there is no need to panic if it fails to find them. The last few dpkg-statoverride commands will also fail if you have ever installed Debian amavisd-new - but this is not an issue either. We are going to place files in the same place the Debian version of amavisd-new would have.

There may be complaints that some things do not exist and other things already exist. This should not be a problem.

```
adduser --group --system --home /var/lib/amavis --shell /bin/sh amavis
mkdir /var/run/amavis
chown amavis:amavis /var/run/amavis
mkdir /etc/amavis
mkdir /etc/amavis/en_US
mkdir /var/lib/amavis/tmp
mkdir /var/lib/amavis/db
mkdir /var/lib/amavis/var
mkdir /var/lib/amavis/virusmails
chown -R amavis:amavis /var/lib/amavis
chmod -R 750 /var/lib/amavis
cd /etc/amavis
wget http://www200.pair.com/mecham/amavisd/2.5.3/amavisd.conf
```

Did you get that wget download OK? If you did, then continue on. If not, then you won't be able to get other downloads from me, so you have a major problem.

```
wget http://www200.pair.com/mecham/amavisd/2.5.3/amavisd.conf-sample
ln -s /etc/amavis/amavisd.conf /etc/amavisd.conf

cp -r /etc/amavis/en_US /etc/amavis/en_US-backup
cd /etc/amavis/en_US
rm charset
rm template-dsn.txt
```

```
rm template-spam-admin.txt
rm template-spam-sender.txt
rm template-virus-admin.txt
rm template-virus-recipient.txt
rm template-virus-sender.txt
wget http://www200.pair.com/mecham/amavisd/2.5.3/en_US/charset
wget http://www200.pair.com/mecham/amavisd/2.5.3/en_US/template-dsn.txt
wget http://www200.pair.com/mecham/amavisd/2.5.3/en_US/template-spam-admin.txt
wget http://www200.pair.com/mecham/amavisd/2.5.3/en_US/template-spam-sender.txt
wget http://www200.pair.com/mecham/amavisd/2.5.3/en_US/template-virus-admin.txt
wget http://www200.pair.com/mecham/amavisd/2.5.3/en_US/template-virus-recipient.txt
wget http://www200.pair.com/mecham/amavisd/2.5.3/en_US/template-virus-sender.txt

cd /usr/local/src
wget http://www.ijs.si/software/amavisd/amavisd-new-2.5.3.tar.gz
tar xzvf amavisd-new-2.5.3.tar.gz
cd amavisd-new-2.5.3
test -e /usr/sbin/amavisd-new && cp /usr/sbin/amavisd-new /usr/sbin/amavisd-new-debian
cp amavisd amavisd-new

cp amavisd-new /usr/sbin/amavisd-new
cp amavisd-new /usr/sbin/amavisd-new-2.5.3
cd /etc/init.d
wget http://www200.pair.com/mecham/debian/amavis-init-20030616
mv amavis-init-20030616 amavis
chmod +x amavis
update-rc.d amavis defaults
cd /etc/cron.daily
wget http://www200.pair.com/mecham/crondaily/amavisd-new.txt
mv amavisd-new.txt amavisd-new
chmod +x amavisd-new
cd /etc/cron.d
wget http://www200.pair.com/mecham/crond/amavisd-new.txt
mv amavisd-new.txt amavisd-new
dpkg-statoverride --add amavis amavis 755 /var/lib/amavis
dpkg-statoverride --add amavis amavis 755 /var/lib/amavis/virusmails
dpkg-statoverride --add amavis amavis 755 /var/run/amavis
```

****

Postfix Configuration Part 1:

Index

****

We need some sample files from the Postfix source code.

```
cd /usr/local/src

Change these next lines to match the (author's) version of Postfix you have (hint: dpkg -l postfix ):
wget http://ftp.debian.org/debian/pool/main/p/postfix/postfix_2.3.8.orig.tar.gz
tar xzvf postfix_2.3.8.orig.tar.gz
```

We always place our source code in /usr/local/src.

List the contents of this directory:

```
ls -l
```

We have created a new subdirectory and unpacked the source code into it. Mine is called postfix-2.3.8. We don't need to keep the compressed file. Make sure you are still in the /usr/local/src directory, then remove the compressed file(s) with the command:

```
rm postfix_2*
rm amavisd-new-2.5.3.tar.gz
ls -l
```

When we downloaded our Postfix source code, a number of sample files were included. We want to make use of those sample files so we will copy them to the postfix directory.

The second line below may need to be edited *if your version of the Postfix source code is different than mine.*
MAKE SURE ***you answer "n" to "overwrite?"*** *Do each section separately.*

```
cp -i /usr/share/postfix/main.cf.debian /etc/postfix/main.cf

cp -i /usr/local/src/postfix-2.3.8/conf/* /etc/postfix

cp -i /etc/postfix/header_checks /etc/postfix/body_checks

cp -i /etc/postfix/access /etc/postfix/sender_access
```

****

Edit master.cf:

Index
****

Read this before you complete this section.
I have done the work of configuring master.cf for you and you may simply download the file from me. If you wish to use my file, follow the first 4 steps below.

The master.cf we download here can be used with recent Postfix versions.

```
postfix stop
cd /etc/postfix
mv master.cf master.cf-original
```

```
wget http://www200.pair.com/mecham/debian-postfix-2.3-amavisd/master.cf
```

Note that wget will not normally overwrite an existing file, so we "moved" master.cf to another file first.
Now you can simply jump to Edit Main.cf:

If you wish to do the work yourself, continue on. Or, you may wish to simply read what changes I have made to master.cf.

```
postfix stop
vi /etc/postfix/master.cf
```

Next, we want to give Postfix some information it will need to talk to the amavisd-new program.

Add these lines near the bottom of master.cf. The "-o" is the lower case letter o, not zero. These settings are from http://www.ijs.si/software/amavisd/README.postfix. You can copy and paste this entire section once the cursor is in the correct position (see below) and you are in insert mode. Note: rather than using a right click of the mouse to paste into the editor, you can also use [Shift]+[Insert]:

```
smtp-amavis    unix    -    -    -    -    2    smtp
    -o smtp_data_done_timeout=1200
    -o smtp_send_xforward_command=yes
    -o disable_dns_lookups=yes

127.0.0.1:10025    inet    n    -    -    -    -    smtpd
    -o content_filter=
    -o smtpd_restriction_classes=
    -o smtpd_delay_reject=no
    -o smtpd_client_restrictions=permit_mynetworks,reject
    -o smtpd_helo_restrictions=
    -o smtpd_sender_restrictions=
    -o smtpd_recipient_restrictions=permit_mynetworks,reject
    -o smtpd_data_restrictions=reject_unauth_pipelining
    -o smtpd_end_of_data_restrictions=
    -o mynetworks=127.0.0.0/8
    -o smtpd_error_sleep_time=0
    -o smtpd_soft_error_limit=1001
    -o smtpd_hard_error_limit=1000
    -o smtpd_client_connection_count_limit=0
    -o smtpd_client_connection_rate_limit=0
    -o smtpd_milters=
    -o local_header_rewrite_clients=
    -o local_recipient_maps=
    -o relay_recipient_maps=
    -o receive_override_options=no_header_body_checks,no_unknown_recipient_checks
```

We also need to add two items below the 'pickup' service type. The 'pickup' service 'picks up' local mail (local meaning "on this machine") and delivers it. Later we will create a daily/weekly report that this box will mail to us and because the report will contain contents that will classify the report itself as

spam, this is a way to bypass content filtering for mail generated by this machine.

Add this just below the 'pickup' service type:

```
   -o content_filter=
   -o receive_override_options=no_header_body_checks
```

When you are all done, the table and the lines right after it should end up looking like this:

```
#
==========================================================
=================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
#
==========================================================
=================
smtp      inet  n       -       -       -       -       smtpd
#  -o receive_override_options=no_address_mappings
#submission inet n       -       -       -       -       smtpd
#  -o smtpd_enforce_tls=yes
#  -o smtpd_sasl_auth_enable=yes
#  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
#smtps     inet  n       -       -       -       -       smtpd
#  -o smtpd_tls_wrappermode=yes
#  -o smtpd_sasl_auth_enable=yes
#  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
#628       inet  n       -       -       -       -       qmqpd
pickup    fifo  n       -       -       60      1       pickup
   -o content_filter=
   -o receive_override_options=no_header_body_checks
cleanup   unix  n       -       -       -       0       cleanup
qmgr      fifo  n       -       n       300     1       qmgr
#qmgr     fifo  n       -       -       300     1       oqmgr
tlsmgr    unix  -       -       -       1000?   1       tlsmgr
rewrite   unix  -       -       -       -       -       trivial-rewrite
bounce    unix  -       -       -       -       0       bounce
defer     unix  -       -       -       -       0       bounce
trace     unix  -       -       -       -       0       bounce
verify    unix  -       -       -       -       1       verify
flush     unix  n       -       -       1000?   0       flush
proxymap  unix  -       -       n       -       -       proxymap
smtp      unix  -       -       -       -       -       smtp
# When relaying mail as backup MX, disable fallback_relay to avoid MX loops
relay     unix  -       -       -       -       -       smtp
   -o fallback_relay=
#       -o smtp_helo_timeout=5 -o smtp_connect_timeout=5
showq     unix  n       -       -       -       -       showq
error     unix  -       -       -       -       -       error
discard   unix  -       -       -       -       -       discard
local     unix  -       n       n       -       -       local
virtual   unix  -       n       n       -       -       virtual
lmtp      unix  -       -       -       -       -       lmtp
anvil     unix  -       -       -       -       1       anvil
scache    unix  -       -       -       -       1       scache
```

```
#
smtp-amavis    unix    -    -    -    -    2    smtp
    -o smtp_data_done_timeout=1200
    -o smtp_send_xforward_command=yes
    -o disable_dns_lookups=yes

127.0.0.1:10025    inet    n    -    -    -    -    smtpd
    -o content_filter=
    -o smtpd_restriction_classes=
    -o smtpd_delay_reject=no
    -o smtpd_client_restrictions=permit_mynetworks,reject
    -o smtpd_helo_restrictions=
    -o smtpd_sender_restrictions=
    -o smtpd_recipient_restrictions=permit_mynetworks,reject
    -o smtpd_data_restrictions=reject_unauth_pipelining
    -o smtpd_end_of_data_restrictions=
    -o mynetworks=127.0.0.0/8
    -o smtpd_error_sleep_time=0
    -o smtpd_soft_error_limit=1001
    -o smtpd_hard_error_limit=1000
    -o smtpd_client_connection_count_limit=0
    -o smtpd_client_connection_rate_limit=0
    -o smtpd_milters=
    -o local_header_rewrite_clients=
    -o local_recipient_maps=
    -o relay_recipient_maps=
    -o receive_override_options=no_header_body_checks,no_unknown_recipient_checks
```

(Don't worry about the stuff below this part displayed above - you won't need to change any of those rows, and they are all listed as "pipe" in the last column.)

****

Edit main.cf:

Index

****

Our next friend is the file /etc/postfix/main.cf the main configuration file for Postfix. Following are suggested values to use in main.cf. These have been tested for this configuration and will work fine, but there are many judgment calls involved in this, and it is a good idea at some point to learn more about Postfix configuration on your own. You could first look at the sample Postfix main.cf file /usr/share/postfix/main.cf.dist . There are comments describing some of the most common options. Refer also to the Postfix documents on your machine in the /usr/local/src/postfix-2.3.8/README_FILES directory, or read the documentation on the Postfix web site http://www.postfix.org/documentation.html. I also recommend http://www.postfix-book.com/.

Since we are setting up our spamfilter to relay all of its mail to another server,

we will be using what Postfix considers a "relay domain address class" which essentially means that we will use (change from the default value), out of the 300+ configurable parameters in Postfix, a small group of parameters that serves our purpose best. This address class is described here: http://www.postfix.org/ADDRESS_CLASS_README.html#relay_domain_class. We are also acting as a primary MX for another server so please read this appropriate section: http://www.postfix.org/STANDARD_CONFIGURATION_README.html#backup

It is common in Postfix to store items in lookup tables. We are going to use several hash tables to store data that Postfix will use. Once we have plain text data in these tables, we use the postmap command to create binary files (Berkeley DB format) that Postfix will ultimately use to retrieve the data. For example, if you have a file called "filename" and you "postmap filename", a new file is created "filename.db". When we reference the file as data type "hash:", Postfix will retrieve data from "filename.db", *not* "filename". There are more than a dozen other types of data files that Postfix can use to store data. Hash tables are an appropriate choice for several tables we will use, and pcre (Perl Compatible Regular Expressions) is appropriate for a couple tables we will use to hold content filtering data. In its simplest form a hash table is comprised of 2 pieces of data, a key and a value; typically referred to as the key/value pair. The key and the value are separated with whitespace (typically a space or tab). The data in a typical table that we use in Postfix would look something like:
user1@example.com OK
user2@example.com OK
user1@example2.com OK
Suggested reading: http://www.postfix.org/DATABASE_README.html

OK, lets get going. Note: in commands wherever quote marks " " are used, use them. Rather than editing main.cf directly (which you may nonetheless do, if you prefer) we'll use a handy tool that comes with Postfix, named "postconf". We will use the -e switch, which means to "edit" main.cf.

We simply need to make a correction to the default setting here:

```
postconf -e "alias_maps = hash:/etc/aliases"
```

Now we will create from the text version of the aliases file, the binary version that Postfix will actually use. We do not need to edit the aliases file at this time but it would be a good idea to do so simply to view the contents. You need to run newaliases now, and every time after you edit the aliases file. The newaliases command is just like postmap except that it's specific to the aliases file.

```
newaliases
```

You will see there is now an "aliases.db" file in /etc/. That is what Postfix reads. Now that you have a proper aliases file, it appears that because we are going to configure our system to relay all mail (no mail will be locally delivered), the aliases file will be ignored by Postfix. We instead will set up virtual_alias_maps that we can use for address rewriting should we need to. Other programs may read/write to the /etc/aliases file, so we do not remove it.

myorigin and **mydomain** The domain name that mail created on this machine appears to come from. For example, if one of our programs (cron for example) sends mail from "root" it will be sent from "root@example.com".

```
postconf -e "myorigin = example.com"
postconf -e "mydomain = example.com"
```

Obviously, in the above, and all the following commands, replace my example parameters, like "example.com", with your own specific values.

myhostname The fully-qualified domain name (FQDN) of the machine running the Postfix system.

```
postconf -e "myhostname = sfa.example.com"
```

mynetworks These are the machines I trust, and will relay mail for, to any destination. Generally, this is set to my LAN, or just one, or a few trusted internal mail servers. Along with "relay_domains", **this is an important one to get right** lest you run the possibility of becoming an "open relay". In other words, your box could accept and forward mail to domains for which it has no business doing so. Being an open relay is a serious issue, and can cause you to get blacklisted by various Internet anti-spam lists, among other problems. You can specify a single computer, multiple individual computers, or any computer on a specified network. You can also exclude certain hosts in your network by preceding the IP address with an exclamation point. Excluded addresses need to be listed before included addresses. If you are using a NAT router that substitutes the real client IP address with its own, then you must exclude the IP address of the NAT router from mynetworks. If you will be dealing with multiple internal mail servers, and/or want to allow several machines and/or subnets to relay through this server (careful!), just add them to this parameter in CIDR format, like this:

```
Please read important notes above.

postconf -e "mynetworks = 127.0.0.0/8, 222.222.222.222/24, 10.10.10.10/24"
```

The above will allow the machines on the networks 222.222.222.222/24, and 10.10.10.10/24 to relay smtp mail through this box. You could also specify a single computer's IP address. If you only know your dotted decimal netmask (i.e. 255.255.255.240) and need to convert it to CIDR format, try the http://www.wildpackets.com/products/free_utilities/ipsubnetcalc/overview. (Input an IP address on your network, select the subnet info tab, select your subnet mask, your network is Subnet ID/Mask Bits.) Or simply take a look at

http://www.belchfire.net/webtools/cidr_conversion_table.html.

message_size_limit Maximum size email that Postfix will let in the "front door".

```
postconf -e "message_size_limit = 10485760"
```

The above allows email up to 10MB; the value is in bytes (10*1024*1024). If you increase this consider that mail larger than 10MB may possibly get bypassed by ClamAV (but we can increase ArchiveMaxFileSize in /etc/clamav/clamd.conf to compensate).

local_transport Give an error message for local delivery attempts.

```
postconf -e "local_transport = error:no local mail delivery"
```

mydestination An empty mydestination tells Postfix this machine is not the final destination.

```
postconf -e "mydestination = "
```

local_recipient_maps An empty local_recipient_maps tells Postfix there are no local mailboxes.

```
postconf -e "local_recipient_maps = "
```

virtual_alias_maps Our spamfilter must be able to receive mail for postmaster@[111.111.111.111]. Reportedly, some things actually expect this ability to exist. We will also allow mail to abuse@[111.111.111.111]. Since we do not allow local mail delivery, mail addressed to our spamfilter IP address will get rejected with an error message. Setting up virtual_alias_maps allows email to these two accounts to be forwarded to an inside address. Make sure your Exchange server is set up to receive messages addressed to "postmaster" and "abuse".

Set up a reference to the virtual file:

```
postconf -e "virtual_alias_maps = hash:/etc/postfix/virtual"
```

Then edit the virtual file:

```
vi /etc/postfix/virtual
```

and add postmaster and admin in the format:

```
postmaster postmaster@example.com
abuse abuse@example.com
```

Save and exit the file, then create the binary file that Postfix will use:

```
postmap /etc/postfix/virtual
```

relayhost Generally speaking, if this machine is on an internal network (with no public IP address), you may need to configure 'relayhost'. If it is on the Internet serving as a gateway server, you would only configure 'relayhost' if you must relay your mail through some other server (like your ISP's), otherwise you would not. The relayhost is the email server you wish to use to send outbound email. The outbound email I'm talking about is the non-local mail, mail bound for domains other than ours, not the normal email that comes from outside the system and is bound for your internal mailboxes (we use /etc/postfix/transport to route those). At this time you are using a machine other than this one to send mail out to the Internet so you could place the IP address of that machine here (the brackets must be there). If preferred, you can use a host name instead of an IP address (keep the brackets, i.e. [gateway.example.com]). If this is blank, or not configured, then Postfix will use DNS to deliver outbound mail (which is the ideal configuration). It is best to not use relayhost, but if you do not, you should first have your reverse DNS record in place and of course your "A" record and "MX" record so other servers on the Internet will accept mail from this machine at this IP address. If these are not yet in place, it may be useful to temporarily configure relayhost. If you point relayhost to the Exchange server, the Exchange server (or other SMTP server) must be configured to allow our spamfilter to relay mail through it http://www.msexchange.org/pages/article.asp?id=54. You need this so bounces have a way out of the network.

Read the notes above before you enter this:
postconf -e "relayhost = [666.666.666.666]"

You can optionally reconfigure your clients and other SMTP servers (including the Exchange server) to use our spamfilter for their outgoing mail. Keep in mind that if you use your spamfilter as your outbound SMTP server that your outbound mail will go through the same scanning process as your inbound mail (unless you prevent it - see http://www.freespamfilter.org/forum/viewtopic.php?t=283 and http://www200.pair.com/mecham/spam/bypassing.html). If you use this setting, make sure the relayhost you designate will accept email from this machine and THAT machine is not using THIS machine for ITS outgoing mail.

relay_recipient_maps - We are going to build a table of every single user in every single domain that we accept mail for. This table will be used to reject mail that is addressed to nonexistent users in our domain(s). Don't freak out just yet. At this time we are only going to set up the structure of the table. Then you will come back to this after your spamfilter is functional and work on finding an automated process that can be used to populate the database. Or, if you have a manageable number of users, manually enter them. If this proves too daunting a task, we may have to rely on the Exchange server to do the 'user unknown' rejects. In some cases you might be able to use reject_unverified_recipient as an alternative. If you are in fact using Exchange, there are HOWTOs available that describe automating the

process of building the relay_recipients table. It has been very common of late for spammers to launch 'dictionary attacks'; sending thousands of messages to a domain using fabricated user names. Our spamfilter will have to process each and every one of these unless you put your valid users in the relay_recipients table or use recipient address verification. **Don't underestimate the importance of this** and make sure you are not the only one in your organization who knows how to make changes to this file.

Set up a reference to a file we will create to store the data:
postconf -e "relay_recipient_maps = hash:/etc/postfix/relay_recipients"

Then edit that file:
vi /etc/postfix/relay_recipients

For the moment, we are going to accept mail for all users in our domain(s) so enter each domain you accept mail for in the format:
@example.com OK
@example2.com OK
@example3.com OK

Then create the binary file that Postfix will use:
postmap /etc/postfix/relay_recipients

The entries above are temporary. They are wildcards that allow mail to your domains. You MUST remove the entries above at some point in the near future *and replace them with every single one of your valid recipients' email addresses. When you are ready to enter each user individually in the relay_recipients file, you would first remove (or comment out) the data above that allows mail to all users in the domain, and then list each user individually in the form:*
user1@example.com OK
user2@example.com OK
user3@example.com OK

Actually, in this particular file the value "OK" listed after each user is not used for anything, but *something* must be there because a hash table requires a value after the key. Note that by eliminating root@example.com from this file, you can prevent users from the Internet from sending mail to root@example.com, but don't do this if some of your own servers need to use this machine to send mail to root. If you use Exchange, here are the HOWTOs I promised you. Even if you don't use Exchange, I found the information regarding file transfers useful.
http://www2.origogeneris.com:4000/relay_recipients.html - http://www-personal.umich.edu/~malth/gaptuning/postfix/ - http://www.unixwiz.net/techtips/postfix-exchange-users.html - http://postfix.state-of-mind.de/patrick.koetter/mailrelay/. There is also a vbs script I have been made aware of:
http://www200.pair.com/mecham/spam/PostfixAddressExtract.vbs.txt

transport_maps Tells Postfix where to look for a transport file. We use the

transport file to tell Postfix where to forward valid mail addressed to our domain(s). Our file will be /etc/postfix/transport and we will set it up in similar fashion to relay_recipients.

Create a reference to it in main.cf:
postconf -e "transport_maps = hash:/etc/postfix/transport"

Then edit it:
vi /etc/postfix/transport

Add 1 new line for each domain for which you will be handling mail, similar to the example below. The IP address is that of whatever server is the final destination of messages addressed to our domain(s) (our Exchange server). It does not matter where you place these items in the file, but I like to put them at the top.
example.com relay:[666.666.666.666]
example2.com relay:[666.666.666.666]
example3.com relay:[666.666.666.666]

(DO include the brackets on these lines.
You can also use a FQDN hostname instead of an IP address (i.e. relay:[exchange.example.com]).

Then create the binary file Postfix will use:
postmap /etc/postfix/transport

relay_domains What destination domains (and subdomains thereof) this system will relay mail for. You want to list here ONLY domains for which you are responsible for accepting mail. In addition to allowing mail to be relayed to these domains, this setting also infers that we do not relay mail to domains not listed here and therefore this is a critical anti-relay control setting. Separate the list with commas or spaces.

```
postconf -e "relay_domains = example.com, example2.com, example3.com"
```

If you have many domains to list here, you would want to use a lookup table. An example would be to create a file "/etc/postfix/relay_domains" with the contents of the file in the format:
example.com 1
example2.com 1
(or)
example2.com OK
then set "relay_domains = hash:/etc/postfix/relay_domains" in main.cf, then "postmap /etc/postfix/relay_domains" and "postfix reload" at the command prompt. See http://www.postfix.org/postconf.5.html#relay_domains for details.

recipient_delimiter If your current SMTP/POP3/IMAP server is configured to use address extensions (for example user+foo@example.com) then

recipient_delimiter should be set to match the delimiter you are currently using to separate the user name from the address extension. Amavisd-new also uses $recipient_delimiter and this setting needs to match the Postfix setting. This has nothing to do with the comma you are using to separate multiple people you send email to (user1@example.com, user2@example.com, user3@example.com) using your email client (MUA).

Typical settings - Choose one

I Don't use recipient delimiters:

```
postconf -e "recipient_delimiter = "
```

I currently use the plus sign:

```
postconf -e "recipient_delimiter = +"
```

I currently use the minus sign:

```
postconf -e "recipient_delimiter = -"
```

If (and only if) the IP address you present to the world is not the IP address of your spamfilter (you are configured to run behind a NAT firewall or a proxy server) please add these two lines to main.cf, then you must uncomment and configure proxy_interfaces (1.2.3.4 represents the public IP address):

```
# Specify your NAT/proxy EXTERNAL address here.
#proxy_interfaces = 1.2.3.4
```

Address rewriting. *When using a content_filter like amavisd-new, because mail is sent to Postfix twice, address rewriting takes place twice unless we disable it either before amavisd-new or after amavisd-new has processed a message. Our virtual file is one example of a table that rewrites addresses. We will disable rewriting by placing 'no_address_mappings' in a 'receive_override_options' override in master.cf.* ***It is up to you to decide*** *whether you want amavisd-new to see the original address, or the rewritten address. I find it preferable to disable rewriting prior to amavisd-new. In master.cf you can disable address rewriting before amavisd-new by setting:*

```
smtp    inet  n       -       -       -       -       smtpd
   -o receive_override_options=no_address_mappings
```

Be careful, there are two lines that look similar. You are looking for smtpd on the right hand side. To instead disable address rewriting when amavisd-new returns (non discarded) messages to Postfix, you would edit the current receive_override_options override on the reinjection port (127.0.0.1:10025). For example, if we now have:

```
127.0.0.1:10025 inet  n       -       -       -       -       smtpd
[...lots of overrides are here...]
   -o receive_override_options=no_header_body_checks,no_unknown_recipient_checks
```

You would change this to:

```
127.0.0.1:10025 inet  n       -       -       -       -       smtpd
[...lots of overrides are here...]
```

-o receive_override_options=**no_address_mappings,**no_header_body_checks,no_unknown_recipient_checks

vi /etc/postfix/master.cf *and* ***edit one or the other****.*

****

Postfix Anti-Spam settings:

Index

****

Preliminary Notes:

When a client (a computer trying to send us mail) connects to Postfix and begins a communication session, Postfix records information about that session. Prior to the point where Postfix accepts mail from that session for delivery, we have the option of evaluating the session and rejecting the mail by setting some restrictions in main.cf. This link illustrates what happens during a typical SMTP session: http://helpdesk.islandnet.com/pep/smtp.php. The restrictions below help ward off some spam and prevent our system from becoming an open relay. These restrictions cause some mail to be rejected by Postfix right at the "front door". This will save system resources because the mail will not enter our system and therefore will not be scanned by amavisd-new (and subsequently by SpamAssassin). This is good and bad. It saves system resources, but it also doesn't let you see the rejected mail. All you will see is a log entry or two from Postfix saying essentially "Hey, a mail server named "xxxxxxxx" at IP address X.X.X.X tried to send some mail in, but it broke rule XYZ, so I rejected it". Now this could have been spam (spammers often intentionally don't follow RFCs in order to accomplish their goals) but if it was a "legitimate" mail, say from a customer whose IT Department has simply misconfigured their mail server (it happens), you could have some ruffled feathers to deal with.

If you want to allow ALL mail addressed to us to come in the front door, and therefore allow amavisd-new/SpamAssassin to handle all spam control within the system, you would need to modify a couple of the settings below. I personally find a combination of Postfix and SpamAassassin anti-spam control to be best. At the very least you need to insure you do not disable the built in default anti-relay control in Postfix (smtpd_recipient_restrictions = permit_mynetworks, reject_unauth_destination).

The configuration below is actually very conservative, allowing most email to come in the front door so amavisd-new and SpamAssassin have their shot at it. For me it (safely) rejects about 35% of the email bound for my users, so I think these settings are quite valuable. I recommend this approach to start. Adding additional restrictions will increase the likelihood of rejecting valid email from improperly configured computers. If you decide to add/remove

permissions/restrictions in the future, do so one at a time and give yourself ample time to evaluate the effect of the change. I strongly suggest you actually have a good understanding of how these restrictions work before you make changes to the entries below. Among other things, getting this stuff wrong could reject legitimate mail and/or cause us to become an open relay. Note that restrictions don't always restrict, some also permit.

If you want to gain a better understanding of these settings, good resources are http://jimsun.linxnet.com/misc/postfix-anti-UCE.txt, http://www.postfix.org/SMTPD_ACCESS_README.html, the somewhat dated (2001) http://www.mengwong.com/misc/postfix-uce-guide.txt, and this excellent book. You will note I only use a couple of the same settings as these, so this configuration is no where near as restrictive. Keep in mind that SpamAssassin and amavisd-new will come into the picture in a bit, and will provide us with much more flexible and configurable options to recognize and manipulate spam.

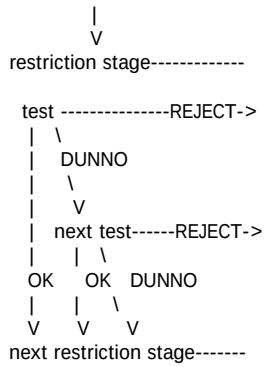
smtpd_helo_required: Make any connecting mail server do a proper smtp "handshake" and announce its name. Internet RFCs require this, so we do too.

```
postconf -e "smtpd_helo_required = yes"
```

Preface: Postfix' restriction stages are as follows, and are processed in the following order:
smtpd_client_restrictions
smtpd_helo_restrictions
smtpd_sender_restrictions
smtpd_recipient_restrictions
smtpd_data_restrictions
We are only going to place entries in the last three restriction stages. Restriction stages are processed in this order regardless of the order listed in main.cf

smtpd_sender_restrictions: This restriction stage restricts what sender addresses this system accepts in MAIL FROM: commands (the envelope sender). We will place three tests (restrictions) in this restriction stage.

A restriction stage holds a list of restrictions (tests). Typically, tests evaluate to either DUNNO, REJECT, or OK. DUNNO means "I don't know what to do, let the next test decide". REJECT simply rejects the mail. OK means no more tests are performed in this restriction stage, tests continue with the next stage (if any). reject_* type tests typically evaluate to REJECT or DUNNO. permit_* type tests typically evaluate to OK or DUNNO, and check_*_access type tests can perform a variety of actions. The illustration shows the basic logic.



1) check_sender_access See http://www.postfix.org/access.5.html. Here we ask Postfix to compare the envelope sender to entries in an /etc/postfix/sender_access database and act upon those entries if a match is found. We also define what action is taken there (OK, DUNNO, REJECT etc.) on a sender by sender basis. If the sender is not listed in the file, the test evaluates to DUNNO, and the next test is performed. I will provide examples a little later. One use of this file is to place a list of senders (email addresses, domains, network addresses etc.) from which we wish to not receive mail (blacklist them). We will also have the ability to blacklist senders in amavisd-new and SpamAssassin but it will save resources if we blacklist them here. We will also use this file to allow specific senders to bypass the next two tests in this restriction stage. If we give them an OK here, no more tests are performed in this restriction stage, the tests continue in the next restriction stage (smtpd_recipient_restrictions). Note that if we were to place this setting in the smtpd_recipient_restrictions restriction stage before the reject_unauth_destination test and we were to give someone the OK there, the reject_unauth_destination test located there would be bypassed. This would be bad because anyone we gave the OK to would then be able to use our server as an open relay. Access restrictions are evaluated in the same order we list them, and if a match is found it will influence how (whether) further restrictions are evaluated.

2) reject_non_fqdn_sender Reject when the envelope sender mail address is not in the proper format. Remember, the "envelope sender" is what the sending mail server gives in the "MAIL FROM:" line during the SMTP session, not the header "From:" line. "Joe" is not allowed to send us mail (because we can't reply to "Joe") but "Joe@example.com" is at the very least an email address. If the sender does not get rejected at this point, this test evaluates to "DUNNO".

3) reject_unknown_sender_domain Reject when the envelope sender's domain part of the mail address has no DNS "A" or "MX" record at all. This setting kicks about 35% of the mail coming in my mail server. It is common for spammers to use a bogus domain name so they don't have to deal with the backlash of rejected mail. It is also important for us not to fill up our queue with bounce notices that can never be delivered due to the fact that the sender's domain does not even exist. If the sender's domain has an "A" or "MX" record, this test will also evaluate to "DUNNO". On occasion, you will see in a report that someone you wish to receive mail from has been rejected by this setting. One possible cause of this is when legitimate senders deliberately use bogus domain names so you will not reply to them. This is where the sender access list comes in handy. You can give them an OK there, and this test will be bypassed.

```
postconf -e "smtpd_sender_restrictions = check_sender_access
hash:/etc/postfix/sender_access, reject_non_fqdn_sender, reject_unknown_sender_domain"
```

smtpd_recipient_restrictions: The access restrictions that the Postfix SMTP server applies in the context of the RCPT TO: command. This refers to the "envelope recipient" which is what the client gave in the "RCPT TO:" line during the SMTP session, not the header "To:" line. smtpd_recipient_restrictions is another restriction stage that holds a list of specific restrictions. Other restriction stages that are evaluated prior to smtpd_recipient_restrictions are smtpd_client_restrictions, smtpd_helo_restrictions and smtpd_sender_restrictions (in that order). Restrictions that would normally go in these prior restriction stages can alternately be placed in smtpd_recipient_restrictions. Therefore, some people prefer to place all the smtpd_*_restrictions that would normally go in prior restriction stages into smtpd_recipient_restrictions (in the proper order) and leave the prior stages unconfigured (empty). In our case it is safer to use smtpd_sender_restrictions and smtpd_recipient_restrictions. Let's look at those specific restrictions (tests) we place in smtpd_recipient_restrictions:

1) permit_mynetworks Allows machines listed in "mynetworks" to skip the rest of the tests in this restriction stage (permit = OK). In other words, it exits this stage and is tested in the next stage (smtpd_data_restrictions). Because permit_mynetworks is placed in front of reject_unauth_destination, this means machines in $mynetworks are allowed to relay mail to any domain. Without this, we would only be able to send mail to our own domain(s). If the IP address of the sender is *not* listed in $mynetworks, the test evaluates to "DUNNO" and continues on to the next test (reject_unauth_destination).

2) reject_unauth_destination This, along with permit_mynetworks is used for relay control. This setting, in essence, means that mail bound for any domain that we have not configured our machine to accept mail for will be rejected. In our case Postfix will use the relay_domains setting (or table) that we configured earlier to determine what domains those are. See http://www.postfix.org/postconf.5.html#reject_unauth_destination for additional details. If the domain *is* listed in relay_domains, this test evaluates to "DUNNO" and the session is allowed to go on to the next test (if any). Just like "mynetworks", this setting is extremely critical. By placing permit_mynetworks directly ahead of reject_unauth_destination, we are assured that we can send mail to domains other than ours, but we will only accept mail addressed to us from computers outside our network, thus permit_mynetworks and reject_unauth_destination work as a team.

3) reject_unauth_pipelining Rejects bulk mailers that attempt to use pipelining to speed delivery, without checking if it is supported first (non-RFC, common among spammers).

```
postconf -e "smtpd_recipient_restrictions = permit_mynetworks, reject_unauth_destination, reject_unauth_pipelining"
```

smtpd_data_restrictions: Optional access restrictions that the Postfix SMTP server applies in the context of the SMTP DATA: command. Like smtpd_recipient_restrictions, this is a restriction stage.

1) reject_unauth_pipelining I repeat this setting in smtpd_data_restrictions as it is not always effective when placed in smtpd_recipient_restrictions. I include it in smtpd_recipient_restrictions as I like to place it prior to any policy servers (discussed in a separate document I show you later). Note that there are only a couple of restrictions that make good use of smtpd_data_restrictions.

```
postconf -e "smtpd_data_restrictions = reject_unauth_pipelining"
```

Postfix content filtering control files:
http://www.postfix.org/header_checks.5.html
/etc/postfix/header_checks and **/etc/postfix/body_checks**
These files will list certain "strings" of text, and tell Postfix what to do with mail if it encounters these strings in email headers or the body of the message. Sample files are already created for us, with comments explaining what to put in them. You can edit them at your leisure. Note that these files require the use of "regular expression format". "regexp" is something you'll want to learn about in order to live in the *nix world. Get a book or research it on the Net sometime. For an example of an elaborate header_checks file from someone with an attitude, take a look at http://www.geekounet.org/filters/header_checks. But don't blindly follow this, it's only an example! Here is an example of a body_checks file: http://mercury.asuka.ne.jp/filters/body_check_regex. Once again, it's only an example! I recommend against making a large number of changes to any part of Postfix without giving yourself plenty of time to evaluate the effects. Like a turtle, go slow, live long.

Another note about header_checks. Some people wish to get rid of the "Received: from localhost" header by making a rule something like:
/^Received: from localhost \(localhost\.localdomain \[127\.0\.0\.1\]\)/ IGNORE
but get frustrated when it does not seem to work. The reason would be because this header is written after the mail has passed from amavisd-new, but in master.cf we have disabled header_checks and body_checks when mail is returned to Postfix with the override:
-o receive_override_options=no_header_body_checks,no_unknown_recipient_checks
so you would have to remove "no_header_body_checks" if you want to work on headers written after the message has left amavisd-new. You can remove the other amavisd-new related header by setting "$insert_received_line = 0;"

in amavisd.conf. The downside would be all your costly header_checks and body_checks will be performed again.

header_checks:

Optional *(only use if you intend on using header_checks):*
postconf -e "header_checks = pcre:/etc/postfix/header_checks"

Note that we do not have to 'postmap' pcre type tables. They remain plain text and Postfix uses them "as is".

body_checks:

Optional *(only use if you intend on using body_checks):*
postconf -e "body_checks = pcre:/etc/postfix/body_checks"

Keep in mind that you could also configure mime_header_checks and nested_header_checks along with header_checks and body_checks. Note that enabling content filtering in Postfix (we are using header_checks and body_checks) can have a **significant impact on performance** in Postfix. If later you find that you have not had occasion to place any entries into these files, you should comment them out in main.cf.

content_filter: Here's where we tell Postfix to use amavisd-new. http://www.postfix.org/FILTER_README.html

```
postconf -e "content_filter = smtp-amavis:[127.0.0.1]:10024"
```

/etc/postfix/sender_access http://www.postfix.org/access.5.html
We referenced this file in smtpd_sender_restrictions. We use this file to check the sender right at the front door. In this file, we'll list certain senders/domains/IPaddress ranges for special handling. Below are bogus examples, create your own as you see fit. Please read /etc/postfix/sender_access for more information. Although you could use this file for various purposes, considering the way we have set this up in smtpd_sender_restrictions, I suggest using it to either blacklist senders, or allow certain senders to bypass the remaining tests in smtpd_sender_restrictions.

```
vi /etc/postfix/sender_access
```

```
#Example sender access map file
makeabuck@mlm.tld 550 No MLM thanks
allspam.tld 550 Spam is not accepted here
badguy.net REJECT
justaspamminfool@allspamallthetime.com REJECT
newsletter-fake-domain.com OK
my-bogus-test-domain.com OK
```

Since this is a hash table, you need to postmap it as usual:

```
postmap /etc/postfix/sender_access
```

Let's take a look at our changes:

```
vi /etc/postfix/main.cf

Check the contents of the file for errors and repair if needed.
You will probably want to edit relay_domains and mynetworks.
```

```
postfix start
```

Check that Postfix responds:

```
telnet 127.0.0.1 25

You should see:
220 sfa.example.com ESMTP Postfix (Debian/GNU)

hit [enter] a few times; then type:
quit to exit
```

If it does not reply in this manner, open another terminal window and stop Postfix (postfix stop). Make sure you ran newaliases and all the postmap commands above. Check all the settings in main.cf and master.cf. Begin debugging by checking the mail.log for a fatal error: grep fatal /var/log/mail.log There is a nice paper on troubleshooting Postfix at http://www.postfix-book.com/debugging.html but keep in mind our system is not ready to relay mail at this point (it will end up in the queue because we have not yet configured amavisd-new).

If you make changes to master.cf or main.cf or to data tables, most of the time it is required that you reload Postfix. One time it may not be required is when you make changes to a map file, and then postmap it. Postfix should recognize the file has changed and reload it. Conversely, if you change a parameter like inet_interfaces, 'postfix reload' won't cut it. You would have to stop and start Postfix:

```
postfix reload
```

Now that we have a basic Postfix configuration, http://www.postfix.org/STANDARD_CONFIGURATION_README.html#firewall and http://www.postfix.org/BASIC_CONFIGURATION_README.html are good places to gain a better understanding of some of the settings we used and at this point these READMEs will make more sense. If using this box on an internal network you may choose to omit some of the Postfix configuration parameters that may only apply to a system that is on the Internet. Once our

system is fully configured you may wish to use the examples in these READMEs to enhance the settings we have made thus far to better suit your needs. Don't send mail through the system just yet, we need to configure amavisd-new first.

By default, our Postfix runs chrooted. If you don't know what that means, it will not matter much at this point. I leave it to you to research what 'chroot' means at some later time. Some of the system files that Postfix needs to run properly were copied to Postfix's chroot jail (/var/spool/postfix) during installation. On occasion, the original files will get modified, and Postfix will complain that the copy it has is not the same as the original. When this happens, you can manually copy the file(s) postfix has complained about to the chroot jail, or we can simply run a script that is supplied with the Postfix source code (called LINUX2) that will once again copy all the files that Postfix needs to where it needs them.

Quick lesson: To change to the Postfix directory shown below, you could type cd p then hit the [tab] key and the bash shell will fill the remainder in with the first unambiguous item. If there is more than one item, you can hit the [tab] key twice and it will list all the available items.
For example, the second line below could be entered as cd p[tab] e[tab] c[tab][enter]

```
cd /usr/local/src

This next line may be edited for a different version of postfix:
cd postfix-2.3.8/examples/chroot-setup

postfix stop
postfix start
chmod +x LINUX2
cp LINUX2 /usr/bin
LINUX2
cd

You can check if Postfix is happy:
postfix check
```

This makes the LINUX2 script executable, copies it to a directory in our path, Then executes it.

****

Configuring amavisd-new:

Index

****

First, make a backup of the amavisd-new config file:

```
cp /etc/amavis/amavisd.conf /etc/amavis/amavisd.conf-2.5.3-original
```

General note. The settings used here are what might be used at small business. A business where the users trust the administrator(s) to make choices for them. In more complex systems, individuals will want their own personal settings. I do not cover that scenario in this document. Suffice it to say that if you need to provide individual settings for a large number of users, the most efficient way to do this would be to use LDAP or SQL to store those settings (dynamic tables). You would need to read the amavisd-new documentation and have some knowledge of LDAP or SQL. You can however provide individual settings to a small number of users by using static tables directly in amavisd.conf, or in plain text files read into amavisd-new via the read_hash() function. There are examples in amavisd.conf. If you use static tables, it is necessary to stop and start amavisd-new after each change.

Next, we'll edit this file, but first let me just mention that this file is a very important part of configuration control in your spam system. There are many settings in here. I won't cover all of them by any means. This file is an abbreviated version of one of those heavily commented config files that some hate, others love. I recommend you return and spend some time reading /etc/amavis/amavisd.conf-sample after this system is all set up, to learn more about just what you can do with amavisd-new. We are going to start at the top of the file and work our way down. Remember, in vi you can search for a string by hitting [Esc], a slash, the text to search for, and optionally a lower case 'n' to find the next occurrence. Once you find what you want, don't forget to use 'i' to start editing. I personally find it *much* easier to edit this particular file using the WinSCP editor (select the file, hit F4 to edit).

Either:
```
vi /etc/amavis/amavisd.conf
```

or use the WinSCP editor to edit this file.

Insure the amavis user and group are set like so:
```
$daemon_user = 'amavis';
$daemon_group = 'amavis';
```

Locate the line that begins with $mydomain:
```
$mydomain = 'example.com';
```

and change to reflect the actual name of this system's domain.
Next, ***uncomment*** # $myhostname *and change it to our host's FQDN:*
```
$myhostname = 'sfa.example.com';
```

Locate this next line:

```
@local_domains_maps = ( [".$mydomain"] );
```

Comment out that line, then add a new one just like it with all your domains listed in it. Like this:

```
@local_domains_maps = ( [ ".$mydomain", '.example2.com', '.example3.com' ] );
```

Or like this:

```
@local_domains_maps = ( [qw( .example.com .example2.com .example3.com )] );
```

The periods in front of the domains are wildcards for subdomains (or hosts). Don't forget the semi-colon at the end. Note that in Perl, single quotes and double quotes work differently from each other.

We need this in order for all our domains to see the SpamAssassin scores in the header of each message. If you have a large number of domains to list, read the instructions in amavisd.conf-sample on creating a Perl hash lookup table. The reason we need our domains listed here is explained in http://www.ijs.si/software/amavisd/#faq-spam

Let's move on to SpamAssassin settings. Amavisd-new calls SpamAssasssin like a sub-routine. SpamAssassin picks up all settings in local.cf, but modified mail as prepared by SpamAssassin is not used, amavisd-new does its own modifications to the original mail. These settings are part of what determines what amavisd-new does with the mail and may include rewriting of the Subject: line, addition of the X-Spam-* lines in the headers and possibly directing amavisd-new to take other action with the mail.

Change

```
$sa_tag_level_deflt = 2.0;
```

to

```
$sa_tag_level_deflt = undef;
```

This will insure all mail addressed to domains in @local_domains will get a spam score in the header, spam or not.

Change

```
$sa_tag2_level_deflt = 6.31;
```

to

```
$sa_tag2_level_deflt = 5.0;
```

This low number is assuming your email clients do not automatically discard spam. Set this to 8.0 or higher if they do.
The Subject line will be prepended with "Spam> " for any mail that scores above $sa_tag2_level_deflt and is passed to a recipient.

Change

```
$sa_kill_level_deflt = 6.31;
```

to

```
$sa_kill_level_deflt = 8.0;
```

On our system, this will trigger the spam to be quarantined if it scores 8.0 or higher.

If you only want spam tagged and sent to the recipients (not quarantined at all), set this to 9999
(this would be for ISPs and large companies that configure MUAs or LDAs to further process spam).

To better understand these and other settings, see http://www200.pair.com/mecham/spam/amavisd-settings.html. I believe these are good starting points. The comments provided in the file give some clues what these settings mean. See amavisd.conf-sample for more clues. With a kill level of 8.0 you can expect a fair amount of spam to slip through this system and into your users' mailboxes initially, but it will get better as time goes on. The Bayes filter included with SpamAssassin will not kick in until it has processed at least 200 pieces of "ham" or "non-spam". The length of time it takes to do this completely depends on how many email messages this system processes per day. Mine took about a week to find 200 legitimate email messages to work with. It does it all automatically but later I will explain how to help it along. Once the Bayes filter kicks in, the kill level of 8.0 will be just about right. It will be on the conservative side. Wait a few weeks and keep an eye on the spam. If you choose a kill_level score lower than the one I suggest, the more effort there is on your part to find false positives because there will be more of them in the spambin. Speaking of the Bayes database: the amavisd-new package maintainers were kind enough to write a couple of scripts that keep the database up to date and clean. They run as cron jobs and consist of /etc/cron.d/amavisd-new and /etc/cron.daily/amavisd-new. Note that when these cron jobs are running, the Bayes files will at times be inaccessible. If you happen to see in a log file that SpamAssassin has timed out, this could be one possible reason.

Now find:
```
$virus_admin = "postmaster\@$mydomain";
```

The default is fine, but you may wish to change it. Now add a line for banned file notifications:
```
$banned_admin = "postmaster\@$mydomain";
```

When we do send a virus or banned file notification, this is whom the notification will go to. Banned attachments are treated much like viruses, and right out of the box, amavisd-new will want to quarantine emails that contain banned attachments. Initially, the only banned attachments it is configured to quarantine are ones with double extensions, i.e. "filename.gif.pif" and a few major risks like ".exe".

Next, locate this section:
```
$mailfrom_notify_admin     = "virusalert\@$mydomain";
$mailfrom_notify_recip     = "virusalert\@$mydomain";
$mailfrom_notify_spamadmin = "spam.police\@$mydomain";
```
and change it to:
```
$mailfrom_notify_admin     = "postmaster\@$mydomain";
```

```
$mailfrom_notify_recip     = "postmaster\@$mydomain";
$mailfrom_notify_spamadmin = "postmaster\@$mydomain";
```
Just below this, remove the '#' to uncomment this line:
```
# $hdrfrom_notify_sender = "amavisd-new <postmaster\@$mydomain>";
```

For a discussion of these settings, see
http://marc.theaimsgroup.com/?l=amavis-user&m=111938103904411

Next, locate # $recipient_delimiter*. You set* recipient_delimiter *in Postfix and the setting in this file needs to be set similarly.* ***This is typically set to one of these three options:***

```
$recipient_delimiter = '';  # No recipient delimiter (this is the default)
```
or
```
$recipient_delimiter = '+';
```
or
```
$recipient_delimiter = '-';
```

Change:
```
#$sa_spam_subject_tag = '***SPAM*** ';
```
to
```
$sa_spam_subject_tag = 'Spam> ';
```
The longer version simply takes up too much real estate on the subject line.

The next line I would like to change begins with:
```
$final_spam_destiny = D_BOUNCE;
```

Change this to:
```
$final_spam_destiny = D_DISCARD;
```

This does not actually discard mail; the spam still ends up in our quarantine. The difference between D_BOUNCE and D_DISCARD is D_DISCARD will not send a bounce notice to the spammer for a piece of mail that is marked as spam. Unlike spam, we may want to notify the sender of a banned attachment that their mail was not delivered. We usually don't want to notify senders of viruses (because the sender is probably spoofed). This is a personal choice; I don't want my queue to be full of bounce notices that can't be delivered to spammers. From their perspective, the spam was successfully sent. If we set our $sa_kill_level_deflt high enough (12 at the very least), D_DISCARD CAN be used to actually discard mail. To do so, you could either set: $spam_quarantine_to = undef; and the email with a SpamAssassin score of 12 or higher would evaporate into thin air, or you could alternately leave $sa_kill_level_deflt at something like 8 and set $sa_quarantine_cutoff_level = 12; (preferably higher - I use 14). **But don't do this; at least not yet.** Setting $spam_quarantine_to = undef; essentially means "we don't have a spam quarantine area so I guess I have to just dump the mail". This is something you probably should not implement at first; at least not if you enjoy being employed. If you decide to do this, I suggest you wait a month or so to get a

feel for the system (and let the Bayes database initialize). You can eliminate about 90% of the email that ends up in the "spambin" by using this suggestion.

Next, locate the line that looks like this:
$virus_quarantine_to = 'virus-quarantine';

I suggest you do one of three things here. You can leave this alone, and viruses will be quarantined on the spamfilter box (to /var/lib/amavis/virusmails), or you can set this to:
$virus_quarantine_to = undef;

and provided we configure a virus scanner, all caught viruses will disappear, or you can send caught viruses to a regular mailbox:
$virus_quarantine_to = "virii\@$mydomain";

You can choose to use "spambin" for the quarantine area for all spam, viruses and email with banned attachments, but I recommend you create separate mailboxes for each.

If you choose to store quarantined viruses (or any other type of quarantined mail) on your spamfilter box, keep in mind that you will have to eventually delete them, and on rare occasions you may need to forward a sample to an email address. These files will be stored in the /var/lib/amavis/virusmails directory. You can use a program supplied with the amavisd-new source code called 'amavisd-release' to send items in the quarantine to an email address if needed. Read the amavisd-release file itself for more details. You will have to make changes to amavisd.conf and make this program executable before you can use it. If you send viruses to a regular mailbox, make sure the desktop virus protection is up to date on the client that retrieves these messages, and remember - don't open attachments with viruses in them!

Next, locate the $banned_quarantine_to line and configure these three lines in this manner:
$banned_quarantine_to = "banned\@$mydomain";
$bad_header_quarantine_to = "banned\@$mydomain";
$spam_quarantine_to = "spambin\@$mydomain";

Make sure you have mailboxes for these two addresses on a destination server. This is where you will review quarantined email, and if ham is found, will forward the ham to the proper recipient.

Locate this block inside the $banned_filename_re = new_RE section:
```
  qr'.\.(exe|vbs|pif|scr|cpl)$'i,              # banned extension - basic
# qr'.\.(exe|vbs|pif|scr|cpl|bat|cmd|com)$'i, # banned extension - basic+cmd
# qr'.\.(ade|adp|app|bas|bat|chm|cmd|com|cpl|crt|emf|exe|fxp|grp|hlp|hta|
#        inf|ins|isp|js|jse|lnk|mda|mdb|mde|mdw|mdt|mdz|msc|msi|msp|mst|
#        ops|pcd|pif|prg|reg|scr|sct|shb|shs|vb|vbe|vbs|
#        wmf|wsc|wsf|wsh)$'ix,  # banned ext - long
```

If you would like to *greatly extend the types of attachments amavisd-new bans, you might want to comment out, and uncomment some lines like so:*

```
# qr'.\.(exe|vbs|pif|scr|cpl)$'i,              # banned extension - basic
# qr'.\.(exe|vbs|pif|scr|cpl|bat|cmd|com)$'i, # banned extension - basic+cmd
  qr'.\.(ade|adp|app|bas|bat|chm|cmd|com|cpl|crt|emf|exe|fxp|grp|hlp|hta|
         inf|ins|isp|js|jse|lnk|mda|mdb|mde|mdw|mdt|mdz|msc|msi|msp|mst|
         ops|pcd|pif|prg|reg|scr|sct|shb|shs|vb|vbe|vbs|
         wmf|wsc|wsf|wsh)$'ix,  # banned ext - long
```

Just edit out the attachment types you would like to receive or edit in any others you would like to ban. Each vertical bar means "or". I would add bin|drv|mht|ocx|ovl|ani|cur|ico| *If you receive mail from people using Outlook in Rich Text mode, make sure you don't block* |tnef
When an email that has one of these attachments comes into the system, the entire message gets placed in quarantine and the sender and postmaster get notified. Using this feature of amavisd-new goes a very long way toward preventing email borne viruses from entering your network. This method should not be your only line of defense however, just another tool in your arsenal.

Next, locate the # ENVELOPE SENDER SOFT-WHITELISTING / SOFT-BLACKLISTING *section.*

Spend a moment to read through the comments at the beginning of this section simply to make you aware of what is there, and what it is used for. Negative scores for sender addresses mean "make their mail appear less spammy" and positive scores mean "make their mail look more spammy".

Save the file with [Esc]:wq *and exit vi. Then make a backup:*

```
cp /etc/amavis/amavisd.conf /etc/amavis/amavisd.conf-13apr07
```

I have a habit of using the date for my backups.

If you have a powerful machine with sufficient RAM (512MB or more), at some time in the future you may want to experiment with increasing the number child processes that amavisd-new spawns. More child processes means that amavisd-new can process more mail in a given amount of time (up to a practical limit). See http://www.ijs.si/software/amavisd/amavisd-new-magdeburg-20050519.pdf. If you configure your system to use more instances of amavisd-new, allocate at least 40MB for each additional instance. It you wanted to double the number of child processes from 2 to 4, you would edit amavisd.conf and change:

$max_servers = **2**;

to

$max_servers = **4**;

Then edit master.cf and change:

smtp-amavis unix - - - - **2** smtp

to

smtp-amavis unix - - - - **4** smtp

Amavisd-new will be the biggest bottleneck in the system. On a busy server you will probably want 2GB RAM so you can accommodate somewhere around 15 $max_servers.

Our system will log to both /var/log/mail.log and /var/log/mail.info (with identical information). We don't really need the log duplicated to /var/log/mail.info:

Optionally *disable logging to /var/log/mail.info:*
vi /etc/syslog.conf

and comment out the mail.info line, like so:
#mail.info -/var/log/mail.info

We can wait for the next reboot for this to take effect.

There are files that act as templates (that can be modified) that are used as the basis for the text used in email notifications. Normally these templates are stored inside the amavisd-new file itself, but it is possible to copy this text into external files, and then place a setting in amavisd.conf that uses these external files. The typical Debian installation is configured to use these external files but I have commented out the setting in our custom amavisd.conf. If you don't plan on customizing any of the templates, you can simply leave the external files disabled:

Optional:
If you would like to enable the external template files (located in /etc/amavis/en_US/) so you may customize them in the future (English only):
vi /etc/amavis/amavisd.conf

and uncomment the line:
read_l10n_templates('en_US', '/etc/amavis');

Let's give amavisd-new a little test run (amavisd-new must be stopped before running it in debug mode):

```
amavisd-new stop
amavisd-new debug
```

After just a few moments, if you have something misconfigured; amavisd-new will tell you. At this point don't worry about the razor2 error if you get it. If you have an error in /etc/amavis/amavisd.conf, it will give you a line number and a brief explanation. Fix anything wrong. This will mean reading closely any error messages, and possibly reading the files in /usr/local/src/amavisd-new-2.5.3. There are friends at the AMaViS mailing lists waiting to help. https://lists.sourceforge.net/lists/listinfo/amavis-user. If everything is OK, and it should be, you will see a lot of lines that look like text log entries (exactly

what debug mode does - logs to the screen), and near the bottom you will see "Parent ready for children". You can ignore the lines above this that talk about things like "INFO: no optional modules:" and "No decoder for" and "Cache not available", these are not errors, they are simply informational. FYI, each time amavisd-new starts up normally on your box, all these lines will be recorded in the system log files.

Use the [Ctrl]+c key combination to exit (kill) amavisd-new debug.

If you are interested (and I realize this link may unnecessarily give you information overload at this point), there is one issue with quarantining mail in the manner I suggest above. The author of amavisd-new (Mark Martinec) suggests using Plus Addressing as an alternative.
http://www200.pair.com/mecham/spam/plusquarantine.html

At this point, if this server is going to be on the Internet (as opposed to residing on an internal network) you can **greatly** decrease the load on the server by increasing the amount of mail that is rejected at the front door. You may choose to continue on without reading it right now, and consider using this document a little later, as it is optional (but **highly** recommended).

Additional anti-UCE settings for our Debian Anti-Spam Anti-Virus Gateway Email Server

****

Pyzor, Razor and SpamAssassin configuration:

Index
****
Pyzor configuration.

Here we supply the IP address of the Pyzor server to Pyzor (for both the 'root' and 'amavis' users). This will create a .pyzor directory in both user's home directories, and place the server's IP address in a 'servers' file therein:
pyzor discover
su amavis -c 'pyzor discover'

Test the pyzor server for a response:
pyzor ping
su amavis -c 'pyzor ping'

More than likely, you will get TimeoutError:. Regardless, I recommend using a server other than the one 'pyzor discover' gives us:
echo "82.94.255.100:24441" > /var/lib/amavis/.pyzor/servers
echo "82.94.255.100:24441" > /root/.pyzor/servers

Then test again:
su amavis -c 'pyzor ping'

If in the future the IP address of the server changes, you will need to run the two 'pyzor discover' commands again. I suggest you subscribe to http://lists.sourceforge.net/lists/listinfo/pyzor-announce.

These commands make the Pyzor files world readable and sets the IP address of a Pyzor server. You can find the address of the current Pyzor server here.

You will need a sample spam to feed to spamassassin:

```
cd /root
wget http://www200.pair.com/mecham/spam/sample-spam.txt
```

This next section gets Razor2 up and running and copies its files where both root and amavis expect to find them. If you were to run `amavisd-new -d config debug-sa` you would notice that amavis expects to find programs and configuration files in certain places. If you were to run `spamassassin -D config </root/sample-spam.txt` as root you would notice that root expects to find the same things, but it expects to find some of them somewhere other than where the amavis user expects. SpamAssassin is designed to enable each user to have their own settings and data. This section will make both users happy, and the reason we want to do this is because if we are debugging SpamAssassin or Razor or Pyzor or DCC, we want to be able to do so with `spamassassin -D </root/sample-spam.txt` rather than `amavisd-new debug-sa` because we don't want to shut amavisd-new down every time we need to debug one of those programs. A better way to debug SpamAssassin is to run the program as the amavis user like so: `su amavis -c 'spamassassin -D </root/sample-spam.txt'`

```
spamassassin -D </root/sample-spam.txt
```

Just to make sure SpamAssassin is alive. Running this may create some files also. We run this the first time as root. If Pyzor is working, you will see "Pyzor: got response:" Pyzor queries a Pyzor server in much the same way your computer queries a DNS server. The only practical difference is the port number that is used. DNS uses port udp 53 and Pyzor uses port udp 24441. If Pyzor is not working, you most likely have a firewall in front of this machine that is blocking the response. It is also possible the Pyzor server is not working. Refer back to the Pyzor configuration section and try the other server listed there. If you need to open a hole in a firewall or router, configure it just as you would DNS port 53, but substitute port 24441. While you are at it, do the same for port 6277. DCC will use port udp 6277. Notice in our firewall settings above that these 2 ports are configured just like DNS. Razor will need the ability to ping the outside world and outgoing tcp port 2703 must not be blocked. As far as DCC, Razor and Pyzor go, try them before you start messing with your Internet firewall. I have my spamfilter behind a screening router, a hardware firewall, and software NAT box firewall and none of them required reconfiguration for these programs to work.

http://www.dcc-servers.net/dcc/firewall.html gives an example of a Cisco router access list entry. Keep in mind that you will likely not run a DCC server, only the DCC client. Also http://flakshack.com/anti-spam/wiki/index.php?page=Provide+firewall+access will give you some idea of what we are after.

We are going to make a backup of the original installation files then copy some of them to where the amavis user expects to find them.

```
cp -ir /root/.spamassassin /root/.spamassassin-backup

Please answer 'n' to "overwrite?":
cp -ir /root/.spamassassin /var/lib/amavis
```

We are going to make a symbolic link, so we only have to deal with one user_prefs file for both the root and amavis users.

```
rm /root/.spamassassin/user_prefs
ln -s /var/lib/amavis/.spamassassin/user_prefs /root/.spamassassin/user_prefs
```

If you run `sa-learn --force-expire` or `spamassassin --lint -D` or other spamassassin commands from the root account, SpamAssassin may change the owner of the Bayes files to 'root'. If it does, amavis will no longer be able to read those files. You would need to run `chown -R amavis:amavis /var/lib/amavis` to regain ownership. In general, if you do any spamassassin maintenance from the command prompt as root, the best thing to do is run `chown -R amavis:amavis /var/lib/amavis` afterwards; just to make sure. You can avoid these problems by remembering to run spamassassin commands as the amavis user. For example `su amavis -c 'sa-learn --sync --force-expire'`

This next section configures Razor; sets the elusive "razorhome" and makes both root and amavis happy in their attempts to figure out "where in the heck are the Razor2 configuration files"?

```
cd
rm /etc/razor/razor-agent.conf
razor-admin -create
razor-admin -create
razor-admin -register

If you get an error, you may need to run the 'razor-admin register' command more than once.
Don't worry about it if /etc/razor/razor-agent.conf does not exist.
Now edit root's razor configuration file:
vi /root/.razor/razor-agent.conf

and change the line:
debuglevel = 3
```

to:
debuglevel = 0

Obviously -zero- not -oh-; Save and exit the file.
OK, now copy root's .razor directory and files to the amavis user's home directory:
cp -r /root/.razor /var/lib/amavis
chown -R amavis:amavis /var/lib/amavis

We copied root's Razor stuff to where the amavis user expects to find it, so now we have a razorhome for both root and amavis. We changed debuglevel to 0 to prevent the log from filling up our entire hard disk.

Now we will modify SpamAssassin's main configuration file:

vi /etc/spamassassin/local.cf

And insert the lines:

bayes_path /var/lib/amavis/.spamassassin/bayes
auto_whitelist_path /var/lib/amavis/.spamassassin/auto-whitelist
whitelist_from spambin@example.com
lock_method flock
bayes_auto_learn_threshold_nonspam -0.5
razor_timeout 8

This insures both the root and amavis users use the same files and do not have to guess where they are, and whitelists our spambin. lock_method flock *is used when the Bayes data resides on the local hard disk and is non NFS.*

Optional:
Since there is a script that runs each day to --force-expire old Bayes tokens "/etc/cron.daily/amavisd-new" (make sure there is if you use this setting!), we can set:
bayes_auto_expire 0

Optional:
Some people believe auto-whitelist is more of a liability than an asset:
use_auto_whitelist 0

Possibly optional, possibly not:
*Depending on your setup, it might be necessary to explicitly set internal_networks and trusted_networks. The trust path tells spamassassin which clients are not trusted. If you are using SpamAssassin version 3.2 or newer, do not include the 127/8 networks shown below. They are automatically included. See http://wiki.apache.org/spamassassin/TrustPath and [this thread](): *

```
# explicitly set our internal_networks (might be the same or similar to mynetworks)
clear_internal_networks
internal_networks 127/8
internal_networks 222.222.222.222/24
internal_networks 10.10.10.10/24
# add the same to trusted_networks, and possibly other computers/networks whose mail we
trust
clear_trusted_networks
trusted_networks 127/8
trusted_networks 222.222.222.222/24
trusted_networks 10.10.10.10/24
```

Save and exit the file. Please read http://spamassassin.apache.org/full/3.1.x/dist/doc/Mail_SpamAssassin_Conf.html at some point keeping in mind that some settings will have no effect when used with amavisd-new. Amavisd-new does not allow SpamAssassin to modify messages. You may also wish to set dns_available yes .

With SpamAssassin version 3.1 or newer, additional configuration is needed:
vi /etc/spamassassin/v310.pre

To enable the ability to use DCC, uncomment the line:
#loadplugin Mail::SpamAssassin::Plugin::DCC

Since we are using SpamAssassin 3.1.1 or greater we can use the new sa-update feature.
First import the gpg key:
cd /etc/spamassassin
wget http://spamassassin.apache.org/released/GPG-SIGNING-KEY
gpg --import GPG-SIGNING-KEY

Then simply run:
sa-update

You should find the new rules in /var/lib/spamassassin/<SA version>.
*You should also run **amavisd-new -d config debug-sa** (run **amavisd-new stop** first) and verify SpamAssassin is locating all of its rule sets in /var/lib/spamassassin/<SA version>. You should also run:*
su amavis -c 'spamassassin --lint'

*after an update, and you should be aware you must reload amavisd-new after the update in order for the new rules to be used. It is important that sa-update completes without error. **Optionally** run sa-update every day (you should probably change the time this runs):*
crontab -e

and insert (on the first available blank line):
5 5 * * * /usr/bin/sa-update && /usr/bin/spamassassin --lint && /etc/init.d/amavis restart

While we are at it, have razor discover its servers once a week:
2 2 * * 2 su amavis -c '/usr/bin/razor-admin -discover'

Save and exit the file. Note: during the time amavisd-new is restarting, mail cannot be delivered to it. Postfix will complain "connect to localhost[127.0.0.1]: Connection refused". Postfix will defer this mail (for about 15 minutes). To speed things up, an impatient person may run 'postfix flush' to flush the deferred queue, but I would not.

```
chown -R amavis:amavis /var/lib/amavis
chmod -R 750 /var/lib/amavis
su amavis -c 'sa-learn --sync'
```

Since we've moved stuff into one of the amavis directories, we used 'chown' to refresh amavis's ownership of the entire tree. This is important. We need to do this any time we place files in the amavis directory that amavis does not already own. You should not have to run 'chmod' again. We also ran sa-learn to create the initial Bayes data.

If you would like to see if SpamAssassin and Razor are working, run spamassassin with the debug (-D) option (run this a couple times):

```
spamassassin -D </root/sample-spam.txt
```

Notice that DCC is not working. That's because we have not installed it yet.

If you debug spamassassin as user root I recommend you refresh file ownership every time:

```
chown -R amavis:amavis /var/lib/amavis
```

A note on Bayes: to better train the Bayes database we need to give it some examples of ham. The problem is, sa-learn --ham /path/to/hamfiles works on files that reside on the spamfilter. We don't keep mail on the spamfilter however. This problem can be solved by using your email client (MUA) to individually save messages in .EML format. Look for this option among the menu choices or 'Save As'. Edit a sample .EML file to insure the file looks like plain text. If it is full of garbage characters, then it cannot be used. Use WinSCP to copy your ham collection to an empty folder on your spamfilter. Then run sa-learn. Pick messages that have some substance to them so the database has something to work with. Use this especially when you receive a message that is tagged as spam, but is in fact ham. Autolearning (bayes_auto_learn) is turned on by default so the system will have no problem gathering plenty of spam. I suggest you only feed it additional examples of ham to begin with. In the future you will want to feed it samples of low scoring spam.

SpamAssassin has the ability to store its Bayes data in a MySQL database. If you are familiar with MySQL (or not) and you wish to improve performance, you should consider using it. Here is my HOWTO to get you going: http://www200.pair.com/mecham/spam/debian-spamassassin-sql.html

****

Installing DCC:

Index
****
DCC is available from the Debian archives, but we will get it from the author and compile it from the source code. Installing it from source is a good exercise and we have better control over how it installs. Installing from source allows us to customize the installation for use with amavisd-new. Note that as of version 1.3.0 of DCC http://www.commtouch.com/ has exclusive marketing rights for DCC. If you resell anti-spam solutions that use DCC and you do not provide your DCC data to the public, you will need to pay for DCC. Please read the license.

```
cd /usr/local/src
wget http://www.dcc-servers.net/dcc/source/dcc-dccproc.tar.Z
tar xzvf dcc-dccproc.tar.Z

Change to the dcc subdirectory by using the [tab] key command completion
shortcut as shown, then ./configure:
cd dcc-dccproc- [tab][enter]

./configure --with-uid=amavis && make && make install

The double ampersands let you run those 3 commands on one line. You will
see 'done' if all goes well.
Place a link to cron-dccd in our path:
cd
ln -s /var/dcc/libexec/cron-dccd /usr/bin/cron-dccd

Test our installation with:
cdcc info

We should get 'requests ok' from the servers (but 'not answering' from
127.0.0.1 is expected).
The instructions say to run cron-dccd each day to clean things up, so we will
do that.
crontab -e

and insert (on the next available blank line):
43 11 * * * /usr/bin/cron-dccd
```

Make sure you have a carriage return at the end of the line, then save the file.

The cron-dccd clean-up program will run every day at 11:43am. We made the symbolic link to cron-dccd so it would be in our executable path, per DCC's instructions. For newbies who would like to understand cron: http://www.unixgeeks.org/security/newbie/unix/cron-1.html

Now test it with SpamAssassin:

```
su amavis -c 'spamassassin -D </root/sample-spam.txt'
```

You should see "dcc: got response" if all is well. If all is not well, it's probably the firewall issue we discussed above. If you just can't get Razor or DCC or Pyzor working properly, you should disable them in SpamAssassin's configuration file to prevent SpamAssassin from wasting its time trying to make them work. SpamAssassin will use all three by default. You have to tell SpamAssassin not to use them.
If one of them is broken, you would edit /etc/spamassassin/local.cf and comment out the offending party as needed:
use_razor2 0
use_pyzor 0
use_dcc 0
I hope you got them all working because they really are valuable resources in the fight against spam. I have noticed over the last year that fewer and fewer people are submitting spam to the pyzor servers with the result being fewer and fewer messages are tagged by pyzor.

Now give it the full blown amavisd-new and SpamAssassin test:

```
amavisd-new stop
amavisd-new debug-sa
```

What can I tell you? It works fine on my computer (and every one of these I have built). You should see nothing but happiness. The "fixed relative path:" messages are normal and "dccifd is not available:" is also normal and expected at this point.

Please stop amavisd-new at this time with [Ctrl]+c.

Normally SpamAssassin runs the 'dccproc' program for each message that it processes. If you have 30MB RAM to spare, you may benefit from running the daemonized version of the DCC client 'dccifd'. I personally recommend you have a system with at least 384MB RAM to do this. Note that if you were unable to get DCC working before, enabling dccifd will not solve the problem and should not be attempted.

Optional *to enable dccifd:*

vi /etc/spamassassin/local.cf

and insert:
dcc_home /var/dcc

save and exit, then:
vi /var/dcc/dcc_conf

and change
DCCIFD_ENABLE=off
to:
DCCIFD_ENABLE=on

Then change:
DBCLEAN_LOGDAYS=14
to:
DBCLEAN_LOGDAYS=1
save and exit.

If you choose to allow logging, cron-dccd should delete old log files when it runs. Keep your eye on the files that accumulate in the /var/dcc/log directory. It's your choice, but I personally don't want to monitor the DCC logs, so I turn off logging altogether by deleting the log directory and commenting out the logdir entry in dcc_conf:
rm -r /var/dcc/log
vi /var/dcc/dcc_conf
and comment out:
DCCIFD_LOGDIR="$DCCM_LOGDIR"

We will use a supplied script (rcDCC) to automatically start dccifd when we boot up:
cp /var/dcc/libexec/rcDCC /etc/init.d/adcc
update-rc.d adcc defaults

Update file ownership:
chown -R amavis:amavis /var/dcc

Because we enabled dccifd in dcc_conf, we can start up dccifd by running the script:
/etc/init.d/adcc start

If you deleted the log directory, you can expect an error message: "log thresholds set with -t but no -l directory".
Now test that SpamAssassin finds dccifd:
su amavis -c 'spamassassin -D </root/sample-spam.txt'

You should see: dbg: dcc: dccifd got response:

****

Local DNS cache:

Index
****
Having a local DNS cache is highly recommended provided you currently do not have a DNS server or other server that provides a DNS caching function on your network and you have sufficient available memory. I have found that using a broadband router (or similar) as a primary name server does not provide the same functionality a true DNS server provides. Name server settings are located in /etc/resolv.conf. If you choose not to use a local DNS cache then at least use a real DNS server as your primary. I have seen SpamAssassin time out on RBL lookups if it cannot perform DNS queries quickly enough. This can have a big effect on the final spam score. I will not pretend that I understand the intricacies of the bind9 program that we will install. We will configure bind9 to be a caching only name server with the option of forwarding requests to another server. You may also consider providing the service we install on this machine to other machines on your network. Note that you will need to modify our local firewall (iptables) if you choose to provide this service to other machines:

```
iptables -A FIREWALL -p udp -m udp --dport 53 -j ACCEPT
iptables -A FIREWALL -p tcp -m tcp --dport 53 -j ACCEPT
```

(or for enhanced security, only allow access from clients on our network):

```
iptables -A FIREWALL -p udp -m udp -s 222.222.222.222/24 --dport 53 -j ACCEPT
iptables -A FIREWALL -p tcp -m tcp -s 222.222.222.222/24 --dport 53 -j ACCEPT
```

See the "Create Firewall Rules:" section above.

The bind9 manual is available at http://www.bind9.net/manual/bind/9.3.4/Bv9ARM.html and I wish to thank this document for much of the information listed here: http://www.falkotimme.com/howtos/perfect_setup_debian_sarge/index.php

Now we install bind9:

We will install version 9.3.4 of bind9:

```
apt-get install bind9
```

For security reasons we want to run BIND chrooted so we will perform the following steps:

```
/etc/init.d/bind9 stop
```

Edit the file /etc/default/bind9 so that the daemon will run as the unprivileged user 'bind', chrooted to /var/lib/named:

```
vi /etc/default/bind9
```

Modify the line: OPTIONS="-u bind" so that it reads:

```
OPTIONS="-u bind -t /var/lib/named"
```

Create the necessary directories under /var/lib:

```
mkdir -p /var/lib/named/etc
```

mkdir /var/lib/named/dev
mkdir -p /var/lib/named/var/cache/bind
mkdir -p /var/lib/named/var/run/bind/run

Then move the config directory from /etc to /var/lib/named/etc:
mv /etc/bind /var/lib/named/etc

Create a symlink to the new config directory from the old location (to avoid problems if bind is upgraded in the future):
ln -s /var/lib/named/etc/bind /etc/bind

Make null and random devices, and fix permissions of the directories:
mknod /var/lib/named/dev/null c 1 3
mknod /var/lib/named/dev/random c 1 8
chmod 666 /var/lib/named/dev/null /var/lib/named/dev/random
chown -R bind:bind /var/lib/named/var/*
chown -R bind:bind /var/lib/named/etc/bind

We can start up bind9 at this point:
/etc/init.d/bind9 start

Let's see if the service is running:
lsof -i | grep :domain

It is also imperative *that after each change we make, we look in our syslog to see if bind9 reported any errors. Here we use* less *to view syslog. Once we are viewing the file,* ***use an uppercase 'G' to go to the bottom of the file*** *(and a lowercase 'q' to quit):*
less /var/log/syslog

We are setting up bind9 as a local caching only name server (later we will optionally configure it as a forwarding server). Here we add some additional security measures that prevent unauthorized machines from using our name server:
vi /etc/bind/named.conf.options

On the line below "directory" we want to add a line that restricts use of our name server to the network our spamfilter is on. Place a [Tab] in front of the entry so it lines up with the other entries. You can add more than one network here if you like. Place a ";" (semicolon) after each network. Note that if you actually want to allow other clients to connect to our name server, as explained in the notes above you would also have to modify IP tables to allow this.
allow-query {222.222.222.222/24;};

Save and exit the file, then I would restart bind9 and check that it is running:
/etc/init.d/bind9 restart
lsof -i | grep :domain

Optionally configure bind9 as a forwarding server. *Bind9 as we have it*

configured now will first query the root servers for hints when needed. I prefer to forward queries to another name server instead. There are advantages and disadvantages in doing this, but I prefer it. It is absolutely imperative that any name servers listed here are known to work from our spamfilter. These will almost certainly be the primary and secondary servers you currently have configured in /etc/resolv.conf (not 127.0.0.1, and not the IP address of the local machine) or your ISP's servers. However, they should preferably point to real name servers and not a DNS proxy like your Linksys broadband router or other gateway device (unless that proxy does not allow proper access to real name servers outside your network - which is sometimes the case). Add the 'forwarders' entry just below the 'allow-query' entry we just made:
vi /etc/bind/named.conf.options
and add:
forwarders {444.444.444.444; 555.555.555.555;};
To never query the root servers, optionally add (personally I do add this):
forward only;

Save and exit the file, then once again I would restart bind9 and check that it is running:
/etc/init.d/bind9 restart
lsof -i | grep :domain

And once again, check for errors:
less /var/log/syslog

Once it is determined bind9 is functioning, you can change the primary nameserver in /etc/resolv.conf:
vi /etc/resolv.conf

and replace the entry for the primary nameserver (the first one listed):
nameserver 444.444.444.444
to the IP address of this machine (our real IP address, not 127.0.0.1):
nameserver 111.111.111.111

Save and exit the file, then test that we are able to resolve host names:
dig yahoo.com

You should see valid data (A records that have IP addresses), and the output will also tell you which name server was used to find the information:
;; SERVER: 111.111.111.111#53(111.111.111.111)

Make sure this shows this machines IP address as configured in /etc/resolv.conf.
Now we can tell Postfix to use the new name server:
LINUX2

****

Test the Installation:

Index
****

Let's start off fresh:

```
reboot
logout
```

If you were at the console you might happen to see the message "SpamAssassin Mail Filter Daemon: disabled, see /etc/default/spamassassin"; please ignore it, we do NOT want to enable spamd.

Other mail servers will need to find your server. Set up name resolution of sfa.example.com to its IP address. How this is done depends on your environment. Basically, you need to make sure that any box that needs to talk to this mail server can resolve its name, either through appropriate DNS server entries somewhere, or local "hosts" files on those machines.

Now let's see if this box is actually an email server!
If necessary add a new entry in the hosts file on your Windows computer that points to the spamfilter:

111.111.111.111 sfa.example.com

You should already have a mailbox set up for "spambin". Create a user in your email client (MUA) for "spambin" so you can send and receive email from that account if you have not already done so. Temporarily set your email client's SMTP (Outgoing) server to sfa.example.com for the "spambin" account so any mail it sends gets sent back through the spamfilter. It is required however to whitelist "spambin" so the email doesn't end up back in the "spambin". This is the problem with forwarding email from the "spambin" (or "banned" or "virii") client. Any email you forward out of these mailboxes may end up back in the quarantine. Once we are done testing, you should set up the "spambin", "banned" and possibly "virii" email clients to send their email directly to the Exchange server. Of course you would never actually forward viruses to end users or open any viruses in this quarantine, but you may occasionally find legitimate mail that was quarantined due to a banned attachment. You should obviously have virus protection on the client machine.

Send yourself an email from the "spambin" account. Wait a minute, and then "Get new mail" or "Send and Receive" or whatever. If you don't get the email "spambin" sent you, obviously "Houston, we have a problem". You might check the mail log on the spamfilter for clues:

```
less /var/log/mail.log
```

If you would like to see the log go by in real time:
tail -f /var/log/mail.log
I use this command extensively and I suggest you use it after any changes are made to Postfix or amavisd-new. This will tell you if your mail system is receiving and processing mail, or if your change just killed the system.

and/or use
amavisd-new stop
amavisd-new debug-sa
or
amavisd-new debug
for more detail.

You can increase the level of detail reported in mail.log by adjusting $log_level in amavisd.conf. Remember to set it back to 0 when finished with your debug session.

cd /var/mail
Then less any files you may find there. There may be a file or two that ended up there before Postfix was configured. Also try the commands mailq and qshape (and qshape deferred) to see if there is mail stuck in the queue. Use amavisd-new debug while you are sending mail through the system to help provide clues to the problem. Open another terminal window and run mailq while the other window is running amavisd-new debug . If you made changes to configuration files and want to flush the queue, try postconf -f and if that does not work try postsuper -r ALL . See http://www.postfix.org/postsuper.1.html. Insure that the relay_domains parameter has been configured correctly. Inspect the /etc/postfix/transport file for errors, and make sure you run postmap /etc/postfix/transport every time you make changes.

If you have problems, the most likely reason is a typographical error, or a wrong IP or network address in one of the configuration files. Hopefully, you know Unix/Linux is cAse senSitiVe. Maybe you accidentally skipped one of the configuration steps. Some of the steps cannot be performed more than once however, so be careful. Remember, read /var/log/mail.log as a first step.

If you do get the email back, change whatever setting or use whatever command or tool you have that will enable you to look at the full headers of the email message. Examine the headers and behold your creation. If you sent an empty message with no subject, Pyzor might just flag it as spam!

As time goes by be sure to keep an eye on the number of files and temporary directories that are created in the /var/lib/amavis directory and its subdirectories. If there are a large number of files and/or temporary directories it would indicate some sort of problem. If you choose to keep quarantined messages on this box instead of sending them to spambin@example.com then they should accumulate in the virusmails directory; that would be considered normal. I have seen on many occasions

people report that amavisd-new fails to delete temporary directories (their name begins with amavis-20) and there seems to be a lack of understanding as to the reason this occurs. It may possibly indicate a problem with one of the shared libraries that amavisd-new uses (uulib zlib etc.) or a problem with Perl itself. See http://www.ijs.si/software/amavisd/#faq-gen. I have seen reports of people writing scripts that delete these temporary directories on a regular basis to work around the problem but this is only sweeping the underlying problem under the rug.

Whenever you edit a few certain files like the hosts file on the spamfilter, we need to supply Postfix with an up to date copy of it or will complain: warning: /var/spool/postfix/etc/hosts and /etc/hosts differ

```
LINUX2

postfix check
```

Congratulations, you did a great job. It's either time for that 4th cup of coffee, or maybe just finish the next few sections and call it a day, because in many respects, we are just getting started.

Huh?

****

Installing ClamAV:

Index

****

This entire section is optional but recommended. It would be a good idea to get ClamAV configured and running, even if you choose to disable it. It's reasonably easy to turn it on and off. If you use this product regularly, consider a donation. It's also a good idea to read the mailing list archives http://lurker.clamav.net/list/clamav-users.en.html and join the clamav-announce mailing list http://lists.clamav.net/cgi-bin/mailman/listinfo/clamav-announce so you will be notified of new releases. Keep in mind that after a new version is released, it may take a couple days for the Debian package maintainer to update Debian.

Read http://www200.pair.com/mecham/spam/clamav-amavisd-new.html if you would like to gain a better understanding of what we are doing here.

```
apt-get update

Then install clamav, and clamav-daemon (from etch stable).
apt-get install clamav clamav-daemon clamav-freshclam
```

As etch gets older you will want to upgrade clamav from the 'Volatile' branch:
apt-get -t etch install clamav clamav-daemon clamav-freshclam

Now, very important, add the clamav user to the amavis group:
gpasswd -a clamav amavis

Configure amavisd.conf so amavisd-new will use ClamAV:

vi /etc/amavis/amavisd.conf

To enable virus scanning, make sure this line is commented out (like this):
@bypass_virus_checks_maps = (1); # uncomment to DISABLE anti-virus code

Locate the line:
@av_scanners = (

Uncomment these 4 lines, then make sure the value after CONTSCAN reads as follows:

```
  ['ClamAV-clamd',
    \&ask_daemon, ["CONTSCAN {}\n", '/var/run/clamav/clamd.ctl'],
    qr/\bOK$/, qr/\bFOUND$/,
    qr/^.*?: (?!Infected Archive)(.*) FOUND$/ ],
```

*While you are at it, if you would like to clean up amavisd.conf a little, you may optionally delete all the virus scanners **between** ClamAV and BitDefender. We want to keep BitDefender. It would be a very good idea to make a backup copy of amavisd.conf before you do this.*

Note: if you have increased message_size_limit in main.cf from the default 10MB then you will probably want to increase ArchiveMaxFileSize in /etc/clamav/clamd.conf also. It's probably a good idea to make this a couple MB larger than message_size_limit (up to some maximum of maybe 35MB or so).

Note that this /var/run/clamav/clamd.ctl value shown above must match the LocalSocket parameter in /etc/clamav/clamd.conf Change it here if necessary to match what is in /etc/clamav/clamd.conf.

The configuration file for ClamAV 0.90.1 is /etc/clamav/clamd.conf. The freshclam daemon is set to check for, and download if necessary, new virus definitions 24 times per day. The configuration file for freshclam is /etc/clamav/freshclam.conf You can check the log files at /var/log/clamav/freshclam.log and /var/log/clamav/clamav.log The startup script for freshclam is /etc/init.d/clamav-freshclam and for clamd it's /etc/init.d/clamav-daemon You can also research additional clamd.conf configuration options by running man clamd.conf

Let's reboot (it might make a difference), and then test the system:
reboot

As an alternate to rebooting, you can stop and start clamd and amavisd-new:
amavisd-new stop
/etc/init.d/clamav-daemon stop
/etc/init.d/clamav-daemon start
amavisd-new start

Once the system comes back up, we need to test ClamAV. To do so, we can simply watch the mail.log go by as we send the Eicar test virus through the system:
tail -f /var/log/mail.log

Send the Eicar test virus through as described below. After you are finished sending the virus through, use [Ctrl]+c to return to the shell prompt.

Once this machine is up and running in a production environment you should only reboot when necessary. There are many processes writing to the disk in this system and killing those processes is sometimes not a good idea.

Now you need to send a test virus to yourself, like eicar.com.txt from http://www.eicar.org/anti_virus_test_file.htm. Note: **rename eicar.com.txt** to something like eicar.txt because amavisd-new bans files with double extensions. Attach eicar.txt to the email you send. Don't use eicar.com or eicar_com.zip. Of course you would have to temporarily disable your desktop antivirus software first. You can use tail -f /var/log/mail.log so you can see what happens.

If ClamAV is not working you may get "Can't connect to UNIX socket". This could indicate a couple of things. The LocalSocket parameter in /etc/clamav/clamd.conf must be the same as the one in /etc/amavisd/amavisd.conf as noted above. The clamav user must also belong to the same group that the amavis user belongs to so both users can write to the UNIX socket file, in our case "clamd.ctl".

I have a funny story. I had this all working fine but it appeared that clamd thought that the eicar_com.zip file was "OK", in other words, it scanned it, but did not detect that is was a virus. It turned out my Internet proxy (WinProxy) had removed the virus from inside the file before I scanned it! It took me an hour to figure it all out.

If you have problems that you can't seem to solve, or you are upgrading to a new major release, completely uninstall the program, and then try another version if available:

apt-get --purge remove clamav clamav-base clamav-daemon clamav-freshclam libclamav2

Watch for errors as the program is being uninstalled, you may have to manually remove any clamav directories that dpkg failed to remove (rm -r). Always be very careful when using the 'rm -r' command.

If you just can't get it working,
uncomment @bypass_virus_checks_maps in /etc/amavis/amavisd.conf to disable virus scanning (or go shopping for another antivirus vendor). Also: make sure everything works AFTER you reboot. Don't leave ClamAV in a non-functional state if you have virus scanning enabled. Amavisd-new will not work properly. Keep an eye on /var/log/clamav/freshclam.log and /var/log/clamav/clamav.log. You need to look for error messages in these files. You may have an error or two when the program is first installed; this is probably OK and may be due to things happening out of sequence. Check the logs for a couple days and make sure the database updates do not fail and that Clamd is notified of updates.

When it is time to upgrade to a new version of ClamAV:
Run 'apt-get update'.
If you are upgrading to a MAJOR new version, completely uninstall ClamAV as described above.
Install the new version of clamav and clamav-daemon.
Run 'gpasswd -a clamav amavis'.
I suggest rebooting, if possible.
Run 'tail -f /var/log/mail.log', and send a test message containing the eicar "virus" through.

There is a third party http://www.sanesecurity.com/clamav/ that maintains additional anti-phishing and scam mail signatures. If you would like to add these I have customized for Debian one of the scripts used to download the signatures:

Optionally *use additional anti-phishing and scam signatures:*
apt-get install curl rsync

cd /usr/sbin
wget http://www200.pair.com/mecham/spam/UpdateSaneSecurity.sh.txt
mv UpdateSaneSecurity.sh.txt UpdateSaneSecurity.sh
chmod +x UpdateSaneSecurity.sh
UpdateSaneSecurity.sh
ls -l /var/lib/clamav

You will notice the data has been downloaded:
drwxr-xr-x 2 clamav clamav 4096 2007-06-16 19:48 daily.inc
-rw-r--r-- 1 clamav clamav 9351789 2007-06-10 21:16 main.cvd
-rw------- 1 clamav clamav 260 2007-06-16 19:14 mirrors.dat
-rw-r--r-- 1 clamav clamav 347982 2007-06-16 19:25 MSRBL-Images.hdb
-rw-r--r-- 1 clamav clamav 228232 2007-06-08 04:33 MSRBL-SPAM.ndb
-rw-r--r-- 1 clamav clamav 1033688 2007-06-16 19:48 phish.ndb
-rw-r--r-- 1 clamav clamav 174338 2007-06-15 02:55 phish.ndb.gz
-rw-r--r-- 1 clamav clamav 516182 2007-06-16 19:48 scam.ndb
-rw-r--r-- 1 clamav clamav 102738 2007-06-15 02:55 scam.ndb.gz
Now we add a crontab entry with download attempts performed every 4th hour:
crontab -e

Insert this entry. ***Replace MM*** *(minutes) below with a number between 1*

and 59:
MM */4 * * * /usr/sbin/UpdateSaneSecurity.sh

Save and exit the file. The above cron job should run every four hours. Logs of the last download are located in /var/tmp/clamdb/

Amavisd-new 2.5.0 or newer can treat these 'viruses' as spam. You just need to add some SpamAssassin rules so they score more than 0.1:
cd /etc/spamassassin
wget http://www200.pair.com/mecham/spam/amavis-sanesecurity.cf
spamassassin --lint

At some point in the future you may get tired of wading through all the comments in amavisd.conf. Here's how to remove them:

***Optionally** remove most comments from amavisd.conf:*
cp /etc/amavis/amavisd.conf /etc/amavis/amavisd.conf-verbose
grep -vE '^$|^#' /etc/amavis/amavisd.conf > /etc/amavis/amavisd.conf-temp
cp /etc/amavis/amavisd.conf-temp /etc/amavis/amavisd.conf

****

Tweaking Notification Settings:

Index
****
This may be a good time to experiment with some of the notification settings in amavisd.conf. Change some of these settings and send yourself email containing viruses, banned attachments, spam, and attachments with double extensions. Do one thing at a time and take notes on what happens and who gets notified. Remember to stop and start (or debug) amavisd-new after each change. Settings to play with are:
$final_virus_destiny
$final_banned_destiny
$final_spam_destiny
$final_bad_header_destiny
$warnvirussender
$warnspamsender #(not really used any more)
$warnbannedsender
$warnbadhsender
$warnvirusrecip
$warnbannedrecip
$warn_offsite

You can uncomment and set $spam_admin if you would like to get detailed diagnostic messages for each spam:
$spam_admin = "postmaster\@$mydomain";

Play with these until you have everything the way you prefer it (as far as notification and destination of all the garbage email goes). Experiment with them a little to get an idea of how they work. I recommend D_DISCARD for $final_virus_destiny in this day and age. Since many modern viruses and worms (like MYDOOM) fake the sender, D_BOUNCE just creates extra useless and confusing Internet traffic and it also clogs up your queues with mail that can't be delivered. It will still quarantine the message and notify the postmaster when a virus is found. If you would like all email containing a virus (or a banned attachment) to disappear into a black hole,
set $virus_quarantine_to = undef;.

****

Back up critical files:

Index
****
Let's back up some of our hard work to a floppy. Grab a plain old High-Density 1.44MB floppy disk, and head on over to the spamfilter. Insert the floppy, and then head back to your Windows machine. Or just type this stuff in at the console if it's a long walk:

We may have to create a device if it does not exist:

```
test -e /dev/fd0u1722 || mknod /dev/fd0u1722 b 2 60
chmod 660 /dev/fd0u1722
chown root:floppy /dev/fd0u1722
```

Create the mount point:

```
mkdir /floppy
```

Format the floppy (insert it first of course):

```
fdformat /dev/fd0u1722
```

Create a file system on the floppy (we need one that accepts long file names):

```
mke2fs /dev/fd0u1722
```

Mount the floppy:

```
mount /dev/fd0u1722 /floppy
```

*Yes I know, working with floppies in *nix is a pain.*

And copy all these files to it: (Yes you can copy and paste this whole section).

```
cp /etc/fstab /floppy
cp /etc/aliases /floppy
cp /etc/postfix/main.cf /floppy
cp /etc/postfix/master.cf /floppy
cp /etc/postfix/sender_access /floppy
cp /etc/postfix/transport /floppy
cp /etc/postfix/virtual /floppy
cp /etc/postfix/relay_recipients /floppy
cp /etc/postfix/body_checks /floppy
cp /etc/postfix/header_checks /floppy
cp /etc/amavis/amavisd.conf /floppy
cp /etc/spamassassin/local.cf /floppy
cp /var/lib/amavis/.spamassassin/user_prefs /floppy
cp /var/lib/amavis/.razor/razor-agent.conf /floppy
cp /etc/apt/sources.list /floppy
cp /etc/apt/preferences /floppy
cp /etc/clamav/clamd.conf /floppy
cp /etc/clamav/freshclam.conf /floppy
```

Wait a minute for the files to copy over, and then list the floppy disk's contents:

```
ls -l /floppy
```

This floppy will not be readable by a Windows machine.
I can't repeat this enough:
Always unmount the floppy before you remove it:
I suggest you have the monitor on at the console so you can see the mess you make if you don't.

```
umount /floppy
```

Remove it, label it and store it in a safe place.

If you wanted to restore the files at some point in the future:

```
mount /dev/fd0u1722 /floppy

cp /floppy/aliases /etc/aliases
newaliases

cp /floppy/main.cf /etc/postfix/main.cf
cp /floppy/master.cf /etc/postfix/master.cf
cp /floppy/sender_access /etc/postfix/sender_access
postmap /etc/postfix/sender_access

cp /floppy/transport /etc/postfix/transport
postmap /etc/postfix/transport

cp /floppy/virtual /etc/postfix/virtual
```

```
postmap /etc/postfix/virtual

cp /floppy/relay_recipients /etc/postfix/relay_recipients
postmap /etc/postfix/relay_recipients

cp /floppy/body_checks /etc/postfix/body_checks
cp /floppy/header_checks /etc/postfix/header_checks
cp /floppy/amavisd.conf /etc/amavis/amavisd.conf
cp /floppy/user_prefs /var/lib/amavis/.spamassassin/user_prefs
cp /floppy/local.cf /etc/spamassassin/local.cf
cp /floppy/razor-agent.conf /var/lib/amavis/.razor/razor-agent.conf
cp /floppy/clamd.conf /etc/clamav/clamd.conf
cp /floppy/freshclam.conf /etc/clamav/freshclam.conf
cp /floppy/sources.list /etc/apt/sources.list
cp /floppy/preferences /etc/apt/preferences
apt-get update

umount /floppy
```

then shutdown and restart amavisd-new and Postfix as needed. Note that we do not restore /etc/fstab, it is just a real good idea to have a copy of to refer to. It describes our partition layout.

If you have an ftp server that you can upload files to, (maybe you got 10MB of web space with a dial up account you have) I will quickly describe backing up the files to that server. It's a good idea to have 2 terminal windows open so we can have ftp running in one, and the command line in the other.

Stop Postfix and amavisd-new so the bayes files are not written to during the backup:

```
postfix stop
amavisd-new stop
su amavis -c 'sa-learn --sync'
```

Start your ftp session (substituting your settings):

ftp -p **server.domain.tld**

Enter your user name and password as requested, and then make a directory to place our files:

ftp> mkdir sfa *(only necessary the first time you do this)*

Change to that directory:

ftp> cd sfa

Then simply copy and paste this entire section:

```
ascii
put /etc/aliases aliases
put /etc/postfix/main.cf main.cf
put /etc/postfix/master.cf master.cf
put /etc/postfix/sender_access sender_access
put /etc/postfix/transport transport
```

```
put /etc/postfix/virtual virtual
put /etc/postfix/relay_recipients relay_recipients
put /etc/postfix/body_checks body_checks
put /etc/postfix/header_checks header_checks
put /etc/amavis/amavisd.conf amavisd.conf
put /etc/spamassassin/local.cf local.cf
put /etc/apt/sources.list sources.list
put /etc/apt/preferences preferences
put /etc/clamav/clamd.conf clamd.conf
put /etc/clamav/freshclam.conf freshclam.conf
put /var/lib/amavis/.razor/razor-agent.conf razor-agent.conf
put /var/lib/amavis/.spamassassin/user_prefs user_prefs
binary
put /var/lib/amavis/.spamassassin/auto-whitelist auto-whitelist
put /var/lib/amavis/.spamassassin/bayes_toks bayes_toks
put /var/lib/amavis/.spamassassin/bayes_seen bayes_seen
ls -l
pwd
quit
amavisd-new start
postfix start
```

Use `get` instead of `put` to restore a file. For example: `get clamd.conf /etc/clamav/clamd.conf`
Don't get your servers mixed up if you have more than one!

****

Set up security reports:

Index
****

We are first going to install `logcheck`

```
apt-get install logcheck logcheck-database

Read the message that comes up. If you would like to change any settings:

vi /etc/logcheck/logcheck.conf
```

Take a moment to test logcheck. Open a second PuTTY session and log in as root but **use the wrong password** on your first attempt to log in. Now run `su -s /bin/bash -c "/usr/sbin/logcheck" logcheck` at the command prompt, wait a minute, and then check the mailbox of the address you configured logcheck to send email. You should get a message indicating the security violation. Logcheck looks for suspicious activity and is scheduled to run once each hour. You will only get a message if it finds something suspicious, but this is configurable. Don't set it to "paranoid", there would be an entry for every email that passed through the system. Read more about logcheck by using `less /usr/share/doc/logcheck/README.logcheck`

If you get annoying repetitive emails from logcheck, try editing /etc/logcheck/ignore.d.server/logcheck and insert a regular expression of the text you wish logcheck to ignore. For example, I inserted
CRON.*: \(pam_unix\) session opened for user
CRON.*: \(pam_unix\) session closed for user

You most likely will not want to be annoyed by every message amavisd-new produces so install this file:
cd /etc/logcheck/ignore.d.server
wget http://www200.pair.com/mecham/spam/logcheck/amavisd-new

Logcheck will also look for keywords (like "attack") contained in the /etc/logcheck/cracking.d/logcheck file. We can tell logcheck to ignore log entries that contain a hostname such as "attackingthedevil.co.uk" by creating a new file in the appropriate logcheck "ignore" directory and placing that text in it. We create a file because none exist at this point:
echo "attackingthedevil.co.uk" >> /etc/logcheck/cracking.ignore.d/logcheck-postfix

If you removed the DCC log directory as I suggested, dccifd will log an error every time it is called. We want logcheck to ignore those log entries (this is a single command):
echo "stat\(log directory /var/dcc/log\): No such file or directory" >>
/etc/logcheck/ignore.d.server/dcc

While we are at it, there is another dcc message we don't care about:
echo ": missing message body" >> /etc/logcheck/ignore.d.server/dcc

and also:
echo "detected 0 spam, ignored for 0, rejected for 0," >>
/etc/logcheck/violations.ignore.d/dcc

and a Postfix message I wish to suppress:
echo "dsn=2.7.0, status=sent \(254 2.7.0 Ok," >>/etc/logcheck/ignore.d.server/postfix

And lastly:
echo "\(su amavis -c '/usr/bin/razor-admin -discover'\)" >>
/etc/logcheck/violations.ignore.d/logcheck-cron

For more information about logcheck rules and patterns to include or ignore:
vi -R /usr/share/doc/logcheck-database/README.logcheck-database.gz
and to debug logcheck:
su -s /bin/bash -c "/usr/sbin/logcheck -otd" logcheck

Subscribe to http://lists.debian.org/debian-security-announce/ and visit http://www.debian.org/security/
You will get an email from them every time any one of approximately 15,000 programs in the Debian catalog have been found to have a security flaw, along with some instructions. You will not have many of those programs installed on your system. The 'stable' version of a security related software

update is released first. The 'testing' and/or 'unstable' versions may take a couple weeks to get released. To do a quick check whether you have a program on your system, try:
which [program name] or whatis [program name] or dpkg -l [package name]

And to get a somewhat full list of packages that are installed on your system:

```
cd

Use dpkg and grep to send the list of installed program files to a file called "progs":
dpkg -l '*' | grep '^i' > /root/progs

Then use less to view the file:
less /root/progs

Or send it in an email to root:
cat /root/progs | mail -s "sfa installed programs" root
```

Note that this **will not include** programs like DCC that you installed by means other than dpkg or apt-get.

I like to be notified each morning that the mail queue on our system is empty (or nearly empty, depending on how busy our system is) because if it is not, we may have a problem. Note that if you do not use the "relayhost" parameter in /etc/postfix/main.cf, you should have items in the mailq most of the time. These will most likely be NDRs that cannot be delivered. If the queue is full of legitimate email, the most likely cause is our Exchange server is down. If you don't get a mailq email at all, it is likely the spamfilter is down. We will only display the last 10 lines of output from the mailq command.

```
crontab -e

And insert at the first available blank line (actually, this report is optional):
30 7 * * * /usr/bin/mailq |/usr/bin/tail |/usr/bin/mail -s "mailq sfa" root

While we are editing crontab, for the forgetful type (forgot that you are not supposed to run 'sa-learn'as root) we will make sure 'amavis' still owns the Bayes and AWL files:
30 17 * * * /bin/chown -R amavis:amavis /var/lib/amavis/.spamassassin

Save and exit, then see if you have this script:
ls -l /etc/init.d/hwclock.sh

If you do not have this script, we will update the hardware clock once each day. Edit crontab once again:
crontab -e

And insert:
10 10 * * * /sbin/hwclock --systohc
```

In addition to mailq (or as an alternate) you may wish to use qshape

```
vi /etc/cron.d/qshape-cron

And insert (2 lines):
PATH=/usr/sbin
31 7 * * * postfix /usr/sbin/qshape incoming active deferred 2>&1 |/usr/bin/mail -s "qshape sfa" root

Save and exit. Logcheck will now complain each day, so we need to shut it up:
vi /etc/logcheck/ignore.d.server/postfix

and insert at the top of the file (1 line):
^\w{3} [ :0-9]{11} [._[:alnum:]-]+ postfix/pickup\[[0-9]+\]: [[:alnum:]]+: uid=[0-4]+ from=<postfix>
```

****

Set up intrusion detection:

Index

****

Don't start this section unless you have a good hour to work on it. This part drove me nuts for a couple of days. I have used tripwire on my Red Hat boxes and felt it was pretty straightforward after I understood the concepts and the commands. When I tried it on Debian (it is available with apt-get) it simply refused to run (segmentation faults and such). I could not find the cause after Googling for a good hour. Any way, I gave up and looked for alternatives. I decided to give AIDE a try since it was also available as a Debian package. Be forewarned you may not enjoy this part. Be aware that your Linux box is an ideal platform to launch an attack on your entire network. So implementing this is highly recommended.

The concept with AIDE is to take a snapshot of some of the characteristics of critical files on our system, and then store that information in a database on read-only media; a write protected floppy for example. Then run a program that compares the current state of those files on our system with the database, and report any differences. We should be able to tell if a hacker has compromised our system if the characteristics of the files on our system differ from the files in the database.

There are some challenges with doing this. Obviously, during normal use, some files change, like log files. So one challenge is to cut down on the unnecessary "chatter" that appears in the report. Adding and removing programs is considered normal use but this type of activity will generate alarms. When we are at the point the report has grown and is causing us to

spend more time reading it than it should, we generate a new database and move it to the read only media. We have to be sure our system has not been compromised at the time we create a new database, so it would be wise to run a report before any new "trusted" software is installed, then another one after, then create a new database so we don't have to look at all that activity on every subsequent report.

When we install the program, it sets everything up on the hard drive, allows us to create the database right off the bat and it sets up a cron job to send us a report each day. This is fine, but completely insecure. A Hacker could replace the program, so we have to move it to the write protected media. We are going to use a write protected floppy disk, for convenience. The configuration file could be hacked, so we have to move it. And the same thing goes for the script that generates the report. These all have to be moved to write protected media. We can add an entry in crontab to run the report script off the floppy, and we will, but keep in mind that this can be hacked also, so the only way to be sure the report has not been tampered with is to run it from the command line off the write protected floppy.

Let's install aide:

For consistency, we are not using the most current version of AIDE:

```
cd /usr/local/src
wget http://www.xmission.com/~jmcrc/aide_0.10-4_i386.deb
dpkg -i aide_0.10-4_i386.deb
```

The installation will have three input screens, answer them as follows:

Daily reports are mailed to root by default. **[Ok]**

Initialize aide database? **[No]**

Before AIDE can be used, you will have to initialize a database. **[Ok]**

The installation should create these files:
/usr/bin/aide = the executable program file
/etc/aide/aide.conf = the configuration file
/etc/cron.daily/aide = a nice script that runs the report

Of course, other files and directories were created, but these are the only files we care about at this time.

We are going to move these files to /root/aide then uninstall the program.

Make a new directory, a convenient place to store the files we want:

```
mkdir /root/aide
```

Then copy the files, note that we will rename the report script "report":
cp /usr/bin/aide /root/aide
cp /etc/aide/aide.conf /root/aide
cp /etc/cron.daily/aide /root/aide/report

Then we remove most aide components:
rm -r /var/lib/aide
rm -r /etc/aide
rm /usr/bin/aide
rm /usr/sbin/aideinit
rm /etc/cron.daily/aide

Let's work on the configuration file:

You can use the WinSCP editor if you prefer:
vi /root/aide/aide.conf

Change:
database=file:**/var/lib/aide**/aide.db
To:
database=file:**/floppy**/aide.db

Change:
database_out=file:**/var/lib/aide**/aide.db.new
To:
database_out=file:**/root/aide**/aide.db.new

Under Custom Rules, edit Binlib, and remove:
+m +c

Edit Devices, and remove:
+i +g +c

under # Kernel, change:
=/boot$ Binlib
to:
/boot Binlib

Comment out the entire section under # Log Files

Under # Check crontabs *add:*
/etc/cron.d Databases
/etc/cron.daily Databases
/etc/cron.hourly Databases
/etc/cron.monthly Databases
/etc/cron.weekly Databases
/etc/crontab Databases

Save and exit.

These are only suggestions; you can tweak this to your hearts content. More info can be found at:
'man aide' and 'man aide.conf'

Let's edit the report script:

vi /root/aide/report

Change:
PATH="/bin:/usr/bin"
To:
PATH="**/floppy:/bin:/usr/bin**"

Change:
CONFFILE="**/etc/aide**/aide.conf"
To:
CONFFILE="**/floppy**/aide.conf"

Change:
[-f **/usr/bin**/aide] || exit 0
To:
[-f **/floppy**/aide] || exit 0

*If you would like to send the report to someone other than root, **optionally** change:*
MAILTO="${MAILTO:-**root**}"
To something like:
MAILTO="${MAILTO:-**someuser@example.com**}"

Change:
DATABASE="${DATABASE:-**/var/lib/aide**/aide.db}"
To:
DATABASE="${DATABASE:-**/floppy**/aide.db}"

Change:
COMMAND="${COMMAND:**-check**}"
To:
COMMAND="${COMMAND:**-update**}"

Change:
aide $AIDEARGS --$COMMAND >"$LOGDIR/$LOGFILE" 2>"$ERRORTMP"
To:
aide $AIDEARGS **--config=/floppy/aide.conf** --$COMMAND >"$LOGDIR/$LOGFILE" 2>"$ERRORTMP"

Save and exit.

If you have a floppy in the drive, unmount it (umount /floppy) then remove it. Insert a floppy in the drive. We are going to format it, create an ext2 file system on it, mount it, and then copy the AIDE files to it.

We may have to create a device if it does not exist:

```
test -e /dev/fd0u1722 || mknod /dev/fd0u1722 b 2 60
chmod 660 /dev/fd0u1722
chown root:floppy /dev/fd0u1722
```

Make sure we have a mount point:

```
mkdir /floppy
```

Run these one at a time and wait for each command to finish:

```
fdformat /dev/fd0u1722

mke2fs /dev/fd0u1722

fsck /dev/fd0u1722

mount /dev/fd0u1722 /floppy
```

Then copy these files over to the floppy:

```
cp /root/aide/aide /floppy
cp /root/aide/aide.conf /floppy
cp /root/aide/report /floppy
```

Wait for the files to copy over (30 seconds).
Now we can run the routine that creates the first database. We only need to run this once:

```
/floppy/aide --config=/floppy/aide.conf --init
```

This creates aide.db.new in the /root/aide directory, this is the initial database and must be copied to the floppy as aide.db.

```
cp /root/aide/aide.db.new /floppy/aide.db
```

Wait long enough (20 seconds) for the file to copy over, and then run the report:

```
/floppy/report
```

Check your inbox for the report. You may want (or need) to fix a couple things in /floppy/aide.conf or /floppy/report. Then run the report again. Then copy /root/aide/aide.db.new to /floppy/aide.db again. The settings in aide.conf may need to be tweaked over the next couple weeks. You can check for errors by using `cat /var/log/aide/error.log`

Now you have to write protect the floppy disk. This is the most important part.

```
umount /floppy
```

Then pull out the disk and push up the write protect tab.
Reinsert the floppy, and mount it again:

```
mount /dev/fd0u1722 /floppy
```

This floppy should remain in the drive all the time.
(Unless we need to use the drive for a moment)

Each time you run the report, it reads the database /floppy/aide.db but it also creates a new aide.db.new database. So why do we care about the aide.db.new? It's a matter of convenience. If the report is getting kind of long, and there appears to be nothing in it you would not expect; the aide.db.new database reflects the current state of the system and could be used to replace aide.db. You replace your aide.db on the floppy with aide.db.new, just as you did when you first initialized the system. If you run the report again, it should be clean, unless files have changed since it was created. You need to understand the risks of relying on the quality of the aide.db.new database if you use the copy that was created during the cron job. The cron job could have been hacked. A hacker could manipulate the file. Only rely on the aide.db.new database if you run /floppy/report from the command line and you inspect the report just prior to copying it over to the floppy. If you left the write protection off the floppy drive for any length of time, it's possible you can no longer rely on the database. Be paranoid.

So here's the gist of maintenance:

- Run the report.
- Read the report.
- If it looks normal, and it has not grown to the point it is annoying, you're done for today.
- If the report lists any critical system files you know you did not change or update, or you notice other suspicious activity,
  I hope you have a cloned hard drive ready to swap out because you may have been hacked. Study the report carefully.
- If everything looks normal and you want it cleaned up, unmount the floppy, flip the write protect tab off, mount the floppy,
  copy /root/aide/aide.db.new to /floppy/aide.db, unmount the floppy,
  flip the write protect tab on, mount the floppy, done.

Note that every time you make a change to the configuration file (aide.conf) you will need to run the report, and then copy the new database to the floppy, otherwise every file in the aide database may possibly be listed as "File /filename in databases has different attributes". This is not an indication the files have changed; it is an indication there is a mismatch between aide.conf and aide.db.

You could put the command /floppy/report in your crontab (we *will* do so), and have it run each day just before you normally open your email, but a hacker could send forged reports for weeks before you knew there was a problem. Be aware of the danger. For a trustworthy report simply run the

report from a command line every day, few days, week, few weeks, month, whatever your level of paranoia permits.

```
crontab -e

And insert:
25 7 * * * /floppy/report

Save and exit
```

Ok, you should make a backup copy of the floppy, because floppies are not reliable:

```
Change to our home directory:
cd

Unmount the floppy, but leave it in the drive:
umount /floppy

Create an image of the floppy and store it to a file:
dd if=/dev/fd0u1722 of=floppy.img

dd stands for `Copy and Convert' and was renamed to `dd` only because `cc'
was reserved for the C compiler. if= input file, of= output file

Remove the source disk, insert the destination disk:
First, format the new floppy:
fdformat /dev/fd0u1722

Then copy the image to the new floppy:
dd if=floppy.img of=/dev/fd0u1722

Simply use the new disk now. Turn write protect on, then mount the floppy:
mount /dev/fd0u1722 /floppy
```

If you think about it for a minute, you will realize that one of these floppies can be used in another machine. You would create the /var/log/aide and /root/aide directories, copy /floppy/aide.db to /root/aide/aide.db.new, create the cron job and the startup script (our next item). Then run the report and copy aide.db.new over to /floppy/aide.db.

We will create a special startup script called /etc/init.d/startflop that runs when the system boots up. We will use this script to mount the floppy automatically when we boot up. There are other ways to accomplish this task, but this one works best. Depending on your hardware, you may need to set your BIOS to boot from the hard drive first to avoid it from attempting to boot from the floppy.

```
vi /etc/init.d/startflop
```

And insert the following text just as it is listed here:

```
#! /bin/sh
#
# very simple startup script

case "$1" in
   start)
   /usr/bin/test -e /dev/fd0u1722 || /bin/mknod /dev/fd0u1722 b 2 60
   /bin/chmod 660 /dev/fd0u1722
   /bin/chown root:floppy /dev/fd0u1722
   /bin/mount /dev/fd0u1722 /floppy
      ;;
   stop)  /bin/umount /floppy
      ;;
esac

exit 0
```

Save and exit the file, then make the file executable:

```
chmod +x /etc/init.d/startflop
```

Now use this command to enable it and prioritize it:

```
update-rc.d startflop defaults 80
```

Now make a symbolic link to it, I will explain why in a moment:

```
ln -s /etc/init.d/startflop /usr/bin/floppy
```

The symbolic link will make our life easier. Now, to mount the floppy simply run:

```
floppy start
```

and to unmount the floppy simply run:

```
floppy stop
```

Note that this can only be used to mount floppies that we formatted with "fdformat /dev/fd0u1722"

Also to make life easier is a script that copies the aide database to the floppy, prompting you to insert and remove the floppy as needed. If you already have a file named 'go', don't overwrite it.

```
cd
wget http://www200.pair.com/mecham/spam/go.txt
mv -i go.txt go
chmod +x go
```

To run the script:

```
./go
```

See how little free space we have on the floppy disk:

```
df /floppy
```

Note that if you ever run out of room on the floppy, you can delete any of the files off the floppy that we copied over from the /bin directory (tempfile, hostname, cat, date, rm, grep) due to the fact that we included /bin in our path statement in /floppy/report. Only delete what is absolutely necessary.

See http://www.oreilly.com/catalog/debian/chapter/book/ch04_04.html for help with mounting devices.

****

CPAN, Pflogsumm and trim_whitelist:

Index
****
We are going to install a couple utilities that will be of value to us. One is pflogsumm.pl that produces statistical reports. We will also install a script to maintain the auto-whitelist file. You could install pflogsumm via apt-get but doing it this way familiarizes us with CPAN.

We need to install a couple of programs using CPAN; CPAN is a system created to make it easier to install Perl modules.
See http://www.cpan.org/ for more information about CPAN.

The first time you start up CPAN, it will ask you to configure it. We want to answer "yes" to configure it. We will accept all the default prompts except we need to use http mirrors and not ftp mirrors. If you have a proxy server in front of your spamfilter that requires user names or passwords or special settings, I can't help you with that. You'll just have to look elsewhere for answers. One more note: if you want to start the configuration process over again, at the `cpan>` prompt, enter `o conf init`

Start a CPAN session:
perl -MCPAN -e shell

Accept the default of "yes" at the: Are you ready for manual configuration? [yes]
Accept all the defaults, eventually you will need to (and this may take quite some time to come up):
Select your continent
Select your country

After you have selected these, hit **[space][enter]** *a few times until you see some* ***http*** *servers.*
Do not pick any ftp servers.
Select a couple of the http servers. Hit [enter] to get back to the cpan> prompt.

One more note: do not update the CPAN program itself.

Now let's install a couple of modules we need:

At the cpan> prompt type these commands in exactly:
install File::MMagic

If you are asked any questions, accept the defaults.

Back at the cpan> prompt:
install Date::Calc

If you are asked any questions, accept the defaults.
When it finishes, exit cpan with:

q

This section will set up an automated preparation of a very nicely laid out report of email activity, and email this report to us once per day, with the previous day's mail statistics. We need the pflogsumm.pl utility:

Go to: http://jimsun.linxnet.com/postfix_contrib.html
And by looking at the link to the program, ***edit these next lines if necessary to reflect the latest production version****:*

```
cd /usr/local/src
wget http://jimsun.linxnet.com/downloads/pflogsumm-1.1.1.tar.gz
tar xzvf pflogsumm-1.1.1.tar.gz
cd pflogsumm-1.1.1
cp pflogsumm.pl /usr/sbin
chmod +x /usr/sbin/pflogsumm.pl
cd
```

Let's set up a cron.daily script that prepares and mails us the pflogsumm report. When I first wrote this HOWTO, the mail.log file rotated once each week. Now it seems to rotate sometimes once a day, sometimes twice a day and sometimes every couple days. I used to be able to create a weekly report when the mail.log files were rotated on a predictable basis but now I have to combine several log files just to be insured I have the data I need to report on the previous day's activity.

```
cd /etc/cron.daily
wget http://www200.pair.com/mecham/spam/pflogsumm
chmod +x pflogsumm
```

You may need to edit the script and change the hostname just after the word "DAILY" to reflect your system's hostname. You may also want to change to whom the report gets mailed:

```
vi /etc/cron.daily/pflogsumm
```

Notice that in order to insure we have all the previous day's data, we include the contents of all the files (mail.log*).

There is one major problem with the report. Any email that gets sent to amavisd-new, which includes most mail that isn't rejected at the front door, also comes back from amavisd-new. This means Postfix sees the email twice. So the report lists them twice. Sorry. For those that are interested, there are a couple scripts out there that pre-process the log files to prevent reporting amavis entries http://web.tiscali.it/postfix/prepflog.html and http://www.caspergasper.com/spam.shtml

I suggest you read the pflogsumm.pl faq; your version number of pflogsumm may be different:

```
less /usr/local/src/pflogsumm-1.1.1/pflogsumm-faq.txt
```

Also see this manpage for pflogsumm:
http://manpage.willempen.org/1/pflogsumm

FYI: /etc/crontab controls when the scripts in /etc/cron.daily /etc/cron.weekly and /etc/cron.monthly run. One of the scripts in /etc/cron.weekly is sysklogd. sysklogd reads its configuration from /etc/syslog.conf. You can see what logs should get rotated weekly by sysklogd by using the command syslogd-listfiles --weekly
Also see http://www.ducea.com/2006/06/06/rotating-linux-log-files/ and http://www.freespamfilter.org/forum/viewtopic.php?t=711&start=15 and http://marc.info/?l=postfix-users&m=117755753816999

If you would like to play with another reporting tool, see http://marc.info/?l=postfix-users&m=117937680408576

You will notice that if you use SpamAssassin's auto-whitelist feature, over time the /var/lib/amavis/.spamassassin/auto-whitelist file will grow to a rather large size. The SpamAssassin source code comes with a utility called check_whitelist that can be used to trim the size down. Run perldoc check_whitelist to see how it is used. There is a modified version of check_whitelist called trim_whitelist we are going to use to reduce the size of the auto-whitelist file. This utility is explained here: http://article.gmane.org/gmane.mail.spam.spamassassin.general/59651.

Download trim_whitelist:

```
cd /usr/sbin
wget http://www.deepnet.cx/~kdeugau/spamtools/trim_whitelist
```

Make it executable:

```
chmod +x /usr/sbin/trim_whitelist
```

List our SpamAssassin directory:

```
ls -l /var/lib/amavis/.spamassassin
```

Run the program in order to test it:
su amavis -c '/usr/sbin/trim_whitelist'

List our SpamAssassin directory again:
ls -l /var/lib/amavis/.spamassassin

If the test is successful, the program will have created a new file: auto-whitelist-old

Now we will create a cron job so trim_whitelist will run once each week:
cd /etc/cron.weekly
wget http://www200.pair.com/mecham/debian/trim_whitelist_weekly
chmod +x trim_whitelist_weekly
cd

****

Whitelisting, Blacklisting, Tweaking:

Index
****
Postfix, Amavis and SpamAssassin all have methods to do whitelisting and blacklisting. Postfix has a number of methods to reject mail before it enters the system, we configured some and I provided links to others during the Postfix Anti Spam configuration. We also discussed /etc/postfix/sender_access that can be used to blacklist senders. We created /etc/postfix/header_checks and /etc/postfix/body_checks that can be used for content filtering and I provided links to some examples. The files themselves also provide examples. When we were editing /etc/amavis/amavisd.conf you noticed sections that dealt with whitelisting and blacklisting. It is recommended you do "soft" whitelisting and blacklisting by adding entries to the @score_sender_maps section of amavisd.conf. Add your entries in the same section that 'nobody@cert.org' => -3.0, is listed. Negative scores will be subtracted from the overall spam score, and positive scores will be added.

Note that Postfix and Amavis will need to be shutdown and restarted (or reloaded) after making changes to configuration files:
postfix reload
will cause Postfix to reload its configuration files
/etc/init.d/postfix restart
will shutdown and restart Postfix, and so will:
postfix stop
postfix start

/etc/init.d/amavis stop
/etc/init.d/amavis start

is the best way to handle Amavis, but this works as well:
amavisd-new stop
amavisd-new start as does this:
amavisd-new reload

You can stop and restart amavisd-new while Postfix is running. Delivery to amavisd-new will fail (and you will get errors in the log), but Postfix will queue the mail in the deferred queue. If amavisd-new has only been down for a moment, Postfix will try to send the mail again in about 16.6 minutes (1000 seconds). If you are impatient, or amavisd-new has been down for a longer period, you may wish to use 'postfix flush' to force Postfix to attempt delivery of the mail currently in the deferred queue. If you have made changes to the way postfix communicates with amavisd-new (for example you comment out content_filter in main.cf to bypass amavisd-new) you may need to requeue the mail with
postsuper -r ALL

Configuring and tweaking SpamAssassin is an art in itself. There is plenty of information on the Internet to help guide you through this. I will simply supply a few tweaking links for you to explore. Don't forget that it is possible to write custom rules with negative numbers that can be used to lower the score. Keep in mind that many custom rules once only available as add-ons are now integrated into SpamAssassin version 3, so do some research to find out if you are duplicating rules.

Many people use Rules Du Jour, but remember, if you use a lot of custom rules, you can use more than 100MB RAM for each amavisd process (and create a major load on the system). Because some of these rule sets are updated frequently it increases the possibility that a rule set may have an error in it. If this happens it could halt your server. **NEVER use blacklist.cf, blacklist-uri.cf or bigevil.cf.** I suggest you go slow, add one at a time, then wait a day or so and check your memory usage:
http://www.exit0.us/index.php?pagename=RulesDuJour

As far as custom rules go, I personally use
http://www.rulesemporium.com/rules/mangled.cf

I also download a set of the safest SARE rules:
http://marc.theaimsgroup.com/?l=spamassassin-users&m=115637139728022

---

cd /etc/spamassassin
wget http://saupdates.openprotect.com/pub.gpg
sa-update --import pub.gpg
sa-update --gpgkey D1C035168C1EBC08464946DA258CDB3ABDE9DC10 --channel saupdates.openprotect.com
spamassassin --lint

and if everything looks OK:
/etc/init.d/amavis restart

Personally I use sa-update, and the OpenProtect SARE rules shown in the link above. Once these are properly configured, you could use something like my script that downloads updates from both places, and if updates are downloaded from one or he other, amavisd-new is restarted:

```
cd /usr/sbin
wget http://www200.pair.com/mecham/spam/sa-update.sh
chmod +x sa-update.sh
```

You would place /usr/sbin/sa-update.sh in your crontab (replacing the existing sa-update command).

And here is a note on how to use sa-update instead of RulesDuJour:
http://marc.theaimsgroup.com/?l=spamassassin-users&m=115545719618240

Remember, adding custom rule sets will require additional RAM.
Use `top` to see what's using memory; then change the sort order with > and <

SpamAssassin Rules Emporium (SARE):
http://www.rulesemporium.com/

To learn more about configuring SpamAssassin, run this at the command prompt:
perldoc Mail::SpamAssassin::Conf

Of course visit the SpamAssassin site:
http://spamassassin.apache.org/

Visit the SpamAssassin Wiki for more ideas:
http://wiki.apache.org/spamassassin/FrontPage

and the Exit0 SpamAssassin Wiki
http://www.exit0.us/

This is an example of a /etc/spamassassin/local.cf file:

```
bayes_path /var/lib/amavis/.spamassassin/bayes
auto_whitelist_path /var/lib/amavis/.spamassassin/auto-whitelist
lock_method flock
#
# We need stuff from these senders, and they tend to get marked as spam.
# We want to whitelist our close business partners.
# We subscribe to industry specific newsletters and whitelist them also.
# Later we manually feed these to Bayes as ham.
whitelist_from spambin@example.com
whitelist_from *@generalmotors.com
whitelist_from *.usanewstoday.com
```

```
#
# We need stuff from autonetamerica and the Lottery and it always gets marked as spam.
# So we will create custom rules that let these particular subject lines reduce the score.
header AUTONETAMERICA Subject =~ /Auto Net America/
score AUTONETAMERICA -5.000
header YOURLOTTERY Subject =~ /Your Lottery Results!/
score YOURLOTTERY -5.000
#
# We change the scores on a few standard tests - these are just examples
score RAZOR2_CF_RANGE_51_100 0.500
score URIBL_WS_SURBL 2.000
score URIBL_PH_SURBL 2.500
score RCVD_IN_SORBS_HTTP 1.000
score RCVD_IN_SBL 1.000
score RCVD_IN_NJABL_PROXY 1.000
score RCVD_IN_SORBS_MISC 0.500
score RCVD_IN_BL_SPAMCOP_NET 2.000
score RCVD_IN_NJABL_SPAM 2.200
#
# use_auto_whitelist 0
# uncomment to disable auto-whitelist - a number of people recommend NOT using auto-
whitelist.
```

Make sure you run `spamassassin --lint` after adding any new rules.

Normally, SpamAssassin is set up as a multi user system and users are able to tweak their own personal settings by editing their individual user_prefs file. In our case the only users that use SpamAssassin are amavis and root.
The `/etc/spamassassin/local.cf` file is used to configure SpamAssassin site wide but in our case, editing `/var/lib/amavis/.spamassassin/user_prefs` would have a similar effect. However, certain global SpamAssassin settings will have no effect if placed in user_prefs, therefore you only need to maintain local.cf.

If image spam is a problem for you, please see another one of my documents that attempts to address that issue. This type of spam is particularly dangerous because it tries very hard to corrupt your Bayes database:

http://www200.pair.com/mecham/spam/image_spam2.html

****

Use the Rescue CD:

Index

****

Boot up using the etch CD, answer the first few prompts (Language, Country, Keyboard), let it discover devices (but go no further!), then use [Alt]+F2 to open a console. At this point you can mount your hard drive and repair GRUB or LILO (and perform other tasks). Here is how I used the rescue CD to repair LILO after a disk clone using Ghost 2003. You MUST know in advance which

partition is your root partition. You should keep a copy of the text in /etc/fstab to refer to. In this example we will assume the root partition was mounted on /dev/hda6, a separate /boot partition was mounted on /dev/hda1 and you are repairing LILO. If you use other mount points (like /var for example) (depending on what you are trying to repair) you may need to mount those too.

To repair LILO:

```
mkdir /hardroot
mount /dev/hda6 /hardroot
chroot /hardroot
```

If you have a separate boot partition, this would need to be mounted too:

```
mount /dev/hda1 /boot
cp /etc/lilo.conf /etc/lilo.conf-original
/sbin/lilo
```

Use [Ctrl][Alt][Del] to reboot. I got a bunch of errors when I ran 'lilo', but it booted up just fine. Once it reboots you should run:

```
shutdown -r -F now
```

To repair GRUB, you would instead run this command (edit as needed to reflect the type of disk you are using):

```
/sbin/grub-install /dev/hda
```

[Ctrl][Alt][Del]

****

Links, FAQs and such:

Index

****

debian http://www.debian.org
postfix http://www.postfix.org
amavisd-new http://www.ijs.si/software/amavisd/
SpamAssassin http://www.spamassassin.org
Razor http://razor.sourceforge.net
DCC http://www.rhyolite.com/anti-spam/dcc/
Pyzor http://pyzor.sourceforge.net
ClamAV http://www.clamav.net

Similar Documents (for other distributions):
http://www.geekcomix.com/cgi-bin/classnotes/wiki.pl?Setting_Up_An_Anti-SPAM_Gateway
http://www.flakshack.com/anti-spam/
http://www.freespamfilter.org/
http://ezine.daemonnews.org/200309/postfix-spamassassin.html
http://freshmeat.net/articles/view/857/

http://www.gentoo.org/doc/en/mailfilter-guide.xml?style=printable
http://www.novell.com/coolsolutions/feature/16093.html

Other Debian documents that also include local mail delivery:
http://workaround.org/articles/ispmail-sarge/
http://www.fatofthelan.com/articles/articles.php?pid=22
http://www.falkotimme.com/howtos/perfect_setup_debian_sarge/index.php

Would you like a web based interface for per-user settings and quarantine control? Don't install this on a production machine until you are comfortable with it and keep in mind that the quarantined email is stored in a MySQL database on this machine which means you need a disk partition large enough to hold it. This software is not without issues and installing and configuring Maia Mailguard is not trivial so I suggest you read through the documentation and mailing list archives.
http://www.renaissoft.com/projects/maia/index.php
http://www.renaissoft.com/pipermail/maia-users/
http://www.linuxjournal.com/article/7427
If you are interested, this will help you get it installed:
http://www200.pair.com/mecham/spam/debian-etch-maia.html

Email testing tools:
http://www.declude.com/Articles.asp?ID=100
http://zmailer.org/mxverify.html
http://www.gfi.com/emailsecuritytest/
http://dnsreport.com/ and http://www.dnsstuff.com/
http://www.eicar.org/anti_virus_test_file.htm
http://www.webmail.us/testvirus
http://rbls.org/
http://spamlinks.net/tools-relay.htm#web

apt-get and dpkg:
http://oat.tao.ca/node/view/141
http://www.knoppix.net/forum/viewtopic.php?t=2638
http://jaqque.sbih.org/kplug/apt-pinning.html
http://www.debian.org/doc/manuals/apt-howto/
http://newbiedoc.sourceforge.net/tutorials/apt-get-intro/info.html.en
http://www.oreilly.com/catalog/linuxnut4/chapter/ch05.pdf

neat way to upgrade only 'stable' packages:
http://www.freespamfilter.org/forum/viewtopic.php?t=265

iptables:
http://iptables-tutorial.frozentux.net/iptables-tutorial.html
http://www.cites.uiuc.edu/wsg/talks/iptables/iptables-rh8.rules.txt
http://www.hackinglinuxexposed.com/articles/20030703.html
http://www.justlinux.com/nhf/Security/IPtables_Basics.html
http://morizot.net/firewall/gen/

DNS based spam databases (RBLs):
http://www.declude.com/junkmail/support/ip4r.htm

UTF-8
http://www.lugod.org/mailinglists/archives/vox-tech/2003-06/msg00128.html

Postfix:
http://www.postfix.org/BASIC_CONFIGURATION_README.html
http://www.postfix.org/STANDARD_CONFIGURATION_README.html
http://www.stahl.bau.tu-bs.de/~hildeb/postfix/
http://www.postfix-book.com/debugging.html
http://www.metaconsultancy.com/whitepapers/smtp.htm
http://mywebpages.comcast.net/mkettler/sa/SA-rules-howto.txt
http://www.mines.edu/academic/computer/spamuserprefs.shtml
http://www.postfix.org/uce.html
http://jimsun.linxnet.com/misc/postfix-anti-UCE.txt
http://www.geekounet.org/filters/header_checks
http://www.fredshack.com/docs/postfix.html

Security:
http://www.debian.org/doc/manuals/securing-debian-howto/
http://linuxgazette.net/issue89/gonzales.html
http://www.acm.org/crossroads/xrds6-1/linuxsec.html
http://www.tldp.org/HOWTO/Security-HOWTO/index.html
http://www.freefire.org/lib/hardening.en.php3

Linux:
http://linux.about.com/library/bl/open/newbie/blnewbie_toc.htm
http://loll.sourceforge.net/linux/links/index.html
http://www.fifi.org/cgi-bin/man2html
http://wireless.ictp.trieste.it/school_2002/lectures/fonda/MoreLinux/rute/node1.html

Cron:
http://www.unixgeeks.org/security/newbie/unix/cron-1.html
http://www.pantz.org/os/linux/programs/cron.shtml

AIDE:
http://linuxgazette.net/issue75/maiorano.html
http://www.gnu.org/directory/aide.html

Debian:
http://articles.linmagau.org/modules.php?op=modload&name=Sections&file=index&req=viewarticle&artid=158&page=1
http://www.debianplanet.org/index.php?or=7
http://newbiedoc.sourceforge.net/
http://www.debian.org/distrib/packages

http://www.desktop-linux.net/debian-rclocal.htm
http://www.backports.org
http://www.falkotimme.com/howtos/index.php

Amavisd-new and SpamAssassin:
http://wiki.apache.org/spamassassin/FrontPage
http://www.exit0.us/
http://www.rulesemporium.com/index.html
http://wiki.apache.org/spamassassin/CustomRulesets
http://www.ijs.si/software/amavisd/#faq-spam
http://sourceforge.net/mailarchive/forum.php?forum=amavis-user
http://news.gmane.org/gmane.mail.spam.spamassassin.general
http://www.surbl.org/

Spam Links:
http://spamlinks.net/

Regular Expressions (Regex):
http://www.regular-expressions.info
http://www.silverstones.com/thebat/Regex.html#intro
http://gmckinney.info/resources/regex.pdf
http://www.contactor.se/~dast/mail2sms/regex.shtml
http://www.quanetic.com/regex.php
http://www.koralsoft.com/regextester/
http://www.renatomancuso.com/software/pcreworkbench/pcreworkbench.htm

Disk Cloning:
http://service1.symantec.com/SUPPORT/ghost.nsf/docid/2001111413481325?Open&src=&docid=19
http://www.rajeevnet.com/hacks_hints/os_clone/os_cloning.html
http://service1.symantec.com/SUPPORT/ghost.nsf/docid/2000042113083825?Open&src=&docid=1999021909463125

Populating relay_recipient_maps:
http://www2.origogeneris.com:4000/relay_recipients.html
http://www-personal.umich.edu/~malth/gaptuning/postfix/
http://www.unixwiz.net/techtips/postfix-exchange-users.html
http://postfix.state-of-mind.de/patrick.koetter/mailrelay/
http://www200.pair.com/mecham/spam/PostfixAddressExtract.vbs.txt

Configuration files quick reference:

/etc/aliases
/etc/postfix/main.cf
/etc/postfix/master.cf
/etc/postfix/sender_access
/etc/postfix/transport
/etc/postfix/virtual

/etc/postfix/relay_recipients
/etc/postfix/body_checks
/etc/postfix/header_checks
/etc/amavis/amavisd.conf
/var/lib/amavis/.spamassassin/user_prefs
/etc/spamassassin/local.cf
/var/lib/amavis/.razor/razor-agent.conf
/etc/clamav/clamd.conf
/etc/apt/preferences
/etc/apt/sources.list

A few unmentioned commands and FAQs:

How do I tell how much space is left on the drive?
Answer: df

How do I tell how my hard drive is partitioned?
fdisk -l /dev/hda or /dev/sda

How do I delete just one email that is in the queue?
Answer: first run "mailq" and note the ID number. Then run postsuper -d <ID number>

I'm still in the testing stages of this box and I have a bunch of test messages in the queue that I want to delete, how do I do this?
Answer: postsuper -d ALL deferred
This will delete messages in the deferred queue.

I would like to change the IP address of this box, how do I do this?.
Stop Postfix and amavisd-new.
Look in these files, and edit as needed:
/etc/network/interfaces
/etc/hosts
/etc/networks
/etc/resolv.conf
/etc/postfix/main.cf
possibly /etc/postfix/sender_access
Run "postmap /etc/postfix/sender_access"
possibly /etc/postfix/transport
Run "postmap /etc/postfix/transport"
possibly /etc/spamassassin/local.cf
Modify your firewall settings, and send the iptables commands.
reboot
Modify other computers' hosts file(s) as needed.
Change PuTTY to the new address
If you find more places to change, let me know,

I want to insure postmaster and abuse @example.com receives mail even if it is spam, how do I do that?
Set:

@spam_lovers_maps = (['postmaster@example.com',
'abuse@example.com'],);

How do I bypass scanning for some senders or recipients?
http://www200.pair.com/mecham/spam/bypassing.html

I want to have my Exchange box send its mail out through the spamfilter, so I can further isolate it from the Internet.
See these posts: http://www.freespamfilter.org/forum/viewtopic.php?t=82
http://marc.theaimsgroup.com/?l=amavis-user&m=113415019700881

The post above also explains how to use altermime to attach a disclaimer to outgoing mail (mail that is sent from the Exchange server).

Some final comments:

Spam traffic is increasing dramatically. About 90% of the email that passes through my system is spam. Setting this system up won't be the end of your anti-spam project, I'm afraid. You'll catch a LOT of spam with this, but some will keep getting through, and some good email will get blocked. You'll learn and tweak things as you go forward.

I hope this doc is useful to you. Feel free to contact me, mr88talent at yahoo dot com, with comments and suggestions or corrections. This website was created for my own personal use and entertainment.

****

Disclaimer:

Index
****

Where links are provided to other websites,
the author accepts no responsibility and shall not be liable,
either directly or indirectly for the content,
legality, accuracy, reliability, suitability, quality
or decency of content, information, product, advice
or services provided by and contained in those sites.

Downloading any information from the Internet is done
at your own risk, and the risk can be substantial.
You knew that, right?